IWCS 2023

The 15th International Conference on Computational Semantics

Proceedings of the Conference

June 21 - 23, 2023
Nancy, France

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-959429-74-6

Message from the Organizers

These are the conference proceedings of IWCS 2023, the 15th edition of the International Conference on Computational Semantics. This conference is supported by the Laboratoire Lorrain de recherche en informatique et ses applications (LORIA), the Institut des Sciences du Digital Management et Cognition (IDMC), the CNRS, Inria, Université de Lorraine, Göteborgs universitet, the Métropole du Grand Nancy, the Atelier du vélo de Maxéville, and Erdil.

This edition of IWCS takes place after a fully remote edition of the conference in Groningen 2021. In order to promote the community getting together IRL after the pandemic IWCS 2023 is held in an in-person format. However, recordings of all talks are available on the iwcs2023.loria.fr `http://iwcs2023.loria.fr/` website. IWCS 2023 spans three days – 21 - 23 June – with an additional day of satellite workshops before the conference:

- **DMR 2023**: The Fourth International Workshop on Designing Meaning Representation
- **InqBnB4**: Inquisitiveness Below and Beyond the Sentence Boundary
- **ISA-19**: 19th Joint ACL – ISO Workshop on Interoperable Semantic Annotation
- **NALOMA'23**: Natural Logic meets Machine Learning 2023

55 submissions were made to the main conference (31 long and 24 short). Each paper was reviewed by three reviewers. 20 long papers were accepted, one of which was withdrawn. This results in 19 long and 14 short papers with a final acceptance rate of 60 % (61 % for long and 58 % for short papers). The final programme is diverse with topics ranging from semantic parsing, question answering, knowledge extraction, semantics representation and Large Language Models. The programme also features two keynotes given by Rachel Fernandez (University of Amsterdam) and Lucia Donatelli (Vrije Universiteit in Amsterdam). We thank them for participating in IWCS 2023! Abstracts of their contributions are available in this volume.

In keeping with IWCS tradition, an unconference event was also organised, to provide an opportunity for open discussion on subjects proposed by conference participants. It's a vital time to take stock of the issues facing the community and to structure ourselves better. Among the topics were ethical issues, compositionality and shared tasks, and closed-source models in NLP. Once again, the discussions proved to be extremely interesting and opened up new avenues for future developments.

We take the opportunity to congratulate the best paper awards:

- Best short paper award goes to Dmitry Nikolaev and Sebastian Padó for their article and poster "The Universe of Utterances According to BERT".

- Best long paper award goes to Jonghyuk Park, Alex Lascarides and Subramanian Ramamoorthy for their article "Interactive Acquisition of Fine-grained Visual Concepts by Exploiting Semantics of Generic Characterizations in Discourse".

Don't hesitate to watch the recordings of their presentations!

This edition of IWCS, which brought participants together in the Octroi hall for the social event, could not have been held without the unfailing help of the local team known as the red t-shirts team. They did an incredible job to make sure everything was ready and running smoothly. We'd like to thank them for their time and energy. We would also like to thank our Programme Committee members for their detailed and helpful reviews.

We hope that IWCS2023 has been an exciting edition of the conference series that will have inspired the computational semantics community to continue discussions while integrating many new themes.

Maxime Amblard and Ellen Breitholtz

Organizing Committee

Organisers:

Local Chair: Maxime Amblard
Program Chairs: Maxime Amblard and Ellen Breitholtz

Program Commitee:

- Rodrigo Agerri
- Alexander Berman
- Jean-Philippe Bernardy
- Yuri Bizzoni
- Moritz Blum
- Maria Boritchev
- António Branco
- Paul Buitelaar
- Harry Bunt
- Aljoscha Burchardt
- Stergios Chatzikyriakidis
- Rui Chaves
- Philipp Cimiano
- Robin Cooper
- Paula Czarnowska
- Philippe de Groote
- Markus Egg
- Guy Emerson
- Katrin Erk
- Arash Eshghi
- Kilian Evang
- Meaghan Fowlie
- Diego Frassinelli
- André Freitas
- Jonathan Ginzburg
- Eleni Gregoromichelaki
- Christine Howes
- Elisabetta Jezek
- Rohit Kate
- Gene Kim
- Ralf Klabunde
- Pavel Kovalev
- Nikhil Krishnaswamy
- Staffan Larsson
- Alex Lascarides
- Alessandro Lenci
- Chuyuan Li
- Vladislav Maraev
- Aleksandre Maskharashvili
- Louise McNally
- Koji Mineshima
- Pascale Moreira
- Larry Moss
- Bill Noble
- Sebastian Padó
- Siyana Pavlova
- Sandro Pezzelle
- Manfred Pinkal
- Paul Piwek
- Violaine Prince
- Stephen Pulman
- Allan Ramsay
- Christian Retoré
- Valentin Richard
- German Rigau
- Mats Rooth
- Mehrnoosh Sadrzadeh
- Asad Sayeed
- Nathan Schneider
- Sabine Schulte im Walde
- Tim Van de Cruys
- Christian Wartena
- Matthijs Westera
- Gijs Wijnholds
- Hitomi Yanaka
- Roberto Zamparelli
- Elena Zotova

Local Staff:

- Marie Cousin
- Valentin Richard
- Siyana Pavlova
- Amandine Lecomte
- Amandine Decker
- Chuyuan Li
- Hee-Soo Choi
- Vincent Tourneur
- Khensa Daoudi
- Bruno Guillaume
- Julie Halbout
- Fanny Ducel
- Laura Masson-Grehaigne
- Nathalie Fritz
- Delphine Hubert
- Anne-Marie Messaoui
- Marie Baron
- Mariana Diaz
- Marie Buchheit
- Marie-Luce Boulet

Invited Speakers

Raquel Fernandez, Institute for Logic, Language & Computation, University of Amsterdam

Common Ground and Audience Design in Referential Games

In conversation, we decide what to say and how to say it on the basis of what we share with our dialogue partner. Yet, it is an open question how such accommodation can be modelled in computational agents. Taking a visually grounded referential game as test bed, in this talk I will present recent work where we use computational methods to analyse repeated references exhibiting lexical entrainment and model audience-aware adaptation between agents with asymmetric knowledge.

Lucia Donatelli, Computational Linguistics and Text Mining Lab (CLTL), Vrije Universiteit

Compositionality and Its Discontents

How do we bridge the gap between compositional semantics and broader notions ofmeaning in computational linguistics? In this talk, I will address this question from several angles. First, I will look at the challenge of adequately representing semantic structure when designing meaning representations, given distinct theoretical and practical considerations. I will present a semantic parsing methodology for normalizing discrepancies between representations at the compositional level to better understand which design differences are semantically rooted and which are superficial. Next, I will present work discussing how representting compositional structure helps on tasks such as cross-lingual parsing and compositional generalization. Finally, I will discuss implications of structured representations and models for generalizing to applications such as situated dialogue and interaction, where compositional semantics alone seems insufficient for robust performance.

Table of Contents

Conference Program

Wednesday, June 21, 2023

10:00–11:00 **Invited speaker: Raquel Fernandez**
Common Ground and Audience Design in Referential Games

11:00–12:30 **Main session 1**
Chair: James Pustejovsky

11:00–11:30 *Can current NLI systems handle German word order? Investigating language model performance on a new German challenge set of minimal pairs*
Ines Reinig and Katja Markert

11:30–12:00 *Contextual Variability depends on Categorical Specificity rather than Conceptual Concreteness: A Distributional Investigation on Italian data*
Giulia Rambelli and Marianna Bolognesi

12:00–12:30 *Probing BERT's ability to encode sentence modality and modal verb sense across varieties of English*
Jonas Wagner and Sina Zarrieß

14:00–15:00 **Main session 2**
Chair: Raquel Fernandez

14:00–14:30 *Dense Paraphrasing for Textual Enrichment*
Jingxuan Tu, Kyeongmin Rim, Eben Holderness, Bingyang Ye and James Pustejovsky

14:30–15:00 *Towards Unsupervised Compositional Entailment with Multi-Graph Embedding Models*
Lorenzo Bertolini, Julie Weeds and David Weir

Wednesday, June 21, 2023 (continued)

15:00–16:00 Poster session 1

Gender-tailored Semantic Role Profiling for German
Manfred Klenner, Anne Göhring, Alison Kim and Dylan Massey

Implicit causality in GPT-2: a case study
Minh Hien Huynh, Tomas Lentz and Emiel van Miltenburg

Multi-purpose neural network for French categorial grammars
Gaëtan Margueritte, Daisuke Bekki and Koji Mineshima

Experiments in training transformer sequence-to-sequence DRS parsers
Ahmet Yildirim and Dag Haug

Unsupervised Semantic Frame Induction Revisited
Younes Samih and Laura Kallmeyer

Towards Ontologically Grounded and Language-Agnostic Knowledge Graphs
Walid Saba

The Universe of Utterances According to BERT
Dmitry Nikolaev and Sebastian Padó

Wednesday, June 21, 2023 (continued)

16:30–17:30 **Main session 3**
Chair: Daisuke Bekki

16:30–17:00 *Sparser is better: one step closer to word embedding interpretability*
Simon Guillot, Thibault Prouteau and Nicolas Dugue

17:00–17:30 *Semantically Informed Data Augmentation for Unscoped Episodic Logical Forms*
Mandar Juvekar, Gene Kim and Lenhart Schubert

Thursday, June 22, 2023

09:30–10:30 **Main session 4**
Chair: Ellen Breitholtz

09:30–10:00 *Meaning-Text Theory within Abstract Categorial Grammars: Toward Paraphrase and Lexical Function Modeling for Text Generation*
Marie Cousin

10:00–10:30 *Measuring Fine-Grained Semantic Equivalence with Abstract Meaning Representation*
Shira Wein, Zhuxin Wang and Nathan Schneider

11:00–12:30 **Main session 5**
Chair: Lucia Donatelli

11:00–11:30 *The Importance of Context in the Evaluation of Word Embeddings: The Effects of Antonymy and Polysemy*
James Fodor, Simon De Deyne and Shinsuke Suzuki

11:30–12:00 *RaTE: a Reproducible automatic Taxonomy Evaluation by Filling the Gap*
Phillippe Langlais and Tianjian Lucas Gao

12:00–12:30 *The Universal Anaphora Scorer 2.0*
Juntao Yu, Michal Novák, Abdulrahman Aloraini, Nafise Sadat Moosavi, Silviu Paun, Sameer Pradhan and Massimo Poesio

Thursday, June 22, 2023 (continued)

14:00–15:30 **Main session 6**
Chair: Lasha Abzianidze

14:00–14:30 *The Sequence Notation: Catching Complex Meanings in Simple Graphs*
Johan Bos

14:30–15:00 *Bridging Semantic Frameworks: mapping DRS onto AMR*
Siyana Pavlova, Maxime Amblard and Bruno Guillaume

15:00–15:30 *Data-Driven Frame-Semantic Parsing with Tree Wrapping Grammar*
Tatiana Bladier, Laura Kallmeyer and Kilian Evang

Friday, June 23, 2023

10:00–11:00 **Invited speaker: Lucia Donatelli**
Compositionality and Its Discontents

11:00–12:30 **Poster session 2**

The argument–adjunct distinction in BERT: A FrameNet-based investigation
Dmitry Nikolaev and Sebastian Padó

Collecting and Predicting Neurocognitive Norms for Mandarin Chinese
Le Qiu, Yu-Yin Hsu and Emmanuele Chersoni

Error Exploration for Automatic Abstract Meaning Representation Parsing
Maria Boritchev and Johannes Heinecke

Unsupervised Methods for Domain Specific Ambiguity Detection. The Case of German Physics Language
Vitor Fontanella, Christian Wartena and Gunnar Friege

Definition Modeling : To model definitions. Generating Definitions With Little to No Semantics
Vincent Segonne and Timothee Mickus

SMARAGD: Learning SMatch for Accurate and Rapid Approximate Graph Distance
Juri Opitz, Philipp Meier and Anette Frank

AMR4NLI: Interpretable and robust NLI measures from semantic graph
Juri Opitz, Shira Wein, Julius Steen, Anette Frank and Nathan Schneider

14:00–16:00 **Main session 7**
Chair: Stergios Chatzikyriakidis

14:00–14:30 *Use Defines Possibilities: Reasoning about Object Function to Interpret and Execute Robot Instructions*
Mollie Shichman, Claire Bonial, Austin Blodgett, Taylor Hudson, Francis Ferraro and Rachel Rudinger

14:30–15:00 *SimpleMTOD: A Simple Language Model for Multimodal Task-Oriented Dialogue with Symbolic Scene Representation*
Bhathiya Hemanthage, Christian Dondrup, Phil Bartie and Oliver Lemon

15:00–15:30 *Grounding and Distinguishing Conceptual Vocabulary Through Similarity Learning in Embodied Simulations*
Sadaf Ghaffari and Nikhil Krishnaswamy

15:30–16:00 *Interactive Acquisition of Fine-grained Visual Concepts by Exploiting Semantics of Generic Characterizations in Discourse*
Jonghyuk Park, Alex Lascarides and Subramanian Ramamoorthy

Can current NLI systems handle German word order? Investigating language model performance on a new German challenge set of minimal pairs

Ines Reinig
Data and Web Science Group
Mannheim University
Germany
ines.reinig@uni-mannheim.de

Katja Markert
Institute of Computational Linguistics
Heidelberg University
Germany
markert@cl.uni-heidelberg.de

Abstract

Compared to English, German word order is freer and therefore poses additional challenges for natural language inference (NLI). We create WOGLI (Word Order in German Language Inference), the first adversarial NLI dataset for German word order that has the following properties: (i) each premise has an entailed and a non-entailed hypothesis; (ii) premise and hypotheses differ only in word order and necessary morphological changes to mark case and number. In particular, each premise and its two hypotheses contain exactly the same lemmata. Our adversarial examples require the model to use morphological markers in order to recognise or reject entailment. We show that current German autoencoding models fine-tuned on translated NLI data can struggle on this challenge set, reflecting the fact that translated NLI datasets will not mirror all necessary language phenomena in the target language. We also examine performance after data augmentation as well as on related word order phenomena derived from WOGLI. Our datasets are publically available at https://github.com/ireinig/wogli.

1 Introduction

German is endowed with a rather free word order (Bader and Portele, 2019), especially when it comes to ordering nominal arguments in a sentence. Currently, large German NLI datasets are only available as translations from other languages. For example, the training portion (392k pairs) of the German XNLI dataset (Conneau et al., 2018) is a machine translation of the English MultiNLI training set (Williams et al., 2018). The testing portion of German XNLI is a manual translation of 5k English premise-hypothesis pairs that were newly created by the authors of XNLI. Such translated sets do not necessarily mirror all German-specific linguistic phenomena, such as the freer German word order.

We construct a new German challenge set named WOGLI (Word Order in German Language Inference). This dataset is handcrafted and does not stem from translation. It contains 16k premises where each premise is accompanied by one entailed (E) and one non-entailed (NE) hypothesis that both contain the same lemmata as the premise but change argument order. Morphological markers are indicative of subject and (direct) object, thus informing about the hypothesis' entailment relationship to the premise. In other words, WOGLI serves as a test bed for current language models' capabilities to distinguish subject from object in the context of German word order.

Our contributions are as follows:

1. We propose the first NLI dataset that specifically targets German word order phenomena.
2. We show that current German autoencoding models fine-tuned on the translated XNLI dataset can struggle on our proposed challenge set (Sections 4 and 5), tending to always predict entailment for both hypotheses.
3. We show that data augmentation can help performance on WOGLI but needs a considerable number of examples to work (Section 6).
4. We derive generalization sets including similar word order phenomena to WOGLI to investigate how the augmented models transfer to these datasets and show that German word order remains challenging in NLI (Section 7).

All our datasets are publically available[1].

2 German Word Order

The topological model. The topological model (Drach, 1937) describes regularities in German

[1] https://github.com/ireinig/wogli

Proceedings of the 15th International Conference on Computational Semantics pages 1–15
June 21–23, 2023. ©2023 Association for Computational Linguistics

Clause	Order	Prefield	L brack.	Middlefield	R brack.	Count (% of accus.)
Main	SO	**Peter**	sieht	den Mann		231 (86%)
		Peter	sees	the man$_{ACC}$		
		Peter	*sees*	*the man*		
	OS	Den Mann	sieht	**Peter**		38 (14%)
		The man$_{ACC}$	sees	**Peter**		
		Peter	*sees*	*the man*		
Emb.	SO		dass	**Peter** den Mann	sieht	546 (99%)
			that	**Peter** the man$_{ACC}$	sees	
			that	*Peter sees the man*		
	OS		dass	den Mann **Peter**	sieht	6 (1%)
			that	the man$_{ACC}$ **Peter**	sees	
			that	*Peter sees the man*		

Table 1: Examples for word order in declarative, active German main and embedded clauses with subject and (accusative) direct object arguments, with corpus statistics from Bader and Häussler (2010). As in the remainder of this paper, the subject is always bold. Transliterations and translations (in italics) are provided below each example.

word order, dependent on the concepts of *prefield* and *middlefield* for constituent positioning. In this model, so-called *left and right brackets* form “[t]he skeleton of the sentence” (Bader and Häussler, 2010, p. 719), while other fields are defined according to the position of the verb (Dürscheid, 2012).

Declarative main clauses, such as *Peter sieht den Mann* at the top of Table 1, have a verb-second order. The left bracket contains the finite verb and the prefield is filled with one constituent (Bader and Häussler, 2010; Dürscheid, 2012). In contrast, embedded clauses, such as *dass Peter den Mann sieht* in the bottom half of Table 1, have a verb-last order. In verb-last clauses, the left bracket is occupied by a subjunction, the right bracket by a finite verb or a verb complex, and other constituents are placed in the middlefield (Dürscheid, 2012).

While subject followed by object (SO) is viewed as the canonical word order, it is possible to place the object before the subject (OS) in both embedded and main clauses (Table 1). In the main clause either the subject or object is placed in the prefield, in embedded clauses both are placed in the middlefield but in varying order.

OS acceptability and minimal pairs. The marked OS order is more frequent in main clauses involving the prefield (Bader and Häussler, 2010) (around 14% of main clauses with accusative direct object) and in the active voice (Bader et al., 2017) (see data and examples in Table 1). Therefore, we construct our challenge set using only such clauses to raise acceptability of the marked OS word order examples. Even in the prefield, OS order can vary in acceptability dependent on relative constituent weight (Siewierska, 1993) (shorter before longer), discourse properties such as givenness (Bader and Portele, 2019) (given before new) and semantic properties such as agency (Siewierska, 1993; Bader and Häussler, 2010) (animate before inanimate). As we focus on simple grammatical examples without further interference, however, all our constituents are short and all premises and hypotheses are single sentences. To ensure that entailed and non-entailed sentences are semantically plausible, all our constituents refer to persons.

German word order in XNLI. We extract hypotheses in the training portion of the translated German XNLI (henceforth, GXNLI-train) that are declarative main clauses with a length between 4 and 9 tokens. The 38,090 extracted clauses are in active voice and contain one subject NP and one direct object NP in accusative case. We exclude clauses that start with prepositions or adverbs to limit ourselves to prefield cases. Only 1.8% (698 clauses) of the extracted clauses are in OS order, compared to the 14% to be expected in a German corpus according to Bader and Häussler (2010). Additionally, a vast majority of the 698 OS clauses start with the same demonstrative pronoun object *das*/*this*, e.g. *Das werde ich tun*/*This I will do*, thus offering little variety. The extreme prevalence of the SO order in GXNLI-train hypotheses may be due to its translated nature.

3 WOGLI construction

Verb Collection. We collected 50 frequent German transitive verb types including agentive (such as *warnen/warn*), object-experiencer (such as *erschrecken/startle*) and subject-experiencer (such as *lieben/love*) verbs. All verbs can take animate (human) subjects as well as animate (human) direct objects, and all objects take the accusative case. All verbs are not symmetric, meaning that they do not lead to bidirectional entailments.[2] In addition, none of the verbs need to split prefixes when used in main clauses so that the resulting premises have a very simple SVO structure. All verbs occur at least 70 times in GXNLI-train. Consequently, any difficulties that a language model will experience are unlikely to be due to verb rarity.

Noun Collection. We collected 144 noun types describing humans that function as direct object or subject in our premises/hypotheses. These include 38 masculine common nouns such as *Gast/guest*, each of which was seen at least 10 times in GXNLI-train and 24 feminine common nouns such as *Lehrerin/(female) teacher*. We collected feminine common nouns by searching for the suffix *in* in GXNLI-train, which often indicates female persons in German. The unbalanced masculine-feminine split is due to the automatic translation of GXNLI-train as gender-neutral English job descriptions, for example *doctor*, are most frequently translated via the German male form, e.g. *Arzt* instead of the female form *Ärztin*[3]. We also collected 41 female and 41 male first names that occur at least 10 times in GXNLI-train. The 144 noun types yield 181 different noun surface forms (nominative/accusative, plural/singular).

Premise and Hypothesis Generation. We automatically generated German premises as declarative, present tense, main clauses in the active voice with SVO structure (see lines 1 and 5) in Table 2). Each SVO premise is accompanied by two hypotheses. H1-SO (NE) exchanges object and subject including changing S/O case markers and potentially verb number markers. Therefore, similarly to English, this change leads to non-entailment, as the premise *The doctor warns the client* and the corresponding H1 *The client warns the doctor* illustrate. We call this subset WOGLI-SO, as the new subject precedes the object. H2-OS (E) simply swaps argument order but keeps case and number markers intact, leading to a sentence synonymous to the premise but with marked OS word order. The resulting set of entailed hypotheses is called WOGLI-OS. Table 2 shows two full examples with case and number marking.

We have 17 patterns due to combinations of different argument NPs, including masculine and feminine proper names and common nouns as well as singular and plural arguments. Subjects/objects are either a simple proper name (such as *Maria*) or consist of an article[4] and a common noun, e.g. *der Arzt/the doctor*. Consequently, each sentence always has a length of four or five words. A list of all 17 patterns is provided in Table 6 in the Appendix; we exclude the patterns in Table 7 in the Appendix as they generate ambiguous hypotheses, due to the absence of disambiguating morphological markers. The 17 patterns in WOGLI can be divided into two groups: 5 **all-singular** patterns that combine two singular nominal arguments (see first example in Table 2) and 12 **singular-plural** patterns in which one argument is singular and the other one is plural (see second example in Table 2). In all 9 patterns involving a masculine singular NP, (i) masculine determiners and (ii) masculine common nouns belonging to the weak declension type[5] carry morphological markers of case. Proper nouns never change surface forms. Additionally, in all **singular-plural** patterns, verb number agreement with the subject always leads to a change in the verb's surface form between E and NE hypotheses.

WOGLI statistics. We generate 1,000 premises per pattern by randomly selecting an appropriate subject/object and verb from our lists, leading to 17,000 possible premises. As in random generation, some premises are generated twice, we deduplicate and are left with 16,971 premises. H1-SO (NE) and H2-OS (E) are deterministically generated from the premises, leading to 33,942 sentence pairs.

[2] For example, for the symmetric verb *heiraten/marry*, X marries Y would entail Y marries X, which would not allow us to automatically derive non-entailed hypotheses.

[3] We could have made up the shortfall by including more feminine forms, even if they do not occur in GXNLI-train, but we consider it more important for this study to keep lexical differences to the fine-tuning set minimal.

[4] We used the articles *ein* (indef.), *der* (def.) and *dieser* (demonstrative), as well as their feminine and plural forms.

[5] The six masculine common nouns in WOGLI that belong to the weak declension type are *Kunde/Kunden/client*, *Student/Studenten/student*, *Journalist/Journalisten/journalist*, *Patient/Patienten/patient*, *Soldat/Soldaten/soldier* and *Zeuge/Zeugen/witness*. The remaining masculine nouns, e.g. *Anwalt/lawyer*, maintain the same surface forms in nominative and accusative.

Premise	**Der/Dieser/Ein**$_{NOM-SG-M}$ **The/This/A**$_{NOM-SG-M}$ *The/This/A*	**Arzt** **doctor** *doctor*	warnt$_{SG}$ warns$_{SG}$ *warns*	den/diesen/einen$_{ACC-SG-M}$ the/this/a$_{ACC-SG-M}$ *the/this/a*	Kunden† client *client*
H1-SO (NE)	**Der/Dieser/Ein**$_{NOM-SG-M}$ **The/This/A**$_{NOM-SG-M}$ *The/This/A*	**Kunde**† **client** *client*	warnt$_{SG}$ warns$_{SG}$ *warns*	den/diesen/einen$_{ACC-SG-M}$ the/this/a$_{ACC-SG-M}$ *the/this/a*	Arzt doctor *doctor*
H2-OS (E)*	Den/Diesen/Einen$_{ACC-SG-M}$ The/This/A$_{ACC-SG-M}$ *The/This/A*	Kunden† client *doctor*	warnt$_{SG}$ warns$_{SG}$ *warns*	**der/dieser/ein**$_{NOM-SG-M}$ **the/this/a**$_{NOM-SG-M}$ *the/this/a*	**Arzt** **doctor** *client*
H3-OS (NE)*	Den/Diesen/Einen$_{ACC-SG-M}$ The/This/A$_{ACC-SG-M}$ *The/This/A*	Arzt doctor *client*	warnt$_{SG}$ warns$_{SG}$ *warns*	**der/dieser/ein**$_{NOM-SG-M}$ **the/this/a**$_{NOM-SG-M}$ *the/this/a*	**Kunde**† **client** *doctor*
Premise	**Der/Dieser/Ein**$_{NOM-SG-M}$ **The/This/A**$_{NOM-SG-M}$ *The/This/A*	**Minister** **minister** *minister*	empfiehlt$_{SG}$ recommends$_{SG}$ *recommends*	die/diese$_{ACC-PL-F}$ the/these$_{ACC-PL-F}$ *the/these*	Autorinnen authors *authors*
H1-SO (NE)	**Die/Diese**$_{NOM-PL-F}$ **The/These**$_{NOM-PL-F}$ *The/These*	**Autorinnen** **authors** *authors*	empfehlen$_{PL}$ recommend$_{PL}$ *recommend*	den/diesen/einen$_{ACC-SG-M}$ the/this/a$_{ACC-SG-M}$ *the/this/a*	Minister minister *minister*
H2-OS (E)*	Die/Diese$_{ACC-PL-F}$ The/These$_{ACC-PL-F}$ *The/This/A*	Autorinnen authors *minister*	empfiehlt$_{SG}$ recommends$_{SG}$ *recommends*	**der/dieser/ein**$_{NOM-SG-M}$ **the/this/a**$_{NOM-SG-M}$ *the/these*	**Minister** **minister** *authors*
H3-OS (NE)*	Den/Diesen/Einen$_{ACC-SG-M}$ The/This/A$_{ACC-SG-M}$ *The/These*	Minister minister *authors*	empfehlen$_{PL}$ recommend$_{PL}$ *recommend*	**die/diese**$_{NOM-PL-F}$ **the/these**$_{NOM-PL-F}$ *the/this/a*	**Autorinnen** **authors** *minister*

Table 2: Two examples of WOGLI premise-hypothesis pairs, one for the pattern `sing_masc_v_sing_masc` and one for the pattern `sing_masc_v_pl_fem`. Underlined words have different surface forms in NE and E hypotheses and carry distinguishing morphological markers of case and/or number. Nouns belonging to the weak declension type are identified by †. Hypotheses H3 are not part of WOGLI proper but will be used in a generalization set called WOGLI-OS-hard as they demand to both process marked OS word order as well as recognising non-entailment in the face of high word overlap. As in the remainder of this paper, hypotheses with a marked word order are identified by an asterisk.

All word lists with GXNLI-train frequencies and translations can be found in our Github repository. Each of the 50 verb types appears between 308 and 383 times (mean: 339.4 times) in the 16,971 premises. They also appear 20 times on average per pattern in the premises. Table 8 in the Appendix gives noun statistics for WOGLI.

4 Experiments on WOGLI

Models. We use two German models and one multilingual BERT model:

- BERT-base[6] is a cased base BERT model pre-trained by the MDZ Digital Library team on 16GB of German-language text.
- GBERT-large[7] is a BERT model pre-trained on 163.4GB of data (Chan et al., 2020), using the same cased vocabulary as BERT-base.[8]
- mBERT-base[9] is a cased BERT model pre-trained on 104 languages (Devlin et al., 2019).

Since models were fine-tuned on GXNLI-train in a three-class setting, we merge contradiction and neutral into non-entailed predictions for evaluations on WOGLI. Fine-tuning details are provided in Section C of the Appendix.

Results (see Table 3). As a sanity check, we first test our models on GXNLI-test. Our models' performances on GXNLI-test are broadly in line with published work. Conneau et al. (2020) achieve an accuracy of 81.2% on GXNLI-test with a monolingual BERT-base model, higher than our 76.67%. However, their model uses a larger vocabulary (40k, ours: 31k) and was pre-trained on a larger corpus (up to 60GB, ours: 16GB). This particular model is unfortunately not available. Other prior work concentrates on multilingual models: GBERT-large's

[6] https://huggingface.co/dbmdz/bert-base-german-cased

[7] https://huggingface.co/deepset/gbert-large

[8] Other large-scale German language models such as GottBERT (Scheible et al., 2020) and GELECTRA (Chan et al., 2020) are of similar size and downstream performance. Thus, we use GBERT-large as a representative.

[9] https://huggingface.co/bert-base-multilingual-cased

size is smaller than mT5-base (580m parameters) by Xue et al. (2021), but its performance of 84.65% on GXNLI-test exceeds the one reported for mT5-base (81.6%). Devlin et al. (2019) achieve an accuracy of 75.9% with mBERT-base (*Translate Train Cased*)[10], in line with ours.

On WOGLI, both base models completely fail, labeling almost all instances as entailments. GBERT-large performs a bit better, suggesting that the language model's scale plays a role in its ability on WOGLI. However, it still shows a strong tendency for the entailment class and the results are not robust across runs. Our vocabulary is frequent and present in GXNLI-train and our sentences have a very simple grammar. Therefore, the models' poor performances on WOGLI suggest that not all German-specific linguistic phenomena are represented in the translated GXNLI-train, similar to our GXNLI word order analysis in Section 2.[11]

5 Error analysis

All analyses in this section are carried out on ensemble predictions (majority vote of the 5 runs) of the strongest model in Table 3, GBERT-large. The ensemble model reaches an accuracy of 57.82% on WOGLI and 27.41% on WOGLI-SO.

5.1 Fluency

We measure the correlation of model performance and linguistic acceptability, approximating the latter via pseudo-loglikelihood (Salazar et al., 2020). WOGLI premises have an average PLL of -30.54 (SD: 8.318). H1-SO (NE) hypotheses have an average PLL of -30.56 (SD: 8.287), while H2-OS (E) hypotheses are less fluent due to marked word order, with an average PLL of -36.53 (SD: 8.535). GBERT-large performs worse on SO (NE) pairs than on the less fluent OS (E) pairs; fluency thus does not play an important role in the model's performance on WOGLI. Instead, the lexical overlap heuristic (Naik et al., 2018; McCoy et al., 2019; Gururangan et al., 2018) is a possible reason for the degradation on non-entailed pairs.

5.2 Performance by subject and object properties

We now focus on WOGLI-SO (NE) only as this is the part of the dataset where the models fail.

Gender. Regarding the gender of arguments in WOGLI, we formulate the following hypothesis:

A1 SO hypotheses with masculine subjects (objects) are easier to classify than the ones with feminine subjects (objects).

A1 can be explained by *(a)* the presence of gender bias due to translation in GXNLI-train (see Section 3) *or (b)* morphological differences between masculine and feminine NPs.

Performance on instances in WOGLI-SO (NE) with masculine common noun subjects is indeed significantly higher than for feminine common noun subjects. The same holds for common noun objects (see also Table 10 in the Appendix). However, this does not transfer to proper names. Gender bias in GXNLI-train *(a)* as an explanation for **A1** is therefore unlikely.

Morphological differences between feminine and masculine NPs *(b)*, however, are a possible explanation for **A1**. Feminine articles and common nouns have the same surface forms in accusative/nominative. Masculine articles and common nouns, however, can bear morphological case markers. The masculine singular articles *der*, *ein* and *dieser* are the only articles in WOGLI to change surface forms in the accusative to *den*, *einen* and *diesen*. Additionally, singular masculine common nouns belonging to the weak declension type also carry case markers. Morphological markers in some masculine NPs could thus be helpful for the model to distinguish subject from object.

Referential properties of subjects/objects. In prefield SO sentences, definite NPs tend to precede indefinite NPs (Weber and Müller, 2004), probably because indefinite constituents are often new and definite constituents are often given (Chafe, 1976). Although XNLI and WOGLI do not contain discourse context, preference for SO sentences with definite before indefinite NPs might be encapsulated in pretraining data. We thus hypothesize that:

A2 SO hypotheses in which a definite NP precedes an indefinite NP are easier to classify.

[10] Results on XNLI are provided in the corresponding GitHub repository: `https://github.com/google-research/bert/blob/master/multilingual.md#results`.

[11] One question that arises is whether even larger models or models pretrained on substantially more data will solve the problem. Other monolingual models for German are sparse. We therefore ran two large, publically available, multilingual model checkpoints fine-tuned on XNLI on WOGLI. They also do not perform well (see Section D in the Appendix).

Evaluation set	BERT-base (110m)	GBERT-large (335m)	mBERT-base (172m)
GXNLI-test	76.65 (0.41)	84.65 (0.163)	75.16 (0.552)
WOGLI	50.16 (0.133)	57.68 (1.86)	50.01 (0.015)
WOGLI-SO (NE)	0.33 (0.269)	27.42 (7.828)	0.02 (0.029)
WOGLI-OS (E)*	100 (0.005)	87.94 (4.171)	100 (0.0)

Table 3: Accuracies for two German and one multilingual model on GXNLI-test and WOGLI, averaged over 5 runs. All are trained on GXNLI-train. Accuracies are computed for 3 classes in GXNLI-test and 2 classes in WOGLI.

We separate WOGLI constituents into definite and indefinite following Prince (1992): definite and demonstrative articles, as well as proper names are markers of definiteness, while indefinite articles point to indefiniteness. We then separate WOGLI-SO (16,971 pairs) into two groups: **preferred** (14,671 pairs) and **dispreferred** (2,300 pairs). Pairs in the **dispreferred** group are opposed to the aforementioned discourse hierarchy in that indefinite constituents precede definite constituents in the SO hypothesis. Pairs in the **preferred** group form the three other possible cases: definite precedes indefinite, definite precedes definite and indefinite precedes indefinite; these cases are not in opposition to the hierarchy.

GBERT-large achieves an accuracy of 29.85% on the SO pairs in the **preferred** group but only 11.78% on the **dispreferred** group (difference significant at 1% significance level, z-test for proportions). Therefore we can confirm **A2**.

Number. Lastly, we analyse the role of verb number agreement in classifying WOGLI-SO (NE) instances. As explained in Section 3, WOGLI patterns either combine only singular arguments (**all-singular**) or a singular and a plural argument (**singular-plural**). Only in the latter group of patterns, subject-verb agreement leads to a change in the verb's surface form from the premise to the H1-SO (NE) hypothesis (see *empfehlen/recommend*$_{PL}$ vs. *empfiehlt/recommends*$_{SG}$ in the second example in Table 2). We investigate the importance of verb number agreement for classifier performance by separating WOGLI-SO (16,971 pairs) into two groups, **all-singular** (4,997 pairs) and **singular-plural** (11,974 pairs).

GBERT-large achieves an accuracy of 36.66% on the SO pairs in the **all-singular** group and 23.54% on the **singular-plural** group (difference significant at 1% significance level, z-test for proportions). Thus the number switch in the verb occurring in **singular-plural** SO hypotheses is not a particularly helpful cue for the classifier.

6 Data augmentation

Following McCoy et al. (2019) and Min et al. (2020) on data augmentation with challenge sets, we hypothesize that augmenting GXNLI-train with a WOGLI subset can be helpful.

We sample 1,037 premises and their corresponding E/NE hypotheses from WOGLI, resulting in 2,074 training instances. Each of the 17 patterns occurs 61 times. All 50 verb lemmas are represented, each appearing between 18 and 25 times. All 181 noun forms appear at least once.[12]

We concatenate these WOGLI instances with GXNLI-train, name the resulting augmented training set GXNLI+1037 and shuffle it before fine-tuning GBERT-large 10 times on this augmented training set. We evaluate on the remaining 31,868 WOGLI instances, named WOGLI-test-1037. This augmented training set allows GBERT-large to classify WOGLI almost perfectly, while maintaining its performance on GXNLI-test (Table 4).

Smaller augmentation size. We fine-tune GBERT-large on a shuffled concatenation of GXNLI-train and only 102 WOGLI premises sampled in a stratified manner from the aforementioned 1,037 premises along with both their corresponding NE and E hypotheses. Each one of the 17 patterns appears 6 times and each one of the 50 verb lemmas appears at least once and at most 4 times. Due to the small augmentation size, it is not possible to ensure representation of all 181 nouns, with 73 not appearing. We evaluate on the remaining 33,738 WOGLI pairs, named WOGLI-test-102. The smaller augmentation size yields a model that performs worse and less robustly on WOGLI test instances (Table 4).

7 Generalization experiments

McCoy et al. (2019) investigate whether augmented models improved by simply memorizing the seen

[12] The nouns occur in varying frequencies due to the small size of the augmentation set.

Evaluation set	GXNLI+1037
GXNLI-test	84.7 (0.301)
WOGLI-test-1037	99.98 (0.016)
WOGLI-SO-test-1037 (NE)	99.99 (0.008)
WOGLI-OS-test-1037 (E)*	99.97 (0.03)

(a) Larger augmentation

Evaluation set	GXNLI+102
GXNLI-test	84.78 (0.244)
WOGLI-test-102	86.04 (4.091)
WOGLI-SO-test-102 (NE)	87.57 (6.428)
WOGLI-OS-test-102 (E)*	84.52 (4.45)

(b) Smaller augmentation

Table 4: Accuracy for GBERT-large fine-tuned on GXNLI-train augmented with WOGLI instances. Results are averaged over 10 runs and computed in a 3-class (GXNLI-test) or a 2-class (WOGLI-test) setting.

templates. To do so, they evaluate them on pairs from unseen patterns. Inspired by this setup, we study the models' generalization capabilities by evaluating them on four new evaluation sets that share structural and lexical similarities with the WOGLI pairs that were seen during fine-tuning.

7.1 Construction of generalization sets

Pronoun subjects: WOGLI-p-subject. We replace the premise subject in WOGLI by a personal pronoun (*He warns the client*). Correspondingly, the H1-SO (NE) hypothesis then has the pronoun as the object (*The client warns him*) whereas the entailed H2-OS (E) hypothesis just swaps word order with regards to the premise (*The client warns he*) (see also Table 11 in the Appendix). To focus on the pronominalization change, the same 17 patterns, verb lemmas, proper nouns and common nouns are also used in WOGLI-p-subject. In addition to the previously mentioned morphological markers of case and/or verb number occurring in WOGLI sentences (Section 3), the masculine singular pronoun *er/he* (nominative) in WOGLI-p-subject changes surface form in the accusative case (*ihn/him*). Feminine and plural pronouns (*sie/she/her/they/them*) in WOGLI-p-subject, however, do not change surface form. Some WOGLI premises can become duplicates after replacing the subject by a personal pronoun. Consider the two premises *Die Ärzte warnen den Kunden/The doctors*$_{masc}$ *warn the client* and *Die Ärztinnen warnen den Kunden/The doctors*$_{fem}$ *warn the client*. After replacing the subject, both premises lead to the new premise *Sie warnen den Gast/They warn the guest*, since plural masculine and plural feminine nominative personal pronouns have the same surface form in German. We keep only one version for such duplicates. The new generalization set contains 13,802 unique premises, or a total of 27,604 pairs.

Dative: WOGLI-dative. We collect a new list of 22 transitive verbs that require dative instead of accusative objects. All verbs are not symmetric, which ensures that NE hypotheses always have the correct gold label. Each verb lemma appears at least 17 times in GXNLI-train. We use the same 144 noun types as in WOGLI to generate new instances. The premises again have SVO structure, and H1 has SO (NE) and H2 has OS (E) structure. Therefore the instances are completely parallel to WOGLI apart from the case of the object.

In these dative constructions, 24 patterns are possible (Table 6 in the Appendix). Each pattern appears 150 times in WOGLI-dative and each verb lemma appears between 132 and 182 times in the premises. All possible noun surface forms appear between 6 and 81 times in the premises. We generate 3,600 premises, or 7,200 pairs in total. Table 12 in the Appendix shows an example. In WOGLI-dative, all determiners (singular and plural, feminine and masculine) change surface forms by case. Additionally, plural masculine common nouns change surface forms in the dative if they do not end with -n in the nominative[13]. As in WOGLI, singular masculine nouns of the weak declension type and verbs in singular-plural patterns also change surface forms.

Ditransitive verbs. We collect 21 ditransitive verbs (such as *schicken/send* and *verheimlichen/conceal*), each of which appears at least 6 times in GXNLI-train. Verbs are grouped into 5 semantic categories (`giving, taking, sending, communication, secret`). Subjects and indirect objects of the verbs are compatible with the semantic class human, so that we can reuse the 144 noun types from WOGLI.

[13] Masculine nouns ending with -n in plural nominative are: *Kunden/clients*, *Professoren/professors*, *Studenten/students*, *Mentoren/mentors*, *Patienten/patients*, *Soldaten/soldiers*, *Journalisten/journalists*, *Zeugen/witnesses*. These nouns maintain the same surface forms in plural dative.

For direct objects, we use a new list of 54 common nouns, appearing at least 15 times in GXNLI-train. They are grouped with the verb semantic categories so that resulting premises/hypotheses are meaningful (thus, you can combine the direct object *Identität/identity* with `secret` verbs but not with `sending` verbs). The direct object is always preceded by a definite article.

Ditransitive premises follow the preferred word order SiO: subject-verb-IndirectObject-DirectObject (*The waitresses give the merchant the cake*). Very similar to WOGLI, the not-entailed H1 hypothesis swaps the underlying arguments of subject and indirect object, adapting case and number (*The merchant gives the waitresses the cake*) whereas the entailed H2 hypothesis reorders subject and indirect object into the marked iOS word order without changing the meaning by keeping case and number markers intact (*The merchant give the waitresses the cake*). The direct object is not affected. An example is shown in Table 13 in the Appendix. With respect to morphological markers, WOGLI-ditransitive follows the same surface form changes between E and NE hypotheses as WOGLI-dative, since indirect objects in WOGLI-ditransitive are in dative case.

Ditransitives allow for 24 unique patterns. We allow each pattern to appear 1,000 times leading to 12,000 premises or 24,000 pairs.

WOGLI-OS-hard (NE). Neither WOGLI nor the previous generalization datasets contain instances where the marked OS word order leads to non-entailment, i.e. where you have to recognise non-entailment in the face of high word overlap while at the same time processing a rare word order. Therefore we create a third hypothesis H3-OS (NE) for each WOGLI premise where we similar to H1 invert the underlying arguments but present this changed meaning in OS word order. Two examples are given as H3-OS (NE) in Table 2. This is possible for all 17 WOGLI patterns. Pairing H3 with each WOGLI premise leads to 16,971 new non-entailed pairs. All premises as well as all lexical items have been seen in normal WOGLI.

7.2 Generalization results

We evaluate GBERT-large fine-tuned on GXNLI without augmentation (GXNLI+0) as well as fine-tuned on GXNLI+1037 and on GXNLI+102 on our four generalization sets. Results are in Table 5.

GXNLI+1037 transfers very well to WOGLI-p-subject, while GXNLI+102 reaches an accuracy of only 59.03% on SO instances and is less robust. Thus, even for simple pronoun replacement a relatively large augmentation size is needed. A similar picture emerges for WOGLI-dative. Since WOGLI-dative contains more patterns than WOGLI, we investigate whether GXNLI+102's poor performance is only observable in patterns that were not seen during fine-tuning but find no preference for seen or unseen patterns.

With respect to ditransitive pairs, GXNLI+1037 has almost perfect accuracy and GXNLI+102 reaches its best generalization set performance, reaching similar results as on standard WOGLI.

We hypothesized that generalization to H3-OS (NE) in WOGLI-OS-hard is the most difficult as it contains both marked word order and non-entailment, whereas (i) in GXNLI, the marked word order is very rare (see Section 2) and (ii) in WOGLI, the marked word order has always been seen with the entailment class, potentially tripping up an augmented model that could have learnt this hypothesis-only fact. This turns out to be true: GXNLI+0 classifies basically all WOGLI-OS-hard (NE) examples wrongly as entailment and performs even worse than the same model on the original WOGLI-SO (NE) non-entailed examples (see the 27.42% in Table 3). With substantial augmentation (GXNLI+1037), performance is slightly better but the results are still both very low and unstable.

Our generalization experiments show that (i) the augmentation set needs to be sufficiently large for successful generalization to new NLI pairs that are structurally similar to WOGLI and (ii) models exposed to WOGLI do not necessarily generalize well to some related datasets at all. As German word order is quite intricate and will have additional variations for embedded or non-declarative clauses this means training datasets need to be very large and varied to learn German word order.

8 Related work

Many English adversarial NLI datasets have been proposed. Some of these (Dasgupta et al., 2018; Kim et al., 2018; Nie et al., 2019; McCoy et al., 2019), like us, include minimal pairs with a high word overlap between premise and hypotheses. Kim et al. (2018), for example, change argument order to generate non-entailments so that "understanding" word order is necessary to solve these. However, in WOGLI, changes in argument order

Evaluation set	GXNLI+0	GXNLI+102	GXNLI+1037
WOGLI-p-subject-test	53.23 (1.715)	77.97 (6.653)	98.89 (0.957)
WOGLI-p-subject-SO-test (NE)	7.34 (4.16)	59.03 (13.353)	97.78 (1.916)
WOGLI-p-subject-OS-test (E)*	99.12 (0.74)	96.91 (1.822)	99.99 (0.008)
WOGLI-dative-test	58.11 (2.789)	79.4 (5.446)	94.72 (0.565)
WOGLI-dative-SO-test (NE)	17.76 (6.223)	60.87 (11.227)	91.49 (1.421)
WOGLI-dative-OS-test (E)*	98.47 (0.776)	97.93 (0.534)	97.96 (0.504)
WOGLI-ditransitive-test	73.93 (6.327)	92.59 (4.634)	99.58 (0.143)
WOGLI-ditransitive-SiO-test (NE)	50.23 (13.11)	86.55 (9.41)	99.62 (0.261)
WOGLI-ditransitive-iOS-test (E)*	97.63 (0.635)	98.63 (0.591)	99.55 (0.276)
WOGLI-OS-hard (NE)*	0.15 (0.082)	0.77 (0.75)	23.45 (15.985)

Table 5: Accuracy on generalization sets, averaged over 5 runs for GXNLI+0 and over 10 runs for remaining models

generate entailed *and* non-entailed hypotheses, depending on keeping or changing corresponding morphology. The more fixed English word order does not allow for flexibility to that degree.

Regarding adversarial NLI datasets for German, Hartmann et al. (2021) investigate negation but do not work on word order. Tikhonova et al. (2022) propose NLI diagnostic datasets for French, German and Swedish. Sentence pairs are manually translated from the Russian TERRa dataset (Shavrina et al., 2020) as well as from the diagnostic dataset of GLUE (Wang et al., 2018). We inspected a random 100 hypotheses of the German TERRa dataset, none of which were in marked word order. The translated GLUE benchmark is annotated with linguistic features relevant for entailment such as lexical semantics, logic, and predicate-argument structure. Only the predicate-argument structure examples include a handful where word order of arguments has been inverted between premise and hypothesis. However, resulting hypotheses were often ambiguous and — in our opinion — wrongly annotated as not-entailed. Consider the premise *John zerbrach das Fenster/John broke the window* and the hypothesis *Das Fenster hat John eingeschlagen*, which is ambiguous between *The window*$_{NOM}$ *broke John*$_{ACC}$ (SO order, NE) OR *The window*$_{ACC}$ *broke John*$_{NOM}$ (OS order, E). This is annotated as non-entailment in the dataset, assuming SO order with an implausible semantic reading, whereas the marked word order with a plausible semantic reading leads to entailment.

Unlike us, both datasets do not emphasise word order. They are also based on translations and therefore rarely contain OS hypotheses.

9 Conclusion

We created WOGLI, a new NLI challenge set, in order to examine the challenges brought by the freer German word order. Premises, entailed and not-entailed hypotheses contain exactly the same lemmata; the two hypotheses differ only in word order and morphological changes but change label. Three current BERT-based models fine-tuned on GXNLI-train struggle on WOGLI pairs. This poor performance mirrors the fact that translated NLI training sets such as GXNLI do not incorporate all required linguistic phenomena that are specific to the target language, German. We find that the number of WOGLI pairs for augmentation during fine-tuning must be sufficiently high in order to (i) learn WOGLI and (ii) generalize to other WOGLI-like pairs. Even with a larger augmentation set and a large pretrained model, a generalization set that differs more from WOGLI, such as WOGLI-OS-hard (NE) , remains difficult.

In future experiments, we will expand WOGLI datasets to contain additional variation, such as tense variation, more complex sentence structure (additional arguments and adjuncts, active/passive), more complex constituent structure and other sentence types (non-declarative, embedded). This will also allow us to conduct more fine-grained error analyses regarding the hierarchies that influence the linearization of arguments and thus word order.

Acknowledgments

Ines Reinig is funded by the German Research Foundation (DFG) under the UNCOVER project (PO 1900/7-1 and RE 3536/3-1).

References

Markus Bader, Emilia Ellsiepen, Vasiliki Koukoulioti, and Yvonne Portele. 2017. Filling the prefield: Findings and challenges. In *DGfS 2016 workshop "V2 in grammar and processing: Its causes and its consequences"*, pages 27–49.

Markus Bader and Jana Häussler. 2010. Word order in German: A corpus study. *Lingua*, 120(3):717–762.

Markus Bader and Yvonne Portele. 2019. Givenness and the licensing of object-first order in German: The effect of referential form. In *Proceedings of Linguistic Evidence 2018: Experimental Data Drives Linguistic Theory*. Universität Tübingen.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wallace Chafe. 1976. *Givenness, constrastiveness, definiteness, subjects, topics, and point of view*, pages 25–55. Academic Press, New York.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. In *Proceedings of the Fortieth Annual Conference of the Cognitive Science Society*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Erich Drach. 1937. *Grundgedanken der Deutschen Satzlehre*. M. Diesterweg.

Christa Dürscheid. 2012. *Syntax: Grundlagen und Theorien*, volume 6. Vandenhoeck & Ruprecht.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Mareike Hartmann, Miryam de Lhoneux, Daniel Hershcovich, Yova Kementchedjhieva, Lukas Nielsen, Chen Qiu, and Anders Søgaard. 2021. A multilingual benchmark for probing negation-awareness with minimal pairs. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 244–257, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Juho Kim, Christopher Malon, and Asim Kadav. 2018. Teaching syntax by adversarial distraction. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of NLI models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874.

Ellen F Prince. 1992. The ZPG letter: Subjects, definiteness, and information-status. *Discourse description: diverse analyses of a fund raising text*, pages 295–325.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. GottBERT: a pure German language model. *arXiv preprint arXiv:2012.02110*.

Tatiana Shavrina, Alena Fenogenova, Emelyanov Anton, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. RussianSuperGLUE: A Russian language understanding evaluation benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4717–4726, Online. Association for Computational Linguistics.

Anna Siewierska. 1993. On the interplay of factors in the determination of word order. In Joachim Jacobs, Arnim von Stechow, Wolfgang Sternefeld, and Theo Vennemann, editors, *An International Handbook of Contemporary Research*, pages 826–846. De Gruyter Mouton, Berlin • New York.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. *arXiv preprint arXiv:2205.11503*.

Maria Tikhonova, Vladislav Mikhailov, Dina Pisarevskaya, Valentin Malykh, and Tatiana Shavrina. 2022. Ad astra or astray: Exploring linguistic knowledge of multilingual BERT through NLI task. *Natural Language Engineering*, pages 1–30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Andrea Weber and Karin Müller. 2004. Word order variation in German main clauses: A corpus analysis. In *20th International Conference on Computational Linguistics*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

A SVO patterns

Table 6 lists the patterns that we use to build WOGLI and generalization sets. Table 7 lists the 8 patterns that we exclude from WOGLI. The noun phrases in the SO and OS form have the same morphological surface form and the verb also has the same form in both word orders. Therefore, SO and OS meaning are not distinguishable.

B WOGLI statistics

Table 8 shows counts and ratios for the subject and object roles in WOGLI. The different noun categories shown in the first column generally take the subject role as often as they take the object role.

C Fine-tuning details

The input sequence, consisting of the premise-hypothesis pair, is encoded using the given BERT model. The final hidden state of the special `[CLS]` token constitutes the aggregate representation of the input sequence, following Devlin et al. (2019). This representation is then passed through a dropout layer and a linear classification layer, which maps it to the three-label classification space. All models were fine-tuned for three epochs, with linear warmup over 6% of the first steps and a maximum sequence length of 128. BERT-base and mBERT-base were fine-tuned with a batch size of 16 and a learning rate of $5e^{-5}$. GBERT-large was fine-tuned with a batch size of 32 and a learning rate of $5e^{-6}$. Regarding mBERT-base, fine-tuning on the translated training set is comparable to the *Translate Train* setup in Conneau et al. (2018). We also experimented with fine-tuning mBERT-base on the English MNLI training set, similarly to the *Zero Shot* setup in Conneau et al. (2018), but found better validation set accuracy using the translated training set. GBERT-large was fine-tuned on one NVIDIA A6000 GPU. Base models were fine-tuned on one NVIDIA T4 GPU. Fine-tuning took approximately 2 hours per run.

D WOGLI results for other models

Table 9 shows results on WOGLI for publically available checkpoints (single runs) of two larger multilingual models: XLM-RoBERTa-large[14] (Conneau et al., 2020) (XLM-R) and the generative encoder-decoder model mT5-large[15] (Xue et al., 2021). These two models have considerably more parameters than GBERT-large. According to the respective model cards:

- XLM-R was fine-tuned on the concatenation of the MNLI training set and the XNLI validation and test sets.
- mT5-large was fine-tuned on the MNLI and the XTREME XNLI[16] (Hu et al., 2020) training sets.

Both large models perform much better than the two base models (see Table 3 in the paper), which suggests again that model scale is relevant on WOGLI. However, they do not achieve higher overall accuracies than GBERT-large (average: 57.68%). Interestingly, mT5-large performs best on WOGLI-SO, but struggles substantially on WOGLI-OS, often labeling these pairs as non-entailments.

ChatGPT: discussion. In a small-scale experiment, we evaluate the ability of the recently made available research preview for the chatbot ChatGPT (February 13 version) by OpenAI[17] on WOGLI. This chatbot is based on the autoregressive GPT-3 model (Brown et al., 2020), as opposed to autoencoding models such as BERT, and has recently drawn a lot of attention in the AI community. We attempted to obtain classifications from ChatGPT on a WOGLI subset consisting of of 51 WOGLI-SO and 51 WOGLI-OS pairs. However, we observed (i) a strong prompt-dependence (Suzgun et al., 2022), as even minor changes in the prompt's phrasing lead to different answers by the chatbot and (ii) overall inconsistent results across multiple instances of showing the model the same sets of pairs. Due to the inconsistency of these preliminary results, we leave it to future work to assess ChatGPT's capabilities on WOGLI in a more systematic manner and for a range of different prompt styles.

E Error analysis: performance by gender

Table 10 provides more detailed results for the analysis discussed in Section 5.2.

[14] `https://huggingface.co/joeddav/xlm-roberta-large-xnli`

[15] `https://huggingface.co/alan-turing-institute/mt5-large-finetuned-mnli-xtreme-xnli`

[16] This version of the XNLI dataset contains different machine translations than the original XNLI dataset: `https://www.tensorflow.org/datasets/catalog/xtreme_xnli`

[17] `https://openai.com/blog/chatgpt`

Dative, Ditransitive	WOGLI, WOGLI-OS-hard (NE)
pnoun_v_sing_masc	pnoun_v_sing_masc
pnoun_v_plural_masc	pnoun_v_plural_masc
pnoun_v_plural_fem	pnoun_v_plural_fem
pnoun_v_sing_fem	
plural_masc_v_pnoun	plural_masc_v_pnoun
plural_masc_v_sing_masc	plural_masc_v_sing_masc
plural_masc_v_sing_fem	plural_masc_v_sing_fem
plural_masc_v_plural_fem	
plural_masc_v_plural_masc	
plural_fem_v_sing_masc	plural_fem_v_sing_masc
plural_fem_v_sing_fem	plural_fem_v_sing_fem
plural_fem_v_pnoun	plural_fem_v_pnoun
plural_fem_v_plural_fem	
plural_fem_v_plural_masc	
sing_masc_v_sing_masc	sing_masc_v_sing_masc
sing_masc_v_plural_masc	sing_masc_v_plural_masc
sing_masc_v_plural_fem	sing_masc_v_plural_fem
sing_masc_v_sing_fem	sing_masc_v_sing_fem
sing_masc_v_pnoun	sing_masc_v_pnoun
sing_fem_v_sing_masc	sing_fem_v_sing_masc
sing_fem_v_plural_fem	sing_fem_v_plural_fem
sing_fem_v_plural_masc	sing_fem_v_plural_masc
sing_fem_v_pnoun	
sing_fem_v_sing_fem	

Table 6: Exhaustive list of patterns used to build WOGLI-dative, WOGLI-ditransitive (24 patterns) and WOGLI (17 patterns). WOGLI-OS-hard (NE) uses the same patterns as WOGLI.

F Generalization sets

Tables 11, 12 and 13 provide examples for pairs created for generalization sets.

Pattern	Premise	Hypothesis	Label
`sing_fem_v_pnoun`	Die Freundin begrüßt David . The friend$_{CASE?-SING-FEM}$ greets David$_{CASE?-SING-MASC}$	David begrüßt die Freundin . David$_{CASE?-SING-MASC}$ greets the friend$_{CASE?-SING-FEM}$	?
`pnoun_v_sing_fem`	David begrüßt die Freundin . David$_{CASE?-SING-MASC}$ greets the friend$_{CASE?-SING-FEM}$	Die Freundin begrüßt David . The friend$_{CASE?-SING-FEM}$ greets David$_{CASE?-SING-MASC}$	?
`pnoun_v_pnoun`	Walter begrüßt David . Walter$_{CASE?-SING-MASC}$ greets David$_{CASE?-SING-MASC}$	David begrüßt Walter . David$_{CASE?-SING-MASC}$ greets Walter$_{CASE?-SING-MASC}$	?
`sing_fem_v_sing_fem`	Die Mitbewohnerin begrüßt die Freundin . The flatmate$_{CASE?-SING-FEM}$ greets the friend$_{CASE?-SING-FEM}$	Die Freundin begrüßt die Mitbewohnerin . The friend$_{CASE?-SING-FEM}$ greets the flatmate$_{CASE?-SING-FEM}$	?
`plural_fem_v_plural_fem`	Die Freundinnen begrüßen die Mitbewohnerinnen . The friends$_{CASE?-PL-FEM}$ greet the flatmates$_{CASE?-PL-FEM}$	Die Mitbewohnerinnen begrüßen die Freundinnen . The flatmates$_{CASE?-PL-FEM}$ greet the friends$_{CASE?-PL-FEM}$	?
`plural_masc_v_plural_masc`	Die Freunde begrüßen die Mitbewohner . The friends$_{CASE?-PL-MASC}$ greet the flatmates$_{CASE?-PL-MASC}$	Die Mitbewohner begrüßen die Freunde . The flatmates$_{CASE?-PL-MASC}$ greet the friends$_{CASE?-PL-MASC}$	?
`plural_masc_v_plural_fem`	Die Freunde begrüßen die Mitbewohnerinnen . The friends$_{CASE?-PL-MASC}$ greet the flatmates$_{CASE?-PL-FEM}$	Die Mitbewohnerinnen begrüßen die Freunde . The flatmates$_{CASE?-PL-FEM}$ greet the friends$_{CASE?-PL-MASC}$	?
`plural_fem_v_plural_masc`	Die Freundinnen begrüßen die Mitbewohner . The friends$_{CASE?-PL-FEM}$ greet the flatmates$_{CASE?-PL-MASC}$	Die Mitbewohner begrüßen die Freundinnen . The flatmates$_{CASE?-PL-MASC}$ greet the friends$_{CASE?-PL-FEM}$	?

Table 7: The 8 patterns that we exclude from WOGLI and WOGLI-OS-hard (NE)

Noun	# types	Subject count			Object count			Subject/Object ratio			
		Mean	Min	Max	Mean	Min	Max	Mean	Median	Min	Max
Masc pnoun	41	36.8 (4.22)	27	49	36.5 (6.89)	25	53	1.05	1.0	0.58	1.72
Fem pnoun	41	36.3 (5.71)	27	52	36.5 (6.18)	23	49	1.02	1.0	0.63	1.57
Masc sing. cnoun	38	131.4 (12.23)	108	163	131.5 (10.38)	114	158	1.01	1.0	0.78	1.38
Masc pl. cnoun	38	78.7 (9.23)	60	97	78.8 (8.86)	53	99	1.01	1.01	0.75	1.36
Fem sing. cnoun	24	124.8 (11.05)	103	143	124.7 (7.90)	109	137	1.01	1.0	0.81	1.25
Fem pl. cnoun	24	124.7 (12.08)	100	144	124.8 (12.40)	101	145	1.01	0.98	0.77	1.29

Table 8: Average counts and subject to object ratios for different groups of nouns in WOGLI. For example, masculine proper nouns are subjects 36.8 times and objects 36.5 times on average. Values in parentheses are standard deviations.

Evaluation set	XLM-R (550m)	mT5-large (1.2b)
WOGLI	55.42	52.26
WOGLI-SO (NE)	46.2	68.7
WOGLI-OS (E)*	64.64	35.82

Table 9: Accuracies on WOGLI for two larger multilingual models. Results are for single runs.

Constituent	Argument		Conditional probability		Signif.
	Premise	Hypo (SO)	Definition	Value (%)	
Common noun	Subject	Object	$p(\text{correct} \mid \text{SO, m. cnoun psubj})$	28.17	95%
			$p(\text{correct} \mid \text{SO, f. cnoun psubj})$	26.22	
Common noun	Object	Subject	$p(\text{correct} \mid \text{SO, m. cnoun pobj})$	33.11	99%
			$p(\text{correct} \mid \text{SO, f. cnoun pobj})$	22.42	
Proper noun	Subject	Object	$p(\text{correct} \mid \text{SO, m. pnoun psubj})$	27.50	n.s.
			$p(\text{correct} \mid \text{SO, f. pnoun psubj})$	27.98	
Proper noun	Object	Subject	$p(\text{correct} \mid \text{SO, m. pnoun pobj})$	23.25	n.s.
			$p(\text{correct} \mid \text{SO, f. pnoun pobj})$	21.06	

Table 10: Conditional probabilities for the correctness of predictions given the subject's or the object's gender. The rightmost column indicates a significant difference between compared proportions at the 99% or 95% confidence level, or no significance (n.s.), using a z-test for the equality of two proportions.

Premise	**Er**$_{NOM-SG-M}$	warnt$_{SG}$	den$_{ACC-SG-M}$	Gast
	He$_{NOM-SG-M}$	warns$_{SG}$	the$_{ACC-SG-M}$	guest
	He	*warns*	*the*	*guest*
H1-SO (NE)	**Der**$_{NOM-SG-M}$	**Gast**	warnt$_{SG}$	ihn$_{ACC-SG-M}$
	The$_{NOM-SG-M}$	**guest**	warns$_{SG}$	him$_{ACC-SG-M}$
	The	*guest*	*warns*	*him*
H2-OS (E)*	Den$_{ACC-SG-M}$	Gast	warnt$_{SG}$	**er**$_{NOM-SG-M}$
	The$_{ACC-SG-M}$	guest	warns$_{SG}$	**he**$_{NOM-SG-M}$
	He	*warns*	*the*	*guest*

Table 11: Examples of WOGLI-p-subject pairs. Just as in WOGLI, the entailed hypothesis has a marked word order.

Premise	**Ein**$_{NOM-SG-M}$	**Richter**	gratuliert$_{SG}$	diesen$_{DAT-PL-M}$	Beratern
	A$_{NOM-SG-M}$	**judge**	congratulates$_{SG}$	these$_{DAT-PL-M}$	consultants
	A	*judge*	*congratulates*	*these*	*consultants*
H1-SO (NE)	**Diese**$_{NOM-PL-M}$	**Berater**	gratulieren$_{PL}$	einem$_{DAT-SG-M}$	Richter
	These$_{NOM-PL-M}$	**consultants**	congratulate$_{PL}$	a$_{DAT-SG-M}$	judge
	The	*consultants*	*congratulate*	*a*	*judge*
H2-OS (E)*	Diesen$_{DAT-PL-M}$	Beratern	gratuliert$_{SG}$	**ein**$_{NOM-SG-M}$	**Richter**
	These$_{DAT-PL-M}$	consultants	congratulates$_{SG}$	**a**$_{NOM-SG-M}$	**judge**
	A	*judge*	*congratulates*	*these*	*consultants*

Table 12: Examples of WOGLI-dative pairs. Just as in WOGLI, the entailed hypothesis has a marked word order.

Premise	**Die**$_{NOM-PL-F}$	**Kellnerinnen**	geben$_{PL}$	einem$_{DAT-SG-M}$	Händler	den	Kuchen
	The$_{NOM-PL-F}$	**waitresses**	give$_{PL}$	a$_{DAT-SG-M}$	merchant	the	cake
	The	*waitresses*	*give*	*the*	*cake*	*to a*	*merchant*
H1-SiO (NE)	**Ein**$_{NOM-SG-M}$	**Händler**	gibt$_{SG}$	den$_{DAT-PL-SG}$	Kellnerinnen	den	Kuchen
	A$_{NOM-SG-M}$	**merchant**	gives$_{SG}$	the$_{DAT-PL-SG}$	waitresses	the	cake
	A	*merchant*	*gives*	*the*	*cake*	*to the*	*waitresses*
H2-iOS (E)*	Einem$_{DAT-SG-M}$	Händler	geben$_{PL}$	**die**$_{NOM-PL-F}$	**Kellnerinnen**	den	Kuchen
	A$_{DAT-SG-M}$	merchant	give$_{PL}$	**the**$_{NOM-PL-F}$	**waitresses**	the	cake
	The	*waitresses*	*give*	*the*	*cake*	*to a*	*merchant*

Table 13: Examples of WOGLI-ditransitive pairs. Just as in WOGLI, the entailed hypothesis has a marked word order.

Contextual Variability Depends on Categorical Specificity rather than Conceptual Concreteness: A Distributional Investigation on Italian data

Giulia Rambelli
University of Bologna
giulia.rambelli4@unibo.it

Marianna M. Bolognesi
University of Bologna
m.bolognesi@unibo.it

Abstract

A large amount of literature on conceptual abstraction has investigated the differences in contextual distribution (namely *contextual variability*) between abstract and concrete concept words (*joy* vs. *apple*), showing that abstract words tend to be used in a wide variety of linguistic contexts. In contrast, concrete words usually occur in a few very similar contexts. However, these studies do not take into account another process that affects both abstract and concrete concepts alike: *specificity*, that is, how inclusive a category is (*ragdoll* vs. *mammal*). We argue that the more a word is specific, the more its usage is tied to specific domains, and therefore its contextual variability is more limited compared to generic words.

In this work, we used distributional semantic models to model the interplay between contextual variability measures and i) concreteness, ii) specificity, and iii) the interaction between the two variables. Distributional analyses on 662 Italian nouns showed that contextual variability is mainly explainable in terms of specificity or by the interaction between concreteness and specificity[1]. In particular, the more specific a word is, the more its contexts will be close to it. In contrast, generic words have less related contexts, regardless of whether they are concrete or abstract.

1 Introduction

In the study of lexical semantic representation, an extensive debate focuses on explaining the differences between words referring to concrete and abstract concepts. According to the *Dual Coding Theory* (Paivio, 1991), concrete words are represented in two different systems, one language-based and one image-based, while abstract words are primarily or exclusively represented in the former system. The *Context Availability Hypothesis* (Schwanenflugel, 2013) instead argues that all word meanings are represented in a single verbal code, but concrete words have stronger and denser associations to contextual knowledge than abstract ones. Both theories agree on two points: i) the meaning of abstract words is essentially acquired via language, for instance, through distributional statistics extracted from the linguistic input, and ii) concrete words are "semantically richer" than abstract ones, thereby explaining their processing advantage, the so-called *concreteness effect* (Jessen et al., 2000).

The investigation of the distributional properties of concrete and abstract concepts and words has taken different paths, implementing different metrics to measure how words behave in context (see Section 2.1). We hereby use the general term *contextual variability* (Hoffman, 2016) as an 'umbrella' that includes all proposed metrics of contextual behaviors, described in the next section. Overall, the previous works on contextual variability showed that words referring to concrete concepts occur in a few but very similar syntagmatic contexts, depending on the fact that their meanings are tied to a fixed class of objects or events in the environment. On the other hand, abstract concepts are characterized by a greater degree of variability across contexts, commonly attributed to their association with less well-defined, intangible experiences or properties.

Notwithstanding, it is worth noting that prior investigations have mainly focused on the divergence between concrete and abstract concepts, while potentially overlooking any discrepancies in specificity, that is, the level of inclusivity in the referential category. This can be problematic because it may lead to comparisons between very specific concrete concepts like *muffler* and very generic abstract concepts like *manner*, or very generic concrete concepts like *substance* and very specific abstract concepts like *sorrow*. Crucially, generic and specific

[1] Data available at https://osf.io/2qm5e/?view_only=fce6b4bb895a41658ed97512afa65ae3

Proceedings of the 15th International Conference on Computational Semantics pages 16–27
June 21–23, 2023. ©2023 Association for Computational Linguistics

words may have different contextual distributions: specific words may tend to be used in limited sets of contexts because they denote precise entities occurring in texts characterized by high-resolution semantics. Conversely, generic words may be used in a wider range of diverse contexts because they are less precise and, therefore, more easily applicable to different contexts; generic words may occur in texts characterized by low-resolution semantics and, therefore, may occur with a wider range of shallowly-related contexts.

With the present study, we tackle the following questions:

- How does concreteness explain the variation in contextual distributions of nouns?
- How does specificity explain the variation in contextual distributions of nouns?
- How does the interaction between concreteness and specificity explain the variation in contextual distributions of nouns?

These questions are addressed through a series of regression studies in which the concreteness ratings Montefinese et al. (2014) and specificity ratings Bolognesi and Caselli (2022) of 662 Italian nouns are modeled with a set of corpus-based indices representing their context variability.

2 Related works

2.1 Operationalizations of Contextual Variability

When investigating how concrete and abstract concepts are processed in the mind, researchers have endeavored to relate such differences to the differences between the contexts of occurrence (a.k.a. *contextual variability*) of concrete and abstract words (Hoffman, 2016, for a review).

The Context Availability hypothesis, for instance, notes that concrete words tend to have more robust and intricate contextual associations than abstract ones. This notion is supported by Schwanenflugel and Shoben (1983) 's early research, which found that speakers find it easier to imagine a context for concrete words compared to abstract words. Schwanenflugel et al. demonstrated that when an explicit context was provided for concrete and for abstract words alike, the processing advantage of concrete over abstract words disappeared. The authors concluded that abstract words were more difficult to process because participants struggled to place them in a meaningful context, but this difficulty was reduced when an explicit context was provided.

Hoffman et al. (2013) employed the term *semantic diversity* to describe the average similarity between the contexts in which a word appears. They discovered that concrete words are used in a limited, closely interconnected set of contexts. For instance, the term ”spinach” typically occurs only in contexts related to cooking and eating which are similar to one another. On the other hand, abstract words (e.g., ”life”) are used in a more diverse range of unrelated contexts, resulting in high semantic diversity values. Moreover, Recchia and Jones (2012) introduced two contextual measures related to abstract and concrete concepts. The first measure, *contextual dispersion* (CD), refers to the number of different content areas (or domains) in which a word appears, as proposed by Pexman et al. (2008). The second measure is the *number of semantic neighbors* (NSN), which measures the number of words that appear within a particular radius of a high-dimensional semantic space. The authors found that NSN is higher for abstract than for concrete words, and this peculiarity facilitated the processing of abstract concepts in lexical decision tasks.

Overall, cognitive studies tend to indicate that abstract words are more likely to be used in a wider variety of linguistic contexts, shallowly related to the target word. Concrete words tend to be used in tighter networks of similar contexts, and this may facilitate their retrieval.

2.2 Computational Models of Abstraction

In the last decade, several computational models have been suggested to automatically validate the cognitive assumptions about the contextual difference between abstract and concrete concepts.

Similarly to Recchia and Jones (2012), Hill et al. (2014) quantitatively analyzed the different patterns of association for words varying in concreteness, providing possible cognitive underpinnings for the differences observed. The authors showed that abstract concepts occur in a broader range of contexts and are organized according to associative principles; concrete concepts instead have few specific contexts of occurrence, and they tend to be organized according to (semantic) similarity principles. Recently, Frassinelli et al. (2017) investigated the degree of concreteness of co-occurring contexts for concrete and abstract English words. They built a vector space model for nouns from the Brysbaert et al. (2014) concreteness norms; to retain concreteness scores of contexts and distributional neighbors, they restricted the vocabulary to nouns attested in the dataset (that is, they built a symmetric co-occurrence matrix in which all targets and context words are from concreteness norms). The authors reported that the more a noun is concrete, the more it tends to appear with other concrete nouns and has a more extensive range of concreteness scores; on the contrary, the more a word is abstract, the more it occurs with other abstract words. While this outcome aligns with multiple studies in the literature, the methodological choice of restricting the number of contexts to the words attested in Brysbaert et al. (2014) may have biased the actual distributional pattern of these words.

Working on Italian, Lenci et al. (2018) observed that abstract words, which according to some studies tend to be characterized by a heavier emotional load compared to concrete words (Vigliocco et al., 2014, i.a.) tend also to co-occur with contexts with an overall higher emotive load. This has been observed based on affective statistical indices estimated as distributional similarity with a restricted number of seed words strongly associated with a set of basic emotions. This study provides additional empirical evidence to support the tendency for more concrete words to be associated with higher contextual richness. Overall, previous studies indicated that concrete words tend to have less diverse but more compact and strongly associated distributional neighbors than abstract words.

While a variety of computational models have been focusing on the contextual properties of concrete and abstract words, there are virtually no computational models focused on the contextual variability of specific and generic words due to the challenges associated with comparing these two variables. One major obstacle is the lack of human ratings available for measuring specificity. Notably, Schulte im Walde and Frassinelli (2022) offer a unique exception to this trend. The authors tested how various distributional measures represent abstract-concrete and general-specific word pairs (represented as hypernym-hyponym pairs from WordNet, Miller and Fellbaum (1991)). Analyses revealed that the distributional similarity of contextual words surrounding a target (i.e., neighborhood density) predicts word concreteness: the higher the similarity, the more concrete the word tends to be, albeit this effect is more pronounced for nouns than for verbs. Nevertheless, this measure is not useful for correctly predicting the specificity of a word, which depends on frequency and word entropy. To the best of our knowledge, they are the first to include both Concreteness and Specificity in this type of investigation. However, there are two limits to this approach. First, as mentioned above, they operationalized word specificity as a binary property (rather than a continuous variable) extracted from WordNet. Arguably, such binary distinction does not capture the fine-grained information encoded in a continuous variable. In a second stance, the authors keep concreteness and specificity separated without considering the interaction between the two variables in relation to their context variability.

3 Materials and Methods

3.1 Concreteness and Specificity datasets

For our study, we employed the Bolognesi and Caselli (2022) dataset (henceforth, **BC**), a collection of human-generated specificity ratings for 1049 Italian words. Specificity ratings were collected online adopting the Best-Worst Scaling method (Louviere et al., 2015); given 4-word tuples (belonging to the same POS), participants had to select the most specific and the least specific word within each tuple. The words used to collect specificity ratings with this methodology are the same used to collect concreteness ratings by Montefinese et al. (2014). Bolognesi and Caselli investigated the relation between human-generated concreteness and specificity ratings and reported a low positive and significant correlation of 0.316 (Spearman correlation coefficient; $p < 0.05$), corresponding to an R^2 of 0.1. This result is evidence that Concreteness and Specificity capture different aspects of abstraction, which are only partially correlated with one another.

The entire BC dataset contains 771 nouns, 220 adjectives, and 59 verbs. Our study focused only on nouns, the larger group among the three parts of speech (Figure 1).

3.2 Italian Distributional Semantic Spaces

For our experiment, we built a Distributional Semantic Space (DSM) for Italian words. We extracted the textual information from La Repubblica (Baroni et al., 2004) and itWaC (Baroni et al.,

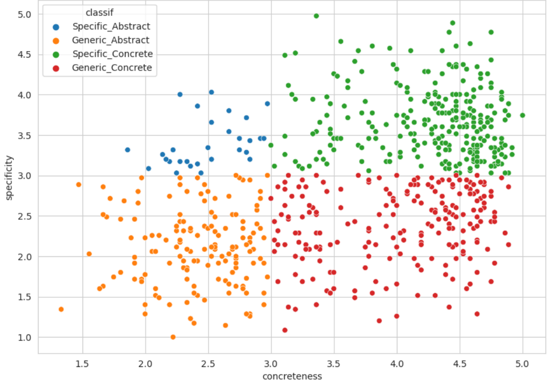


Figure 1: Distribution of the 662 nouns used in the analysis. To approximate the four prototypical types of words different colors are hereby used, although concreteness and specificity have been analyzed as continuous and not as categorical variables.

2009), two pos-tagged and dependency-parsed corpora of Italian. Specifically, we selected a list of nouns, verbs, and adjectives (lemmas used as contexts) with a frequency ≥200 and collected their co-occurrences within a 2- and 10-word symmetric window centered on the target word, which was a noun. We filtered out <target, context> pairs with a frequency of less than 20[2]. The resulting co-occurrence counts were used to i) extract the most associated contexts of a word, using Positive Pointwise Mutual Information (PPMI[3]) score, and ii) built a count-based matrix[4] with PPMI weights and reduced it to 300 dimensions by applying the Singular Value Decomposition (SVD) transformation (Landauer and Dumais, 1997). While we are aware that there are more recent and sophisticated methods, we rely on more stable and explicable representations for the aim of this investigation. We obtained two semantic spaces depending on the context window: **ITAw2** selects nearby words (±2 lemmas surrounding the target word) and contains 19,054 lemmas; **ITAw10** considers a wide contextual window (±10 words) and includes 65,532 lemmas. ITAw10 covers most of the nouns of the BC dataset (754/771), while ITAw2 includes only 662 nouns.

We performed qualitative analyses of the top contexts (CX) and nearest neighbors (NN) for words exemplifying the four prototypical configurations of concreteness and specificity: *abitazione* (‘house’; generic concrete), *ambulanza* (‘ambulance’; specific concrete), *fantasia* (‘fantasy’; generic abstract), and *bancarotta* (‘bankrupt’; specific abstract). Tables 1,2, 3, and 4 report the top neighbors (NNs) ordered by cosine similarity, and the top contexts (CXs) ranked by their PPMI with the target word. Comparing the values reported in the tables reveals differences in the contexts extracted using different window sizes. As expected, verbs and adjectives are the most associated contexts within a ±2-word window. Considering a larger context, top contexts are mostly nouns for concrete words (*abitazione*, ‘house’ and *ambulanza*, ‘ambulance’; Table 1 and 2); some verbs are however highly associated to abstract words (*fantasia*, ‘fantasy’ and *bancarotta*, ‘bankrupt’; Table 3 and 4). While the contexts selected are pretty different, the resulting spaces are coherently similar: the neighbors produced by the two spaces overlap a lot, specifically for *abitazione* (‘house’; Table 1) and *bancarotta* (‘bankrupt’; Table 4). However, similarity scores are considerably lower for ITAw2, indicating that the space is less dense than ITAw10, probably because of the lower number of lemmas and occurrences used to build the DSM.

3.3 Distributional Measures of Contextual Variability

The outcome provided by previous empirical models is that the more abstract a word is, the higher the number of contexts in which it occurs. Conversely, the more concrete a word is, the lower should be the number of its contexts. As introduced above, several computational measures have been proposed to operationalize contextual variability, i.e., how close a word and its contexts are, by relying on DSMs. Given the variety of formulas and terminology, we decided to re-implement previous measures of contextual variability, distinguishing between two subgroups: neighborhood density and contextual richness.

Neighborhood density quantifies how dense the distributional space is near a target word, that is, how close its paradigmatic neighbors are. Looking at a different angle, the higher the average similarity between a word and its neighbors means that many words have a similar contextual distribution. Following Schulte im Walde and Frassinelli (2022), we provide two measures of neighborhood density, Target-Neighbors (TN) similarity and Neighbors-

[2] We tested different values for the filter hyper-parameters and selected the combination that best balances coverage with parser noise.

[3] This is the standard Pointwise Mutual Information, but with negative values raised to 0.

[4] We employed DISSECT toolkit (Dinu et al., 2013).

CX				NN			
w2		w10		w2		w10	
dibire-v ‘adhibit-v’	10.82	*censimenti-n* ‘census-n’	10.07	*appartamento-n* ‘apartment-n’	0.87	*alloggio-n* ‘lodging-n’	0.8
perquisire-v ‘search-v’	10.44	*furti-n* ‘thefts-n’	9.34	*alloggio-n* ‘lodging-n’	0.7	*appartamento-n* ‘apartment-n’	0.79
irruzione-n ‘raid-n’	9.17	*enfiteusi-n* ‘emphyteusis-n’	9.16	*edificio-n* ‘building-n’	0.63	*fabbricato-n* ‘building-n’	0.78
perquisizione-n ‘search-n’	8.80	*pertinenziali-n* ‘appurtenant-n’	8.97	*immobile-n* real ‘estate-n’	0.61	*abitativo-a* ‘housing-a’	0.76
lussuoso-a ‘luxurious-a’	8.54	*sfitto-a* ‘vacant-a’	8.85	*villa-n* ‘villa-n’	0.6	*condominio-n* ‘condominium-n’	0.74
situare-v ‘situate-v’	8.24	*unifamiliare-a* ‘single-family-a’	8.68	*albergo-n* ‘hotel-n’	0.58	*edificio-n* ‘building-n’	0.72

Table 1: Top 6 contexts (CX) and nearest neighbors (NN) of *abitazione* (‘house’; spec:2.2, conc:4.63).

CX				NN			
w2		w10		w2		w10	
sirena-n ‘siren-s’	12.28	*automedica-n* ambulance ‘car-s’	14.95	*pullman-n* ‘bus-s’	0.66	*autoambulanza-n* ambulance ‘car-s’	0.86
autista-n ‘driver-s’	11.20	*barellieri-n* ‘stretcher_bearers-n’	14.93	*trafelato-a* ‘breathless-a’	0.62	*soccorrere-v* ‘rescue-v’	0.81
attrezzare-v ‘equip-v’	10.51	*suem-n* ‘Medical Service acronym’	13.52	*taxi-n* ‘taxi-n’	0.61	*pompiere-n* ‘firefighter-n’	0.8
caricare-v ‘load-v’	9.86	*bonura-n* -	13.22	*autoambulanza-n* ‘ambulance-n’	0.61	*elisoccorso-n* ‘helicopter-n’	0.78
trasportare-v ‘transport-v’	9.73	*voltolini-n* ‘private ambulance service’	12.84	*barella-n* ‘stretcher-n’	0.6	*soccorso-n* ‘rescue-n’	0.78
croce-n ‘cross-n’	8.85	*elisoccorso-n* ‘helicopter_ rescue-n’	12.45	*autobus-n* ‘bus-n’	0.59	*soccorritore-n* ‘rescuer-n’	0.78

Table 2: Top 6 contexts (CX) and nearest neighbors (NN) of *ambulanza* (‘ambulance’; spec: 4.14, conc:4.75).

CX				NN			
w2		w10		w2		w10	
inventivo-a ‘inventive-a’	12.37	*juvenilia-n* ‘juvenilia-n’	11.48	*immaginazione-n* ‘imagination-n’	0.7	*immaginazione-n* ‘imagination-n’	0.81
fervido-a ‘fervid-a’	12.21	*hamill-n* ‘hamill-n’	10.68	*invenzione-n* ‘invention-n’	0.58	*fantastico-a* ‘fantastic-a’	0.78
stuzzicare-v ‘tease-v’	12.00	*sbizzarrire-v* ‘indulge-v’	10.27	*intelligenza-n* ‘intelligence-n’	0.54	*emozione-n* ‘emotion-n’	0.76
guizzo-n ‘leer-n’	11.12	*solleticare-v* ‘tickle-v’	9.57	*immaginario-n* ‘imaginary-n’	0.54	*fascino-n* ‘charm-n’	0.76
scatenato-a ‘unbridled-a’	10.83	*pindarico-a* ‘pindaric-a’	9.21	*estro-n* ‘whimsical-n’	0.52	*passione-n* ‘passion-n’	0.75
sfrenato-a ‘unbridled-a’	10.56	*trezzano-n* ‘trezzano-n’	9.13	*passione-n* ‘passion-n’	0.5	*invenzione-n* ‘invention-n’	0.75

Table 3: Top 6 contexts (CX) and nearest neighbors (NN) of *fantasia* (‘fantasy’; spec:1.62, conc: 1.66).

CX				NN			
w2		w10		w2		w10	
fraudolento-a ‘fraudulent-a’	15.03	*fraudolento-a* ‘fraudulent-a’	13.51	*falso-n* ‘false-n’	0.8	*concussione-n* ‘concussion-n’	0.88
orlo-n ‘hemming-n’	11.53	*pluriaggravato-a* ‘aggravated-a’	13.29	*peculato-n* ‘embezzlement-n’	0.79	*peculato-n* ‘embezzlement-n’	0.86
concorrere-v ‘concur-v’	10.02	*orlo-n* ‘hem-n’	10.78	*appropriazione-n* ‘embezzlement-n’	0.78	*fraudolento-a* ‘fraudulent-a’	0.85
concorso-n ‘conspiracy-n’	9.36	*crac-n* ‘crac-n’	9.71	*concussione-n* ‘concussion-n’	0.76	*aggiotaggio-n* ‘agiotage-n’	0.84
truffa-n ‘fraud-n’	7.90	*bancarotta-n* ‘bankruptcy-n’	9.64	*ricettazione-n* ‘fencing-n’	0.75	*crac-n* ‘cracking-n’	0.82
falso-n ‘forgery-n’	7.25	*delinquere-v* ‘delinquency-v’	9.36	*truffa-n* ‘swindling-n’	0.75	*truffa-n* ‘fraud-n’	0.81

Table 4: Top 6 contexts (CX) and nearest neighbors (NN) of *bancarotta* (‘bankrupt’; spec: 4, conc: 2.27).

Neighbors similarity (NN):

- **TN**: the average vector-space distance between *t* and its *k* nearest neighbors.

- **NN**: the average vector-space distance between the *k* nearest neighbors of *t*.

Vector-space distance is computed as the cosine similarity between two word vectors.

Conversely, **context richness** looks at the syntagmatic contexts in which a word occurs. It considers the strength of a target noun with its most associated contexts and looks at their respective similarity (similar to semantic diversity). In this case, the highest the value, the more the top contexts have similar vectorial representations, so they refer to similar objects and events; on the contrary, lower scores represent a high variability in the contexts. We implemented several measures of context richness. Target-Contexts similarity (TC) and Contexts-Contexts (CC) similarity are derived from Schulte im Walde and Frassinelli (2022), while *Distributional of Context Richness* (DCR) index was proposed by Lenci et al. (2018):

- **TC**: the average vector-space distance between *t* and its *k* top contexts.

- **CC**: the average vector-space distance between the *k* top contexts of *t*.

- **DCR**: the mean of the PPMI scores of the *k* top contexts of the target noun *t*.

Additionally, we computed the contextual entropy, or average information content (Shannon, 1948), which is a classic measure in computational linguistics and is used as an estimate of context informativeness. The assumption is that the higher the entropy, the more uncertain a word is, or a word is less expected given the linguistic contexts. This measure has been previously introduced as a measure of hypernymy prediction (Santus et al., 2014; Shwartz et al., 2017). We calculated the word entropy (H) considering all the probability between a word and the contexts selected to create the vector space:

$$\mathbf{H}(w) = -\sum_{c} p(c|w) * log_2(p(c|w) \qquad (1)$$

where $p(c|w)$ is obtained through the ratio between the frequency of $< w, c >$ and the total frequency of w.

	ITAw10		ITAw2	
	M	St.dev	M	St.dev
TN_5	**0.771**	0.069	**0.648**	0.133
TN_10	0.741	0.069	0.610	0.133
TN_20	0.706	0.069	0.566	0.131
TN_50	0.651	0.067	0.498	0.122
NN_5	0.694	0.110	0.609	0.214
NN_10	0.653	0.104	0.558	0.216
NN_20	0.606	0.099	0.496	0.210
NN_50	0.535	0.091	0.392	0.182
TC_5	0.457	0.174	0.239	0.171
TC_10	0.433	0.159	0.225	0.158
TC_20	0.406	0.145	0.208	0.144
TC_50	0.364	0.133	0.191	0.131
CC_5	0.434	0.206	0.277	0.251
CC_10	0.392	0.171	0.231	0.200
CC_20	0.351	0.134	0.191	0.154
CC_50	0.306	0.100	0.154	0.113
DCR	6.009	2.158	5.979	3.476
H	4.783	1.003	4.641	1.065

Table 5: Descriptive statistics of CV measures.

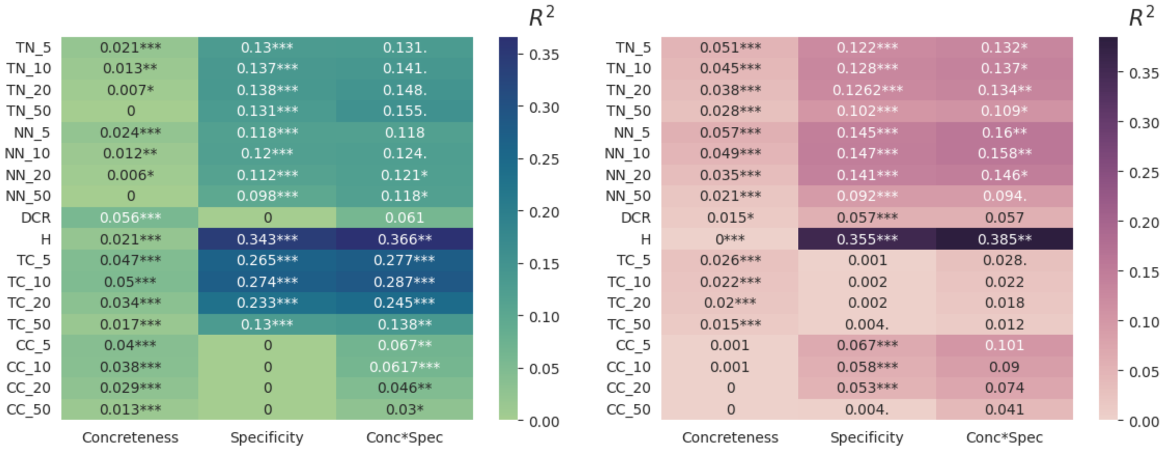
Neighborhood density and context richness are complementary aspects of contextual variability; however, we keep them separated to avoid theoretical and methodological misinterpretations. Formulas are reported in Appendix A.

4 Experimental investigations

Given the 662 nouns attested both in ITAw2 and ITAw10 spaces, we computed all the contextual variability metrics introduced above. We performed the computation with different values of *k* (5, 10, 20, 50) to see how many contexts/neighbors influence the overall score. Table 5 summarizes all computed measures' mean and standard deviation. We observed that the higher the number of contexts/neighbors we select, the lower the overall mean. Moreover, the DCR metric has a high standard deviation, indicating that PPMI scores are not well distributed. The low PPMI scores could be the cause of this issue (see the qualitative analyses above), probably a consequence of the small dimension of the corpora used to extract co-occurrences. This issue is also partially reflected in the entropy measure, with a standard deviation of around 1.

Subsequently, we ran a series of regression analyses[5] aimed at understanding the relations between contextual variability metrics and concreteness/specificity scores. In detail, we ran linear regressions having a context variability metric as the dependent variable; as the independent variable, we consider i) only the Concreteness score, ii) only the Specificity score, and iii) the interaction between

[5] We ran linear models in R (v. 3.6.3) with `stats` package.



	Concreteness	Specificity	Conc*Spec
TN_5	0.021***	0.13***	0.131.
TN_10	0.013**	0.137***	0.141.
TN_20	0.007*	0.138***	0.148.
TN_50	0	0.131***	0.155.
NN_5	0.024***	0.118***	0.118
NN_10	0.012**	0.12***	0.124.
NN_20	0.006*	0.112***	0.121*
NN_50	0	0.098***	0.118*
DCR	0.056***	0	0.061
H	0.021***	0.343***	0.366**
TC_5	0.047***	0.265***	0.277***
TC_10	0.05***	0.274***	0.287***
TC_20	0.034***	0.233***	0.245***
TC_50	0.017***	0.13***	0.138**
CC_5	0.04***	0	0.067**
CC_10	0.038***	0	0.0617***
CC_20	0.029***	0	0.046**
CC_50	0.013***	0	0.03*

(a) ITAw10

	Concreteness	Specificity	Conc*Spec
TN_5	0.051***	0.122***	0.132*
TN_10	0.045***	0.128***	0.137*
TN_20	0.038***	0.1262***	0.134**
TN_50	0.028***	0.102***	0.109*
NN_5	0.057***	0.145***	0.16**
NN_10	0.049***	0.147***	0.158**
NN_20	0.035***	0.141***	0.146*
NN_50	0.021***	0.092***	0.094.
DCR	0.015*	0.057***	0.057
H	0***	0.355***	0.385**
TC_5	0.026***	0.001	0.028.
TC_10	0.022***	0.002	0.022
TC_20	0.02***	0.002	0.018
TC_50	0.015***	0.004.	0.012
CC_5	0.001	0.067***	0.101
CC_10	0.001	0.058***	0.09
CC_20	0	0.053***	0.074
CC_50	0	0.004.	0.041

(b) ITAw2

Figure 2: Summary of the linear models using Concreteness, Specificity, Concreteness*Specificity as independent variables, and various context density measures as the dependent variable. Cells report Adjusted R^2 values and p-values. ‘.’=$p < 0.1$, *=$p < .05$, **=$p < .01$, and ***=$p < .001$.

Concreteness and Specificity. The results of the models are reported in Figure 2. The values in the cells correspond to the coefficient of determination R^2, which represents the proportion of the total variation in the dependent variable y accounted for by the regression model. Values of R^2 closer to 1 (darker colors) imply that the regression model explains a large portion of the variance in context variability.

4.1 Main study

The analysis below focuses on interpreting the distributional measures of contextual variability computed on the larger vector space, that is, ITAw10 (Figure 2a).

Concreteness effects Linear models with Concreteness as the independent variable are generally significant, but Concreteness ratings only explain between 1.3% and 5% of contextual variability scores (left column). This outcome reveals that **contextual variability metrics vary as a function of concreteness, but the effect of concreteness on contextual variability is not very high**.

Specificity effects Conversely, Specificity explains the variability of contextual variability values (middle column): TN and NN neighborhood density (around 11-13%), TC context richness (27%), and entropy (34%). However, it does not explain CC metrics. In detail, Specificity explains most of the TC_10 and entropy variance, achieving the highest R^2 scores. The scatterplot in Figure 3 reveals a positive correlation between the two scores (Spearman's ρ = 0.516, $p < 0.001$). Vice versa, entropy is negatively correlated with Specificity (Spearman's ρ = -0.617, $p < 0.001$): the lower the entropy, the higher the Specificity of a word (Figure 4). The two measures reflect the same situation that we can interpret as follows: **more specific words occur in similar contexts**, so they are strongly related to one another, and the word is more expected. Contrariwise, **more generic words are used in a variety of contexts that are not tightly bonded to the target**, so a word is more uncertain for the given context.

We performed a qualitative analysis to corroborate the observed trend. Let us consider the contexts of hamburger (spec: 4.5, conc: 4.1, TC_10: 0.7), a very specific and concrete word. Its contexts are highly similar, and all indicate other kinds of food, such as *ketchup-n*, *patatina-n* (‘fries’), *polpetta-n* (‘meatball’), *panino-n* (‘sandwich’), *manzo-n* (‘beef’). Besides, abstract words with high specificity scores have similar associated contexts. Given collera (‘rage’; spec: 2.9, conc: 2.8, TC_10:0.71), its contexts are other kinds of emotions, like *lussuria-n* (‘lust’), cupidigia-n *(‘ cupidity’)*, *insaziabile-a* (‘voracious l’), *brama-n* (‘eagerness ’), *avidità-n* (‘greed’).

On the contrary, generic words (i.e., with a low Specificity score) have more heterogeneous contexts, causing a drop in the TC values. For instance, acqua (‘water’) is concrete but also quite generic (spec: 2.7, conc: 4.7, TC_10: 0.04), and this is reflected in the variety of less related contexts, such as *canaletti-n* (‘channels’), *cascatelle-n*

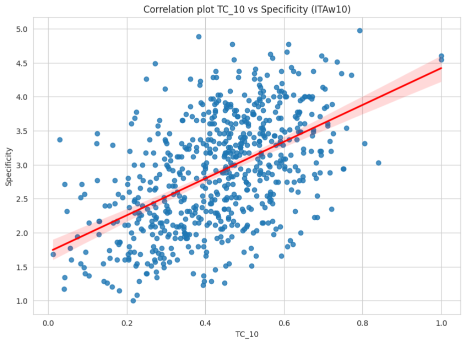


Figure 3: Correlation plots between Specificity and TC_10 measure computed in the ITAw10 space.

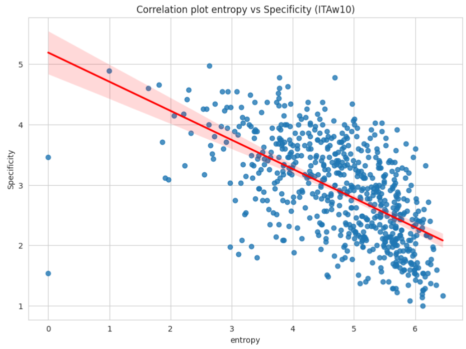


Figure 4: Correlation plots between Specificity and entropy (H) measure computed in the ITAw10 space.

(‘cascade’), *gocciolina-n* (‘drip’), *refrigeratore-n* (‘chiller’), *rigonfiare-v* (‘swell’). Similarly, *tempo* (‘time’; spec: 1.6, conc: 1.6, TC_10: 0.05) has contexts related to the weather, time-traveling, verbal mode, rhythm, and epoch: *viaggiatori-n* (‘traveler’), *zeitgeist-s, trapassato-a* (‘past-tense’), *tiranno-a* (‘tyrant’), *tiranneggiare-v* (‘tyranny’).

It is worth noticing that verbs are more associated with general contexts than specific ones. Qualitative analysis reveals that the difference in the contextual distribution does not overlap with the distinction between abstract and concrete nouns: **Contexts vary depending on the specificity of a word, and this phenomenon is independent of their concreteness**.

Interaction effects Finally, we investigated the interaction between Specificity and Concreteness (right column). Similar to the Specificity models, the interaction explains TC_10 contextual richness (28.7% of the variance) and entropy measures (37% of the variance). However, it has a limited effect on CC measures and is not significant for neighborhood density metrics. Figure 5 illustrates the marginal effects of the interaction of the two terms over TC_10. We can interpret this plot as follows: words with low specificity scores (red line) have lower context richness (TC), but within this group, the more words are concrete, the more they tend to have higher TC scores. However, this effect is reversed for highly specific words (blue line): TC scores tend to decrease for more concrete words.

A similar outcome is observed for the entropy measure (Figure 6). Generic words, both concrete and abstract, have a high entropy (pink line), meaning that these words are little expected given the context words. Conversely, specific words (green line) have a low entropy value, with abstract-specific words having lower entropy than concrete-specific words, meaning that abstract words are more predictable from context than concrete words.

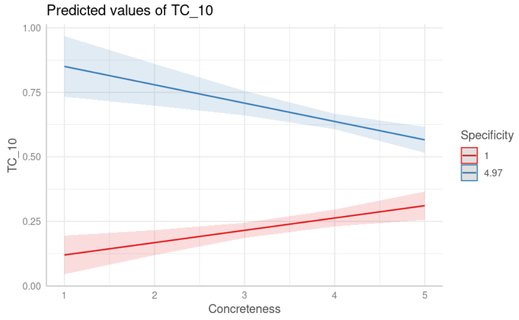


Figure 5: Interaction plot showing the relationship between Concreteness and TC_10 for different levels of Specificity (see also Appendix B).

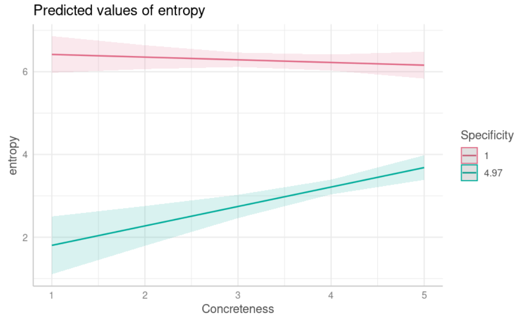


Figure 6: Interaction plot showing the relationship between Concreteness and entropy for different levels of Specificity.

The interaction models reveal a scenario that diverges from previous works: contextual variability does not depend on the dichotomy concrete-abstract, but more on the specificity of the word itself. Surprisingly, abstract-specific words like ‘bankruptcy’ have lower contextual variability than concrete-specific words like ‘hamburger’; that is, **abstract and specific words occur in a more**

limited and predictable number of selected contexts.

4.2 General observations

Comparing the linear models for the two spaces, the heatmaps in Figure 2 show that regression models are similar for neighborhood density (top of the heatmaps). This suggests that the two distributional spaces, while relying on different co-occurrence patterns, tend to build similar word representations. However, coefficients differ for context richness. High R^2 values are obtained considering the average cosine similarity between the target word and its context (TC) for the ITAw10 space in both Specificity and Interaction models, and average context-context (CC) similarity explains part of the variance in the Interaction model. Interestingly, ITAw2 shows an opposite trend: TC scores are not significant (Specificity and Interaction models), and a small variance is explained for CC values by the Specificity model. This outcome seems to confirm that a 2-word window is too small to extract useful distributional information. Overall, the analyses suggest that distributional measures are helpful for investigating cognitive assumptions, but the choice of the model could influence the final outcome.

We also run correlations across contextual variability measures in order to see how they overlap and complement each other. The main outcome is that TC_10 and entropy are strongly negatively correlated (Spearman's ρ = -0.713, $p < 0.001$), but only for ITAw10 space. As observed in the "Specificity effects" section, they represent the same distributional signature of a word but from a different perspective. Moreover, entropy negatively correlates with neighborhood density scores for both spaces. For instance, the correlation between entropy and TN_50 is ρ = -0.513 (ITAw10) and ρ = -0.472 (ITAw10), $p < 0.001$. In contrast, we see low or no correlations between neighborhood density and context richness measures. Correlation matrices are reported in Appendix C.

To conclude, while neighborhood density measures capture some information related to both Concreteness and Specificity, entropy and TC_10 are the best contextual variability metrics associated with Specificity. It is worth noticing that TC_10 was the best measure reported by Schulte im Walde and Frassinelli (2022), but for predicting the concreteness of a word in a pair.

5 Discussion and Conclusion

These analyses provide an enriched view of the relationship between abstraction and contextual variability compared to previous research. In particular, by adding a neglected aspect of abstraction, namely categorical Specificity, we observed that the difference in contextual variability is actually more dependent on Specificity than on Concreteness. These analyses provide an enriched view of the relationship between abstraction and contextual variability compared to previous research. In particular, by adding a neglected aspect of abstraction, namely categorical Specificity, we observed that the difference in contextual variability is actually more dependent on Specificity than on Concreteness. In particular: similar and targeted contexts occur with specific words, while generic words (both abstract and concrete) are associated with more extensive and heterogeneous contexts. To answer our initial research questions, therefore: concreteness explains part of the variation in contextual variability of nouns, but more variation is explained by specificity and by the interaction between the two variables.

Three key points that the current study makes: First, it revises various terminologies related to contextual variability. Second, it is the first study to directly explore contextual variability using the relationship between specificity and concreteness operationalized through human-generated ratings. Finally, it is the first study to conduct this analysis within the context of the Italian language. The outcomes hereby reported corroborate Bolognesi et al. (2020)'s argument: Categorical abstraction (specificity) is a variable that is deeply affected by language rather than by perceptual information, and therefore it has a stronger relationship with how words are used in context (contextual variability). Conversely, concreteness is less shaped by the patterns of linguistic occurrences, and arguably it is more deeply affected by perceptual experience.

Future investigations could focus on fine-grained analyses of different types of nouns, as well as on adjectives and verbs. Co-occurrence patterns differ across part-of-speech, but given the limited number of verbs (less than 60), we preferred to focus on nouns only. The present study opens the way to a new line of research in cognitive and computational linguistics and provides a promising different perspective on the analysis of concepts at different levels of abstraction.

Acknowledgements

Funded by the European Union (GRANT AGREEMENT: ERC-2021-STG-101039777). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. We thank the reviewers for their useful suggestions.

References

Marco Baroni, Silvia Bernardini, Federica Comastri, Lorenzo Piccioni, Alessandra Volpi, Guy Aston, and Marco Mazzoleni. 2004. Introducing the la repubblica corpus: A large, annotated, tei (xml)-compliant corpus of newspaper italian. In *LREC*.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43:209–226.

Marianna Bolognesi, Christian Burgers, and Tommaso Caselli. 2020. On abstraction: decoupling conceptual concreteness and categorical specificity. *Cognitive Processing*, 21(3):365–381.

Marianna Marcella Bolognesi and Tommaso Caselli. 2022. Specificity ratings for italian data. *Behavior Research Methods*, pages 1–18.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46:904–911.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. DISSECT - DIStributional SEmantics composition toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 31–36, Sofia, Bulgaria. Association for Computational Linguistics.

Diego Frassinelli, Daniela Naumann, Jason Utt, and Sabine Schulte Im Walde. 2017. Contextual characteristics of concrete and abstract words. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers*.

Felix Hill, Anna Korhonen, and Christian Bentz. 2014. A quantitative empirical analysis of the abstract/concrete distinction. *Cognitive science*, 38(1):162–177.

Paul Hoffman. 2016. The meaning of 'life' and other abstract words: Insights from neuropsychology. *J. Neuropsychol.*, 10(2):317–343.

Paul Hoffman, Matthew A Lambon Ralph, and Timothy T Rogers. 2013. Semantic diversity: A measure of semantic ambiguity based on variability in the contextual usage of words. *Behavior research methods*, 45:718–730.

Frank Jessen, Reinhard Heun, Michael Erb, D-O Granath, Uwe Klose, Andreas Papassotiropoulos, and Wolfgang Grodd. 2000. The concreteness effect: Evidence for dual coding and context availability. *Brain and language*, 74(1):103–112.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Alessandro Lenci, Gianluca E Lebani, and Lucia C Passaro. 2018. The emotions of abstract words: A distributional semantic analysis. *Topics in cognitive science*, 10(3):550–572.

Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

George A Miller and Christiane Fellbaum. 1991. Semantic networks of english. *Cognition*, 41(1-3):197–229.

Maria Montefinese, Ettore Ambrosini, Beth Fairfield, and Nicola Mammarella. 2014. The adaptation of the affective norms for english words (anew) for italian. *Behavior research methods*, 46:887–903.

Allan Paivio. 1991. Dual coding theory: Retrospect and current status. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 45(3):255.

Penny M Pexman, Ian S Hargreaves, Paul D Siakaluk, Glen E Bodner, and Jamie Pope. 2008. There are many ways to be rich: Effects of three measures of semantic richness on visual word recognition. *Psychonomic Bulletin & Review*, 15:161–167.

Gabriel Recchia and Michael N Jones. 2012. The semantic richness of abstract concepts. *Front. Hum. Neurosci.*, 6:315.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte Im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42.

Paula J Schwanenflugel. 2013. Why are abstract concepts hard to understand? In *The psychology of word meanings*, pages 235–262. Psychology Press.

Paula J Schwanenflugel and Edward J Shoben. 1983. Differential context effects in the comprehension of abstract and concrete verbal materials. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 9(1):82.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 65–75, Valencia, Spain. Association for Computational Linguistics.

Gabriella Vigliocco, Stavroula-Thaleia Kousta, Pasquale Anthony Della Rosa, David P Vinson, Marco Tettamanti, Joseph T Devlin, and Stefano F Cappa. 2014. The neural representation of abstract words: the role of emotion. *Cerebral Cortex*, 24(7):1767–1777.

Sabine Schulte im Walde and Diego Frassinelli. 2022. Distributional measures of semantic abstraction. *Frontiers in artificial intelligence*, 4:206.

A Contextual Variability Measures

Measures of neighborhood density:

- **TN**: the average vector-space distance between *t* and its *k* nearest neighbors.

$$TN(t) = \frac{1}{k}\sum_{i=1}^{k} similarity(t,i) \qquad (2)$$

- **NN**: the average vector-space distance between the *k* nearest neighbors of *t*.

$$NN(t) = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{k} similarity(i,j) \qquad (3)$$

where $i \neq j$

Measures of context richness:

- **TC**: the average vector-space distance between *t* and its *k* top contexts.

$$TC(t) = \frac{1}{k}\sum_{c=1}^{k} PPMI(t,c_i) \qquad (4)$$

- **CC**: the average vector-space distance between the *k* top contexts of *t*.

$$CC(t) = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{k} similarity(i,j) \qquad (5)$$

where $i \neq j$

- **DCR**: the mean of the PPMI scores of the *k* top contexts of the target noun *t*.

$$DCR(t) = \frac{1}{k}\sum_{i=1}^{k} PPMI(t,i) \qquad (6)$$

B Interaction plot

The plot reported in Figure 5 offers a graphical representation of the interaction (or relationship) between two continuous predictors, namely Concreteness and Specificity. In detail, we displayed the fitted values of the dependent variable (TC_10) on the y-axis and the values of the first factor (Concreteness) on the x-axis. The second factor (Specificity) is represented through lines on the chart – each possible value of the second factor gets its own line. As representative values of Specificity, we arbitrarily chose to plot only the two extreme values (1, 4.49 of the Specificity predictor. However, we could have plotted more values of Specificity (see Figure 7).

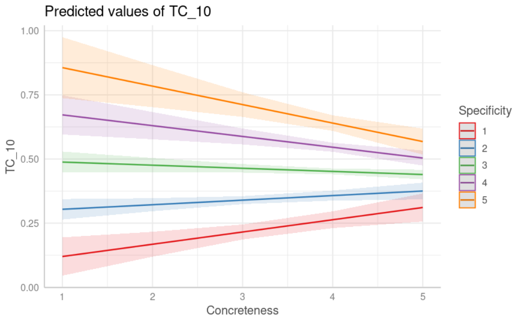


Figure 7: Interaction plot showing the relationship between Concreteness and TC0 for five different levels of Specificity.

C Correlations Between Measures

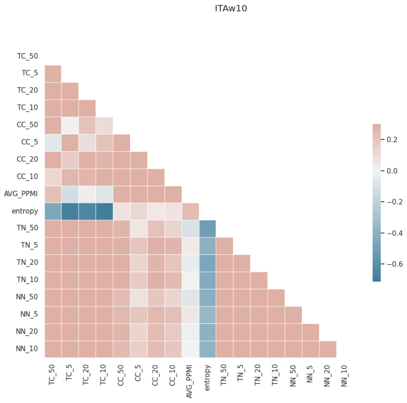


Figure 8: Spearman's ρ correlations among contextual variability measures for ITAw10.

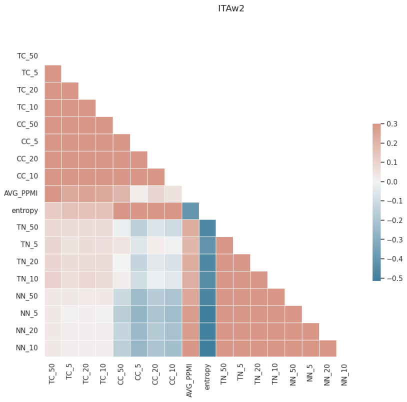


Figure 9: Spearman's ρ correlations among contextual variability measures for Itaw2.

Probing BERT's ability to encode sentence modality and modal verb sense across varieties of English

Jonas Wagner and **Sina Zarrieß**
Bielefeld University
Faculty for Linguistics and Literary Studies
{jonas.wagner,sina.zarriess}@uni-bielefeld.de

Abstract

In this research, we investigate whether BERT can differentiate between modal verb senses and sentence modalities and whether it performs equally well on different varieties of English. We fit probing classifiers under two conditions: contextualised embeddings of modal verbs and sentence embeddings. We also investigate BERT's ability to predict masked modal verbs. Additionally, we classify separately for each modal verb to investigate whether BERT encodes different representations of senses for each individual verb. Lastly, we employ classifiers on data from different varieties of English to determine whether non-American English data is an additional hurdle. Results indicate that BERT has different representations for distinct senses for each modal verb, but does not represent modal sense independently from modal verbs. We also show that performance in different varieties of English is not equal, pointing to a necessary shift in the way we train large language models towards more linguistic diversity. We make our annotated dataset of modal sense in different varieties of English available at https://github.com/wagner-jonas/VEM.

1 Introduction

Work on contextualised embeddings learned by large bidirectional language models such as BERT (Devlin et al., 2019) indicates that they may capture *senses* of lexical items (Loureiro et al., 2021). This has the potential to greatly accelerate variationist research, for example by finding community-specific senses of words (Lucy and Bamman, 2021) or tracing contact-induced semantic shifts (Miletic et al., 2021). Modal sense[1] variation has been an area of interest for variationist researchers (see, e.g. Collins et al., 2014, Hansen, 2018, or Loureiro-Porto, 2019), but has, so far, received comparably little attention in NLP. In this paper, we investigate to what extent modal sense is encoded in BERT embeddings across varieties of English, and if so, at which layer(s) and in what form.

Modality is generally analysed on sentence level (Portner, 2009, 2–6) and is primarily expressed in English by the use of modal verbs (Portner, 2009, 4), with each verb potentially evoking different senses. Consider *must* in the following two sentences: "You *must* complete all tasks for course credit" and "You *must* be tired after the long journey". In the first sentence, *must* has deontic sense, i.e. it is used to express orders or recommendations, which can also be expressed by e.g. *should*. In the second sentence, *must* has epistemic sense, i.e. a qualification of certainty. This can be expressed by many modal verbs, such as *may*, *can*, *could*, or *might*. In addition to these two, there are also concessive (granting or denying permission, e.g. *may* and *can*) and dynamic (expressing ability, e.g. *can*) sense.[2] The modal verb therefore affects the interpretation of the sentence as a whole: swapping *must* and *may* in "You *must*/*may* leave now" clearly affects more than only the meaning of the modal verbs themselves.

How often each modal verb expresses which sense is prone to variation and at times starkly differs between varieties of English. This has been researched in-depth. For example, Collins et al. (2014) investigate domain-specific variation of modal verb sense distribution in Philippine English and compare it to American and British English. Hansen (2018) provides what is probably the most comprehensive treatment of modal verb sense in varieties of English, finding that e.g. British and Indian English have high incidences of epistemic

[1] Linguists often differentiate between modality, which is analysed on sentence level, and modal verb sense for individual modal verbs. As we investigate both, we use the term "modal sense" where we refer to both.

[2] Other senses exist, but will not be discussed in this work; see also Ruppenhofer and Rehbein (2012).

Proceedings of the 15th International Conference on Computational Semantics pages 28–38
June 21–23, 2023. ©2023 Association for Computational Linguistics

must, while Hong Kong and Singapore English have higher incidences of deontic *must*.

Most of these studies remain small in scale. They investigate only a small number of varieties, a small number of modal verbs, or small corpora. This is unsurprising, as modal verb sense annotation is largely done manually. Large-scale computational investigations in this area would be a valuable contribution, but these different distributions of modal senses may pose a challenge for pre-trained language models, which are often not trained on diverse data and may struggle with other varieties' different modal verbs being usage.

These simple facts about modal sense raise questions regarding BERT's potential to capture modal sense which have not been addressed in recent work on probing BERT's abilities to encode lexical semantics, cf. among others, Aina et al. (2019); Pilehvar and Camacho-Collados (2019); Vulić et al. (2020); Garí Soler and Apidianaki (2021). Ideally, BERT would capture modal sense both at sentence (in the [CLS] token) and word level (in modal verbs' embeddings). The latter needs more differentiation: representation could be independent from the individual verbs (e.g. epistemic *must* and epistemic *may* share encoded epistemic sense) or different modal senses are only represented for each individual verb (epistemic and deontic *must* encode different senses, but these would not be shared by epistemic and deontic *should*). Further, it should show systematic encodings of lexical and sentential modal sense across different layers, in light of other work showing linguistic systematicity in processing different aspects of linguistic knowledge across layers (Aina et al., 2019; Pilehvar and Camacho-Collados, 2019; Vulić et al., 2020; Garí Soler and Apidianaki, 2021). And, last but not least, it should encode modal sense in a way that is robust to distributional differences of modal senses and verbs across varieties of English.

Beyond accelerating variationist research, correct classification of modal sense also has relevance for NLP tasks. Modal sense classification has been used in connection with sentiment analysis (Liu et al., 2014), hedging and detection of hypotheses and speculation (Morante and Daelemans, 2009; Vincze et al., 2008; Malhotra et al., 2013),[3] and factuality detection (Saurí and Pustejovsky, 2012), among others. These are key tasks that, ideally, should function in different varieties of English – not just majority varieties.

We conduct a series of experiments to investigate if, and how, BERT encodes modal sense. We train probing classifiers on annotated datasets (see Section 3 for our data) and classify modal sense. We do this for modal verbs' embeddings as well as sentence embeddings (experiment 1, Section 4). We also train separate classifiers for each modal verb (experiment 2, Section 5); we extend this methodology to data from several different varieties of English (experiment 4, Section 7). We also test whether BERT can predict masked modal verbs, even if it cannot classify modal sense (experiment 3, Section 6).

2 Background

2.1 Semantic knowledge encoded in BERT

While BERT (Devlin et al., 2019) has been used to investigate many facets of the semantic meanings of words (e.g. Wiedemann et al., 2019; Vulić et al., 2020; Zhang et al., 2020; Bhardwaj et al., 2021; Garí Soler and Apidianaki, 2021; Lucy and Bamman, 2021; Miletic et al., 2021; Apidianaki, 2023 among others), some aspects of meaning cannot be captured by BERT embeddings. Ettinger (2020) found that BERT does not appear to process negation at all: both *a robin is a* and *a robin is not a* are predicted to most likely end with *bird* or *robin*. Therefore, more research into the kinds of meaning contained in BERT embeddings is necessary.

Simultaneously, previous research on classifying modal senses with static embeddings indicates that contextualised word embeddings may be useful to improve modal sense classification. Li et al. (2019) use static embeddings for modal sense classification, but adjust each embedding's weight based on distance from the modal verb and POS-tag, which improves results. Marasović et al. (2016) present one of the most comprehensive studies on modal sense classification to date, and point to the importance of lexical features of embedded verbs and the subject in the sentence as giving cues to the modal verbs' meanings. Their experiments also analyze the effect of variation in the distribution of modal senses in different datasets and genres. In more recent work, Pyatkin et al. (2021) go beyond Marasović et al. (2016)'s setup that is restricted to modal verbs and propose a more complex modality detection task involving a broader set of modality triggers and the detection of events associated with

[3] While Vincze et al. (2008) do not explicitly mention modal sense, they do point to the importance of modal auxiliaries in uncertainty detection.

them. As our work aims for a controlled analysis of the representation of modal verb sense across varieties of English, we follow Marasović et al. (2016) and leave the exploration of further modality triggers to future work.

2.2 Variationist NLP research

There has been some NLP research into variation within English. For example, Lucy and Bamman (2021) successfully use contextualised BERT embeddings to find community-specific meanings of words like *python*, which may refer to a programming language or a fictional spaceship. Similarly, Miletic et al. (2021) use contextualised BERT embeddings to find contact-induced semantic shift in English in Quebec. These studies demonstrate that BERT can be used to study variation within English, even between different varieties. But we see two issues with them. Firstly, much World Englishes research focuses types of sense variation other than homonymy, such as the different distributions of modal senses. Secondly, by using BERT to investigate variation, the authors inherently assume that BERT can capture such variation. While their results support this assumption, this does not mean that BERT is an adequate tool to measure all kinds of differences between varieties of English.

The exact nature of BERT's training data is opaque, but Devlin et al. (2019) mention that they use two sources of data for pre-training. These are the 800 million word BooksCorpus (Zhu et al., 2015), consisting of 11,038 unpublished books, and a large 2.5 billion token corpus of Wikipedia entries. While the exact makeup of who wrote those texts is unknown, some reasonable guesses can be made regarding the larger Wikipedia sample. Wikipedia publishes data on the demographic makeup of its contributors,[4] which indicates that a plurality of edits are made from the United States, followed by the United Kingdom and Canada. This is not a perfect method – just because a user is accessing Wikipedia from the United States does not mean that they also speak American English – but it still provides a basis for the assumption that most of BERT's training data comes from the so-called "Inner Circle" (Kachru, 1985), i.e. those countries where English is spoken as a first language by most of the population. This suggests that BERT's training data lacks diversity with regards to varieties of English, which may adversely affect its ability to process English produced by speakers of those under- or unrepresented varieties.

3 Data

We use two existing datasets to test whether BERT can differentiate modal verb senses and construct a new one, taking data from the International Corpus of English (ICE). The first is a portion of the Multi-Perspective Question Answering (MPQA) dataset that has been annotated for modal sense (Ruppenhofer and Rehbein, 2012). This consists of 1,201 sentences taken from news articles dated June 2001 to May 2002 (Wiebe et al., 2005; Ruppenhofer and Rehbein, 2012). The second is the heuristically tagged EPOS-E dataset (Marasović et al., 2019), based on the EUROPARL and OpenSubtitles-English datasets, consisting of data from the European Parliament and film subtitles, comprising 2,453 sentences. Modal sense is annotated for each sentence in both datasets. For comparability, we do not report results for *ought*, as it is not evaluated in previous publications either (Marasović et al., 2016). We also remove *might* and *shall* from our results due to their low frequency.

The main difference between the two datasets (besides the annotation methodology) is size, with EPOS-E being almost twice the size of MPQA. They also draw their data from different sources, which is important given the genre effects found by Marasović et al. (2016). MPQA includes more senses than EPOS-E, which we discard in our analysis to maintain comparability. The balancing for the different senses for each modal verb also varies between them: the most common sense for *must* makes up 92% of instances in MPQA, but only 60% in EPOS-E; for *may*, this is 74% and 87%; for *can* 67% and 84%; for *could* 65% and 43%; and for *should* 92% and 94%. This, naturally, may impact classification results. As in previous research, we only investigate the modal verbs that are annotated in the dataset in cases where there is more than one modal verb per sentence.

For the last experiment, in which we test a classifier trained on EPOS-E on data from different varieties of English, we use the written components of eight sub-corpora from the International Corpus of English (ICE; `https://www.ice-corpora.uzh.ch/en.html`), a comparative corpus of varieties of English. For each variety, the same kinds of documents (like

[4] `https://en.wikipedia.org/wiki/Wikipedia:Who_writes_Wikipedia` and `https://stats.wikimedia.org/wikimedia/squids/SquidReportPageEditsPerLanguageBreakdown.htm`

student writings or fiction) are used to compile subcorpora of about 400,000 written tokens for each variety (see http://ice-corpora.net/ice/index.html for more information). We investigate Philippine (PH), Canadian (CA), Irish (IR), Hong Kong (HK), Sri Lankan (SL), Jamaican (JA), Nigerian (NI), and Indian (IN) English. For each modal verb in each variety, we randomly extract 20 sample sentences that contain the modal verb for a total of 800 sample sentences. Three annotators independently annotate these. We discard all instances where no two annotators agree on one sense, where the sense is unclear (e.g. due to missing context), and false positives (e.g. *must* as a noun instead of a modal verb). This leaves 782 sentences for analysis. Agreement between the first and second annotator is highest (83.75%), followed by agreement between the second and third annotator (79.88%), and between the first and third annotator (78.00%). We use the majority labels as gold labels. We call this corpus VEM – the **V**arieties of **E**nglish **m**odal sense corpus – and make it available at https://github.com/wagner-jonas/VEM.

4 Experiment 1

4.1 Methods

In the first experiment, we investigate whether modal verb sense classification is successful using the modal verbs' contextualised embeddings and sentence embeddings in the form of [CLS] tokens. We use a logistic regression classifier (from scikit-learn, version 1.0.2; Pedregosa et al. (2011)) with elasticnet penalty and the L1-ratio set to 0.5. We only train one classifier each for the modal verbs' embeddings and [CLS] token, but report the results split by modal sense and modal verb.

We replicate the setup from Marasović et al. (2016): first, we randomly split MPQA into training (80%) and test (20%) sets. We then train a logistic regression classifier on the training set and predict modal senses in the test set. Then, we add the data from EPOS-E to the MPQA training set – we borrow the name $CL\text{-}^{b}{}_{ME}$ for this from Marasović et al. (2016) – and predict modal senses in the same test set.

We report accuracy for each layer and sense. There, the baseline is the sum of frequencies of modal verbs for which that sense is the most frequent one. That is, if *must* and *shall* are both most frequently deontic, we add up their frequencies to determine the baseline. Split by modal verb, we do not report for each layer, as we only use the 12th layer for classification using modal verb embeddings and the 7th layer for classification using the [CLS] token, since they showed the strongest overall performance (see also Figures 1 and 2). Here, the baseline is the frequency of the verb's most frequent sense.

4.2 Results

Classifying modal verb embeddings, we reach overall accuracies between 0.70 (*can* in MPQA) and 1.0 (*must* in MPQA). We beat our baseline (the most common sense for each modal verb) for *could* and *must* in both datasets, and additionally for *can* in $CL\text{-}^{b}{}_{ME}$ and *may* in MPQA. We only dip below our baseline for *should*. Marasović et al. (2016) beat their respective baseline for *should* and *must* only. Taking the mean accuracy for all senses any individual verb can express, accuracies vary between 0.25 (*may* in $CL\text{-}^{b}{}_{ME}$) and 1.0 (*must* in MPQA). In this case, we beat our baseline for *could* and *must* in MPQA.

Classifying with the [CLS] token instead, we reach overall accuracies between 0.02 (*should* in $CL\text{-}^{b}{}_{ME}$) and 0.73 (*may* in $CL\text{-}^{b}{}_{ME}$ and *could* in MPQA). Here, we only beat our baseline once - for *could* in MPQA. In general, precision and recall are lower than accuracy and results are stronger for MPQA than for $CL\text{-}^{b}{}_{ME}$ - drastically so when classifying using the [CLS] token. For all results, see Table 1.

Separating the results by sense, MPQA performs better than $CL\text{-}^{b}{}_{ME}$ (see Figure 1). Deontic sense is the only sense which (semi-)consistently performs above baseline; other senses hardly, if ever, exceed their baseline. Classifying the [CLS] token (Figure 2), no sense consistently performs above baseline. Accuracy in $CL\text{-}^{b}{}_{ME}$ fluctuates between layers, with one sense usually reaching perfect accuracy and others at zero accuracy.

4.3 Interpretation

Results of the first experiment suggest that there is no single layer of the BERT model that captures modal sense (see Figures 1 – 2). Deontic sense appears easiest to classify (as it is the only sense with accuracies above baseline). Overall, modal sense classification is not successful. It also appears that no modality information is encoded at sentence level, at least in the [CLS] token, given the wild fluctuations between layers (see Figure

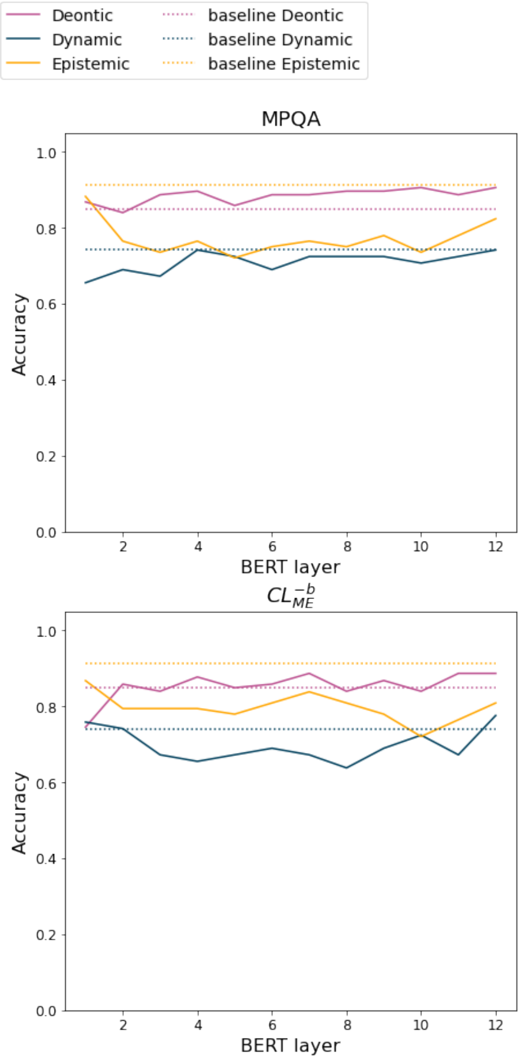


Figure 1: Accuracy of classification of modal verb embeddings per layer, split by dataset and sense.

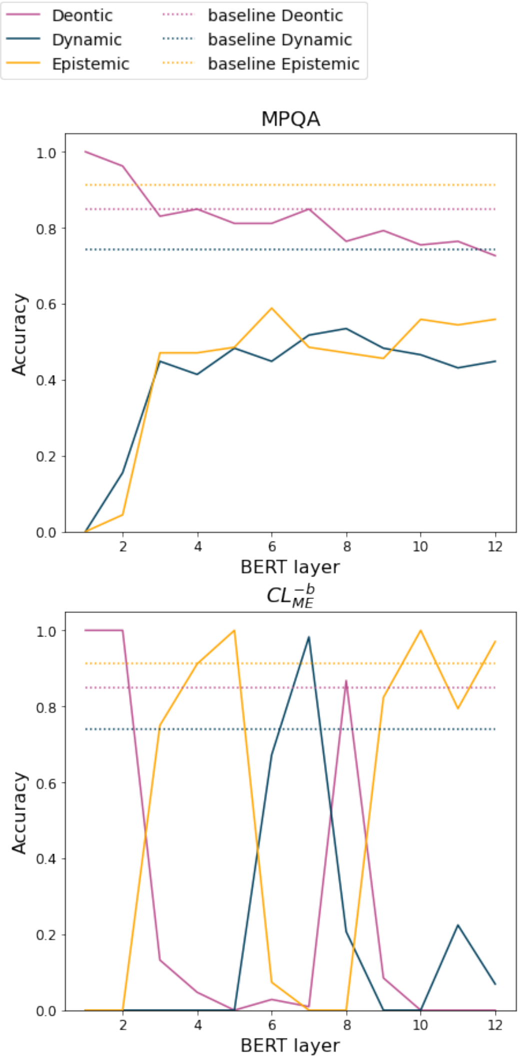


Figure 2: Accuracy of classification of [CLS] token embeddings per layer, split by dataset and sense.

2) – though this might be caused by our choice of classifier.

Viewing individual modal verbs paints a more interesting picture. Some modal verbs appear to be easier to classify, like *could* and *must*. This cannot (just) be due to lower baselines (i.e. a more balanced nature), as *could* has a baseline of 0.67 while *must*'s baseline lies at 0.91. Classification is a lot less successful using the [CLS] token rather than modal verbs' embeddings, which serves to re-affirm the notion that no modality information is encoded on sentence level.

But human speakers (and annotators) do not process modal sense in isolation - they take whichever modal verb is present into account. Thus, it may be that modal sense is not encoded as its own category, but that differences between senses for each individual modal verb (e.g. epistemic and deontic *must*) are. In the next experiment, we therefore train classifiers for each individual modal verb.

5 Experiment 2

5.1 Methods

In this experiment, we train logistic regression classifiers for each modal verb separately, using embeddings from the 12th BERT layer. We use the same parameters as in the first experiment. Note that this does not use the same train and test data as before; as we train separate classifiers for each modal verb, we split data for each modal verb into train and test sets separately. This means that we randomly split data from EPOS-E into 80% training and 20% test data and do the same for MPQA. Note also

Modal verb	*could*		*should*		*can*		*must*		*may*	
Instances	45		57		61		33		33	
Baseline	0.67		0.96		0.70		0.91		0.79	
Training data	MPQA	CL^{-b}_{ME}	MPQA	CL^{-b}_{ME}	MPQA	CL^{-b}_{ME}	MPQA	CL^{-b}_{ME}	MPQA	CL^{-b}_{ME}
Modal verb embedding										
Mean precision per sense	0.28	0.22	0.44	0.1	0.23	0.23	1.0	0.42	0.22	0.25
Mean recall per sense	0.42	0.33	0.44	0.11	0.33	0.33	1.0	0.5	0.25	0.25
Mean accuracy per sense	**0.68**	0.67	0.65	0.31	0.46	0.46	**1.0**	0.5	0.44	0.25
Overall accuracy	**0.71**	**0.69**	0.93	0.89	0.70	**0.72**	**1.0**	**0.94**	**0.88**	0.79
[CLS] embedding										
Mean precision per sense	0.09	0.55	0.09	0.33	0.09	0.34	0.16	0.33	0.1	0.12
Mean recall per sense	0.33	0.75	0.11	0.33	0.22	0.5	0.28	0.33	0.12	0.12
Mean accuracy per sense	0.28	0.6	0.28	0.33	0.27	0.34	0.41	0.33	0.2	0.23
Overall accuracy	0.33	**0.73**	0.82	0.02	0.46	0.07	0.70	0.06	0.64	0.73
Marasović et al (2016)										
Overall accuracy	**0.72**	**0.68**	**0.93**	**0.92**	0.66	0.63	0.94	0.87	0.93	0.90
Baseline	0.65		0.91		0.70		0.94		0.94	

Table 1: Results of modal classification of modal verb/[CLS] token embeddings per modal verb. Mean precision, recall, and accuracy per sense. **Bolded** accuracies are above respective baseline(s) (most frequent sense for each verb). Results from (Marasović et al., 2016) use their semantic features (F_{Sem}), which generally performed best.

that this is a novel methodology and not directly comparable to previous research. And since we train and test on data from MPQA and EPOS-E, respectively, results may skew somewhat positive as we avoid some of the genre effects that Marasović et al. (2016) observe.

5.2 Results

For MPQA, classification of sense for each modal verb shows accuracy between 0.64 (*could*) and 0.96 (*should*). We reach the lowest precision and recall for *may* (precision = 0.29; recall = 0.35); the highest for *must* (precision = 0.83; recall = 0.94). See Table 2 for more results.

For EPOS-E, nearly all metrics are higher than for MPQA. We reach the highest accuracy for *may* at 0.98, the lowest for *could* at 0.84. The lowest precision and recall are reached for *can* (precision = 0.33; recall = 0.31). We reach the highest precision and recall for *may* (precision = 0.95; recall = 0.97).

Accuracy beats the baseline (the frequency of each verb's most common sense) for *could* and *must* in both datasets, additionally for *should* and *can* in MPQA and *may* in EPOS-E. Mean accuracies for each modal verb's potential senses exceed the baseline for *could*, *must*, and *may* in EPOS-E.

5.3 Interpretation

The much improved classification results obtained in this experiment as opposed to the first, where we used a classifier trained on all modal verbs rather than a different one for each modal verb, point to BERT encoding modal verb sense separately for each modal verb. Classification accuracy in both datasets meets or beats the baseline of its most common sense for all verbs. For modal verbs that are dominated by one sense (like *should*), we only rarely exceed the baseline, which is expected, but we do not dip below it, either. The mean accuracy across a modal's possible senses beat the baseline for *could*, *must*, and *may*. These all share a comparatively low baseline, meaning their senses are more balanced than for other modal verbs (though note that *can*, *could*, and *may* in MPQA share this lower baseline but classification is less successful, indicating that it is not the only factor).

In EPOS-E, *must* and *may* see particular success, both reaching precision, recall, and accuracy exceeding 0.93 with baselines of 0.63 and 0.86, respectively. Clearly, BERT does not simply assign one sense to each of these modal verbs. These results suggest that representations for e.g. deontic and epistemic *must* are different, but that there is no overall representation for any one sense.

Lastly, BERT was trained to predict masked tokens. The final test to ascertain BERT's ability to recognise modal sense is therefore masked prediction: can BERT predict masked modal verbs?

6 Experiment 3

6.1 Methods

We mask modal verbs from MPQA and EPOS-E and let BERT predict them, using the *pipeline*

Modal verb	*could*		*should*		*can*		*must*		*may*	
Data set	MPQA	EPOS-E	MPQA	EPOS-E	MPQA	EPOS-E	MPQA	EPOS-E	MPQA	EPOS-E
Instances	45	19	53	30	75	34	38	218	29	213
Mean precision per sense	0.17	0.52	0.62	0.5	0.21	0.33	0.75	0.47	0.15	0.47
Mean recall per sense	0.28	0.61	0.75	0.5	0.33	0.33	0.83	0.5	0.2	0.5
Mean accuracy per sense	0.44	**0.75**	0.75	0.5	0.42	0.33	0.83	**0.94**	0.29	**0.95**
Overall accuracy	**0.64**	**0.84**	**0.96**	0.97	**0.69**	0.91	**0.95**	**0.94**	0.72	**0.98**
Baseline	0.62	0.58	0.92	0.97	0.68	0.91	0.87	0.63	0.72	0.86

Table 2: Modal sense classification results, separate training of classifiers for each modal verb. Overall and mean results by senses. Accuracies that meet or exceed baseline in **boldface**.

function from huggingface's *transformers* library (version 4.23.1; Wolf et al. 2020).

6.2 Results

Success of masked modal verb prediction depends on the modal verb. In both datasets (see Table 3), *should* is predicted correctly most commonly, with an accuracy of 0.44 in MPQA for the top prediction and 0.80 for the top three predictions. In EPOS-E, this rises to 0.52 and 0.83, respectively. *Could* and *must* also are frequently predicted correctly in both datasets, though they switch places: *must* is predicted correctly more often than *could* in EPOS-E, but the reverse is true in MPQA. *May* is predicted correctly least often in all layers. Words other than modal verbs are only predicted rarely: accuracies lie between 0.87 (EPOS-E, first prediction only) and 0.98 (MPQA, top 3 predictions) of predictions are modal verbs.

6.3 Interpretation

This experiment indicates that, as expected from results of per-modal-verb classification in the second experiment (see Section 5), BERT succeeds at predicting modal verbs where it failed at classifying modal verb sense. *May* appears most difficult to predict. *Should*, despite not being the most common modal verb, especially in EPOS-E, where *must* occurs over seven times as often, is correctly predicted most frequently. These results are strong considering the relatively minute semantic differences between modal verbs – syntactically, any of them would be an acceptable prediction.

This partially confirms the observations made in the first and second experiment (Sections 4 and 5). There, too, classification of *must* is overall most successful. Combining this with the results from the second experiment (Section 5), it appears that the relatively strong prediction performance may be unrelated to an overarching representation of modal sense.

Lastly, the question remains whether BERT embeddings encode modal verb sense equally well in different varieties of English.

7 Experiment 4

7.1 Methods

For each modal verb in the varieties of English modal sense corpus (VEM, see Section 3), we train a logistic regression classifier on that verb's instances in the EPOS-E dataset, mirroring the methodology from the second experiment (Section 5). We then predict modal verb senses for that verb in the Varieties of English modal sense corpus (VEM) and compare overall accuracy, precision, and recall for each modal verb and for each variety.

We do not train a separate classifier on each variety of English. While this will undoubtedly lead to diminished success for some (or all) varieties, we believe that this reflects real-world scenarios. By its nature, the amount of data for minority varieties of English will be lower than for the varieties present in EPOS-E. As the point of this experiment is to see whether automatic modal sense classification for other varieties of English is viable, we therefore use the large pre-existing EPOS-E dataset.

7.2 Results

Classification of modal verbs' senses (see Table 4) is most successful for *must* (overall accuracy = 0.90; mean accuracy = 0.86). We reach the lowest overal accuracy for *could* (0.70) and the lowest mean accuracy for each verb's possible senses (0.32), mean precision (0.33), and mean recall (0.30) for *can*.

Modal verb	*could*		*should*		*can*		*must*		*may*		modal rate	
Data set	MPQA	EPOS-E	MPQA	EPOS-E	MPQA	EPOS-E	MPQA	EPOS-E	MPQA	EPOS-E	MPQA	EPOS-E
Instances	216	88	254	139	355	154	173	1054	136	1009		
acc@1	0.34	0.43	0.44	0.52	0.43	0.60	0.38	0.48	0.35	0.44	0.89	0.87
acc@2	0.59	0.61	0.67	0.72	0.59	0.71	0.58	0.68	0.54	0.61	0.95	0.93
acc@3	0.78	0.75	0.80	0.83	0.69	0.74	0.72	0.78	0.65	0.71	0.98	0.95

Table 3: Results of masked modal verb prediction. The last column shows the rate of modal verbs in the top predictions.

Modal verb	*could*	*should*	*can*	*must*	*may*
Instances	156	156	154	158	158
Mean precision per sense	0.23	0.50	0.11	0.43	0.25
Mean recall per sense	0.33	0.50	0.22	0.50	0.25
Mean accuracy per sense	**0.70**	0.50	0.32	**0.86**	0.50
Overall accuracy	**0.70**	0.90	0.73	**0.90**	0.88
Baseline	0.52	0.90	0.79	0.75	0.89

Table 4: Results of modal verb sense classification on varieties of English. **Bolded** accuracies are above respective baseline(s) (most frequent sense for each verb).

Variety	PH	HK	NI	IN	SL	JA	IR	CA
Instances	99	98	99	96	97	98	96	99
Mean precision per modal verb	0.59	0.56	0.57	0.66	0.60	0.59	0.58	0.65
Mean recall per modal verb	0.55	0.54	0.60	0.69	0.60	0.58	0.53	0.61
Mean accuracy per modal verb	0.78	0.76	0.79	0.85	0.85	0.87	0.77	0.91
Overall accuracy	0.78	0.77	0.79	0.85	0.85	0.87	0.77	0.91

Table 5: Results of modal verb sense classification on varieties of English: mean metrics for each variety. Note: we do not report a baseline since, without separating by modal verbs, this would be meaningless.

We reach the highest overall accuracy for Canadian English (0.91), followed by Jamaican (0.87), Sri Lankan, and Indian English (both 0.85). We reach the lowest overall accuracy for Hong Kong English and Irish English (both 0.77) For more results, see Table 5.

We choose the Nigerian English results for a brief example. In Sentence (1), *may* is predicted to have deontic sense, when annotators agreed it should be epistemic. Note the lack of space between *i'm* and *wrong* as well as the (subjectively) non-standard use of *wonder*:

(1) I wondered at one point that you may have forgotten us, but your mail now makes me think i'mwrong

Conversely, in Example (2), epistemic *may* was classified correctly:

(2) Chieftains from the 55 local councils may be lending moral and financial support to their counterparts in the two Ibeju-Lekki councils, sources said

In all correct classifications of *may* in the Nigerian English sample, *be* occurs in the vicinity of *may* – at times negated. The reason for incorrect classification can not be as simple as non-occurrence of *be*, as *be* also occurs in 5 of 15 instances of misclassified *may*, such as in Example (3):

(3) He may be very poor, poorer than a church rat

The instance of *may* in Example (3) was also classified incorrectly as deontic. Note that this sentence appears much less non-standard than the previous example. It must be kept in mind that classification in the second experiment (see Section 5) was also not perfect, meaning that (at least some) wrong classification despite no discernible presence of non-standard language may be caused by general model errors rather than meaning variation. Genre variation and register may also play a role: Example (1) is taken from a social letter, Example (3) from a novel; Example (2), in which modal sense was classified correctly, is taken from press coverage, which may be more similar to the parliament proceedings used in EPOS-E.

7.3 Interpretation

Some of the classification performance differences between modal verbs are mirrored in the second experiment (see Section 5), though nearly all performance metrics are lower compared to the second experiment. This may be due to the different register of the texts: while EPOS-E is comprised of European Parliament proceedings and subtitles, the ICE corpora consist of various kinds of writings, none of which include parliamentary writings or subtitles. This does not account for differences between the varieties, however.

Sense classification being most successful in Canadian English is not surprising, as BERT's training materials are likely predominantly comprised of American English, to which Canadian English bears the greatest similarity (Schneider, 2006; Kytö, 2019). The strong performance reached for Sri Lankan English may be due to the later collection date in the 2010s as opposed to the majority of the ICE corpora, which were collected in the 1990s. Thus, "colonial lag" (Hundt, 2009) may be causing this data to be more similar to the EPOS-E data, though the concept is disputed. The strong performances on Jamaican and Indian English (as Outer Circle varieties; Kachru, 1985) and the poor performance on Irish English (as an Inner Circle variety), are more surprising and warrant further investigation. While the difference between varieties is not enormous (overall accuracies range from 0.77 to 0.91), they are not negligible, either.

8 Conclusion and outlook

Our experiments have demonstrated that BERT does not appear to have any representations of modal sense as its own category. Classification did not show satisfactory results for either modal verb sense the embeddings of modal verbs or modality in the [CLS] token. However, BERT showed some ability to predict masked modal verbs, though its success depends greatly on which modal verb has been masked, making it unclear whether this is truly an ability to predict specific modal verbs or rather prediction of *any* modal verb. Modality does not appear to be encoded in the [CLS] token at all, calling into question whether sentence-level encodings of modality exist in BERT. However, different classifiers may yield different results, and representations of sentence meaning other than the [CLS] token (such as summing up embeddings) may yet encode modality. Further research is thus necessary to come to a complete conclusion.

Classification was most successful when done separately for each individual modal verb. This indicates that, while BERT may not have representations of modal verb sense as its own category, it does appear to encode sense differences for each modal verb. Thus, it can differentiate between *must* in sentences like "You *must* complete all tasks for course credit" and "You *must* be tired after the long journey", but it also views the deontic modal verbs *must* and *should* in a sentence like "You *must*/*should* do your homework" as different. This has some intuitive appeal - clearly, the actual meanings of the sentence change quite considerably with the strength of deontic obligation expressed by *must* and *should*, respectively.

The results of the last experiment demonstrate that the difference in modal verb sense use across different varieties of English may negatively impact this performance. Some varieties (Canadian, Sri Lankan, Indian, and Jamaican English) reach comparatively good performance, while modal verb sense classification in Irish English proves difficult. It is clear that more focus must be put on linguistic diversity for language models to be more useful for such (often marginalised) varieties.

Further research into BERT's representations of modal sense may focus on non-categorical representations of modal sense. Those who have annotated modal sense can attest that it is not always very clear-cut, and often, more than one interpretation of modal sense can be perceived as valid. As BERT embeddings are continuous - only our classification forces them into categories - researchers may want to investigate whether non-continuous BERT embeddings of modal verbs also match human annotators' certainties or disagreements.

Acknowledgements

We would like to thank our student assistants Hannes Gröner and Bastian Bunzeck for their diligent work annotating the modal verb senses in VEM. Without them, this work would not have been possible. We also thank the comments and suggestion of the anonymous reviewers – they have contributed to improving our work. Part of this research was carried out as part of the SAIL project which is funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia under the grant no NW21-059A.

References

Laura Aina, Kristina Gulordava, and Gemma Boleda. 2019. Putting words in context: LSTM language models and lexical ambiguity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3348, Florence, Italy. Association for Computational Linguistics.

Marianna Apidianaki. 2023. From Word Types to Tokens and Back: A Survey of Approaches to Word Meaning Representation and Interpretation. *Computational Linguistics*, pages 1–59.

Rishabh Bhardwaj, Navonil Majumder, and Soujanya Poria. 2021. Investigating gender bias in BERT. *Cognit. Comput.*, 13(4):1008–1018.

Peter Collins, Ariane Macalinga Borlongan, and Xinyue Yao. 2014. Modality in Philippine English: A diachronic study. *Journal of English Linguistics*, 42(1):68–88.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Aina Garí Soler and Marianna Apidianaki. 2021. Let's play mono-poly: BERT can reveal words' polysemy level and partitionability into senses. *Transactions of the Association for Computational Linguistics*, 9:825–844.

Beke Hansen. 2018. *Corpus Linguistics and Sociolinguistics: A Study of Variation and Change in the Modal Systems of World Englishes*. Brill, Leiden, The Netherlands.

Marianne Hundt. 2009. Colonial lag, colonial innovation or simply language change? In Günter Rohdenburg and Julia Schlüter, editors, *One language, two grammars?*, pages 13–37. Cambridge University Press.

Braj B. Kachru. 1985. Standards, codification and sociolinguistic realism. The English language in the Outer Circle. In Randolph Quirk and H.G. Widdowson, editors, *English in the World. Teaching and Learning the Language and Literatures*, pages 11–30. Cambridge University Press.

Merja Kytö. 2019. English in North America. In Daniel Schreier, Marianne Hundt, and Edgar W.Editors Schneider, editors, *The Cambridge Handbook of World Englishes*, Cambridge Handbooks in Language and Linguistics, pages 160–184. Cambridge University Press.

Bo Li, Mathieu Dehouck, and Pascal Denis. 2019. Modal sense classification with task-specific context embeddings. In *ESANN 2019 – 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 1–6, Bruges, Belgium.

Yang Liu, Xiaohui Yu, Bing Liu, and Zhongshuai Chen. 2014. Sentence-level sentiment analysis in the presence of modalities. In *Computational Linguistics and Intelligent Text Processing*, pages 1–16, Berlin, Heidelberg. Springer Berlin Heidelberg.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47(2):387–443.

Lucia Loureiro-Porto. 2019. Grammaticalization of semi-modals of necessity in Asian Englishes. *English World-Wide*, 40(2):115–143.

Li Lucy and David Bamman. 2021. Characterizing English variation across social media communities with BERT. *Transactions of the Association for Computational Linguistics*, 9:538–556.

Ashutosh Malhotra, Erfan Younesi, Harsha Gurulingappa, and Martin Hofmann-Apitius. 2013. 'HypothesisFinder:' a strategy for the detection of speculative statements in scientific text. *PLoS Computational Biology*, 9(7):e1003117.

Ana Marasović, Mengfei Zhou, Alexis Palmer, and Anette Frank. 2016. Modal sense classification at large: Paraphrase-driven sense projection, semantically enriched classification models and cross-genre evaluations. In *Linguistic Issues in Language Technology, Volume 14, 2016 - Modality: Logic, Semantics, Annotation, and Machine Learning*. CSLI Publications.

Ana Marasović, Mengfei Zhou, and Anette Frank. 2019. *The MSC Data Set*. heiDATA.

Filip Miletic, Anne Przewozny-Desriaux, and Ludovic Tanguy. 2021. Detecting contact-induced semantic shifts: What can embedding-based methods do in practice? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10852–10865, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Roser Morante and Walter Daelemans. 2009. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the BioNLP 2009 Workshop*, pages 28–36, Boulder, Colorado. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Paul Portner. 2009. *Modality*. Oxford University Press.

Valentina Pyatkin, Shoval Sadde, Aynat Rubinstein, Paul Portner, and Reut Tsarfaty. 2021. The possible, the plausible, and the desirable: Event-based modality detection for language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 953–965, Online. Association for Computational Linguistics.

Josef Ruppenhofer and Ines Rehbein. 2012. Yes we can!? Annotating English modal verbs. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1538–1545, Istanbul, Turkey. European Language Resources Association (ELRA).

Roser Saurí and James Pustejovsky. 2012. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 38(2):261–299.

Edgar W. Schneider. 2006. English in North America. In *The Handbook of World Englishes*, pages 58–77. Blackwell Publishing Ltd.

Veronika Vincze, György Szarvas, Richárd Farkas, György Móra, and János Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(S11).

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for language understanding. *Proc. Conf. AAAI Artif. Intell.*, 34(05):9628–9635.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the International Conference on Computer Vision (ICVV)*, pages 19–27.

Dense Paraphrasing for Textual Enrichment

Jingxuan Tu[*] and **Kyeongmin Rim**[*] and **Bingyang Ye** and
Eben Holderness and **James Pustejovsky**
Department of Computer Science
Brandeis University
Waltham, Massachusetts
{jxtu,krim,byye,egh,jamesp}@brandeis.edu

Abstract

Understanding inferences from text requires more than merely recovering surface arguments, adjuncts, or strings associated with the query terms. As humans, we interpret sentences as contextualized components of a narrative or discourse, by both filling in missing information, and reasoning about event consequences. In this paper, we define the process of rewriting a textual expression (lexeme or phrase) such that it reduces ambiguity while also making explicit the underlying semantics that is not (necessarily) expressed in the economy of sentence structure as *Dense Paraphrasing (DP)*. We apply the DP techniques on the English procedural texts from the cooking recipe domain, and provide the scope and design of the application that involves creating a graph representation of events and generating hidden arguments through paraphrasing. We provide insights on how this DP process can enrich a source text by showing that the dense-paraphrased event graph is a good resource to large LLMs such as GPT-3 to generate reliable paraphrases; and by experimenting baselines for automatic DP generation. Finally, we demonstrate the utility of the dataset and event graph structure by providing a case study on the out-of-domain modeling and different DP prompts and GPT models for paraphrasing.

1 Introduction

Two of the most important components of understanding natural languages involve recognizing that many different textual expressions can correspond to the same meaning, and detecting those aspects of meaning that are not present in the surface form of an utterance or narrative. Together, these involve broadly three kinds of interpretive processes: (i) recognizing the diverse variability in linguistic forms that can be associated with the same underlying semantic representation (paraphrases); (ii) identifying semantic factors or variables that accompany or are presupposed by the lexical semantics of the words present in the text, through "hidden" arguments (e.g., *"stir vigorously."*; the argument of *stir* is not in the surface form); and (iii) interpreting or computing the dynamic consequences of actions and events in the text (e.g., *slicing an onion* brings about *onion slices*).

The first of these, the problem of paraphrasing, has been addressed computationally since the early days of natural language processing (NLP). The other two mentioned above, however, are more difficult to model with current machine learning approaches, which rely heavily on explicit textual strings to model semantic associations between the elements in the input. Many Question Answering (QA) systems, for example, rely on such syntagmatic forms in the training data for modeling potential associations that contribute to completion or generation task performance. Hence, if predicates or arguments are missing, implied, or interpreted from context, there is rarely anything to encode, and consequently little to decode as output, as well. Consider the following example from the traditional paraphrasing task. The text difference between the input and output only comes from a lexical substitution, rather than the rephrasing or addition of hidden arguments.

(1) Paraphrasing:
Chop onions, saute until browned. →
***Cut** onions, saute until **done**.*

To solve this problem, some recent attempts have been made to enrich surface forms that are missing information through "decontextualization" procedures that textually supply information which would make the sentence interpretable out of its

[*] These authors contributed equally to this work.

Proceedings of the 15th International Conference on Computational Semantics pages 39–49
June 21–23, 2023. ©2023 Association for Computational Linguistics

local context (Choi et al., 2021; Elazar et al., 2021; Wu et al., 2021).

The Focus of the decontextualization is on enriching text through anaphora resolution and knowledge base augmentation, which works well on arguments or concepts that can be linked back to existing knowledge sources, such as Wikipedia. Consider the following example of the this task. It is able to decontextualize *Barilla sauce* in (2a), but does not reintroduce any semantically hidden arguments from the context in (2b), making inferences over such sentences difficult or impossible.

(2) Decontextualization:
a. Add Barilla sauce, salt and red pepper flakes. →
*Add Barilla sauce, **the tomato sauce,** salt and red pepper flakes.*
b. Simmer 2 minutes over medium heat. →
Simmer 2 minutes over medium heat.

In this paper, we argue that the problems of paraphrasing and decontextualization are closely related for the purpose of clarifying meaning through verbal, nominal, or structural restatements that preserve (and enhance) meaning (Smaby, 1971; Kahane, 1984; Mel'cuk, 1995; Mel'Čuk, 2012). We propose *Dense Paraphrasing*, the process for the enrichment of the expression through both its lexical semantics and its dynamic contribution to the text in the whole narrative, which are less focused on by other work.

Consider the DPs of the sentences from examples (1) and (2), illustrated below in (3). Compared to the aforementioned tasks, DP aims to recover the semantically hidden arguments that fit the local context of the event (e.g., *pan* for the *saute* event) or carry a broader view of the context of the text (e.g., *sauted chopped onions* shows its transformation through multiple events).

(3) DP:
Chop onions, saute until browned. →
*Chop onions **on a cutting board with a knife** to get **chopped onions**, saute **chopped onions on a pan with a spatula**, resulting in **sauted onions** until browned.*

Add Barilla sauce, salt and pepper to the saucepan. Simmer 2 minutes over medium heat. →
*Add Barilla sauce, salt and pepper to the saucepan **by hand to get sauce mixture**. Simmer the **sauce mixture** 2 minutes **in the saucepan** over medium heat to get **simmered sauce mixture**.*

We argue that our work can potentially help and complement these generation tasks by enriching the source text with information that is not expressed in the surface structure. Table 1 shows a complete dense paraphrased document that shows how DP is applied on a multi-sentence level. To show the usage of our method, we experiment with baselines of neural models for text generation tasks that involve dense paraphrased text, based on datasets that are heavily annotated with event-participant structures.

Passage: Peel and cut apples into wedges. Press apple wedges partly into batter. Combine sugar and cinnamon. Sprinkle over apple. Bake at 425 degF for 25 to 30 minutes.

Dense Paraphrased (DP'ed) Passage:
Using peeler, peel apples, resulting in peeled apples; and using knife on cutting board, cut peeled apples into peeled wedges.
Using hands, press apple wedges partly into batter in the cake pan.
Combine sugar and cinnamon in a bowl, resulting in cinnamon sugar.
Sprinkle cinnamon sugar over apple wedges in batter in cake pan, resulting in appelkoek.
In oven, bake appelkoek at 425 degF for 25 to 30 minutes, resulting in baked appelkoek.

Table 1: Example DP'ed document from our dataset. Color-coded text spans represent locations of events in the input text where dense paraphrases are generated to enrich local context. Underlined text shows the appearance of the ingredient "apple" with transformation in a chain of events. Hidden arguments are added back to the text following simple syntactic rules (e.g., *using X, do Y in/on/at Z, resulting in R*).

In the remainder of the paper, we first review related work and background (§2), and give more detailed definitions of the DP schema (§3). We then apply the DP techniques on a cooking recipe dataset to show its ability to enrich the raw text with paraphrases (§4). §5 provide details of experiments we conducted to validate the utility of the proposed methodology, along with a discussion of our results. §6 explores the case studies on applying DP on the out-of-domain data and the comparison between GPT models on the paraphrasing task. We then conclude our work in §7. The source code and data will be publicly available.

2 Related Work

There is a long history in linguistics, dating back to the early 1960s, of modeling linguistic syntagmatic surface form variations in terms of transformations or sets of constructional variants (Harris, 1954, 1957; Hiż, 1964). (Smaby, 1971) formally defines this process of preserving the meaning from lexical, phrasal, or sentential expressions E_i to E_j as paraphrasing.

For NLP uses, paraphrasing has been a major part of machine translation and summarization system performance (Culicover, 1968; Goldman, 1977; Muraki, 1982; Boyer and Lapalme, 1985; McKeown, 1983; Barzilay and Elhadad, 1999; Bhagat and Hovy, 2013). In fact, statistical and neural paraphrasing is a robust and richly evaluated component of many benchmarked tasks, notably MT and summarization (Weston et al., 2021), as well as Question Answering (Fader et al., 2013) and semantic parsing (Berant and Liang, 2014). To this end, significant efforts have gone towards the collection and compilation of paraphrase datasets for training and evaluation.

In addition to the meaning-preserving paraphrase strategies mentioned above, there are several directions currently explored that use strategies of "decontextualization" or "enrichment" of a textual sequence, whereby missing, elliptical, or underspecified material is re-inserted into the expression. The original and target sentences are compared and judged by an evaluation as a text generation or completion task (Choi et al., 2021; Elazar et al., 2021; Gao et al., 2022; Chai et al., 2022; Eisenstein et al., 2022; Tu et al., 2022b; Ye et al., 2022; Katz et al., 2022). Our work applies both strategies of paraphrasing to the procedural text domain, which is new to the field. Unlike typical paraphrase generation tasks (Zhou and Bhat, 2021) which paraphrase full sentences and favor different wording and structure, our task performs at the entity-level.

Recent studies in procedural texts focus on tracking the state of events and entities in artificial corpora from arbitrary domains (Dalvi et al., 2019; Kazeminejad et al., 2021; Tandon et al., 2020). Some works also treat recipes as a rich resource for procedural texts. (Bosselut et al., 2017; Yamakata et al., 2020) leverage structured representations of domain-specific action knowledge for modeling a process of actions and their causal effects on entities. Other works try to resolve the anaphoric relations between recipe ingredients (Fang et al., 2022; Jiang et al., 2020). While these works all create corpora suitable for their own problems, our work, in contrast, embeds enriched information of both entities and events in the recipe using dense paraphrasing.

Enrichment of VerbNet predicates can be seen as an early attempt to provide a kind of Dense Paraphrasing for the verb's meaning. In Im and Pustejovsky (2009, 2010), the basic logic of *Generative Lexicon*'s subevent structure was applied to VerbNet classes, to enrich the event representation for inference. The VerbNet classes were associated with event frames within an Event Structure Lexicon (ESL), encoding the subevent structure of the predicate. If the textual form of the verb is replaced by the subeventual description itself, classes such as `change_of_location` and `change_of_possession` can help encode and describe event dynamics in the text, as shown in (Brown et al., 2018; Dhole and Manning, 2021; Brown et al., 2022). For example, the VerbNet entry *drive* is enriched with the ESL subevent structure below:

(4) **drive** in *John drove to Boston*
se1: pre-state: not_located_in (john,boston)
se2: process: driving (john)
se3: post-state: located_in (john,boston)

Such techniques will be utilized as part of our Dense Paraphrasing strategy to enrich the surface text available for language modeling algorithms.

3 Method

In this section, we detail the procedure involved in creating DPs. The DP method can be seen as the method for creating sets of semantically "enriched, but consistent" expressions, that can be exploited by either human consumption (e.g., natural language paraphrases) or machine consumption (e.g., configurable graphs). Specifically, we currently adopt a template-based method along with heuristics to generate DPs that account for hidden entities and entity subevent structure.

Sub-Event Structure DP starts by identifying events from the text. As mentioned above, ESL represents an event as having three parts: **begin** (B_e), **inside** (I_e), and **end** (E_e). In our method, we use this subevent structure not only to track the begin and end state of an event, but to create *textual redescriptions* of the changed event arguments. To illustrate, in Table 1 the *peel* and *cut* events form a two-event sequence through the DP subevent descriptions of the beginning and ending entities (*apples* → *peeled apples* → *apple wedges*).

Hidden Arguments DP also recovers hidden arguments that are not present in the surface form of the text to ensure the richness of the subevents. The changed entities associated with the begin or end events can be either hidden or explicit. For example, the *bake* event from Table 1 has both the hidden beginning and ending entity. In addition, DP also recovers relevant arguments in the same context of the event (e.g., the *bake* event occurs in the *oven*).

4 Experiment 1: Dense Paraphrasing from annotation

We use the text data from the subdomain of cooking recipes to demonstrate the application of the DP. Compared to texts of news or narratives, procedural text such as recipes tend to be task-oriented and highly contextualized, allowing the DP to focus on the hidden information and changes that are taking place in the course of a sequence of events in the narrative. Specifically, we apply the DP on the existing Coreference under Transformation Labeling (CUTL) dataset (Rim et al., 2023). CUTL consists of a subset of 100 cooking recipes from a larger Recipe-to-Video Questions (R2VQ) dataset (Tu et al., 2022a). It contains rich annotation of the cooking-related events and entities (both explicit and hidden), as well as the coreference relations between the entities.

4.1 Event structure for Dense Paraphrasing

To prepare the CUTL dataset for the DP, we transform the annotation into a set of "events", as events are primary anchors for applying DP. Adapted from (Rim et al., 2023), we define an event as an event predicate, a set of cooking-related entities and relations. The ingredient entities are associated with the begin and end subevents (of the event predicate) and re-described to show the subevent change. An example is shown in Figure 1. The entity can be hidden or explicit, and the entity types include the EVENT-HEAD, INGREDIENT, TOOL and HABITAT. The relations include BEGINNING and ENDING for ingredients, as well as PARTICIPANT-OF for tools and habitats. Each event has only one predicative verb (EVENT-HEAD), and all the relations within the event are linked from corresponding entities to the predicate. In addition, the event must have at least one beginning ingredient entity and one ending ingredient entity. Table 2 shows the statistics of the events in the dataset. The high ratio of the hidden entities makes it effective to demonstrate the utility of the DP.

4.2 Paraphrasing Hidden Entities

In this stage, we propose a semi-automatic approach to paraphrase the hidden entities that are annotated and represented in text placeholders (`verb.RES`) from the CUTL annotation. Formally, it involves two steps: generate text realizations of the hidden entities, and paraphrase the text realization to be useful for DP or other downstream tasks. We propose two methods to create the text realization of hidden entities: prefix paraphrasing (PP) and subgraph linearization. For the latter, we apply GPT-3 (Brown et al., 2020) on the text realizations to generate paraphrases, and then compare the generated PP paraphrase, the subgraph paraphrase, and the PP text directly used as the paraphrase.

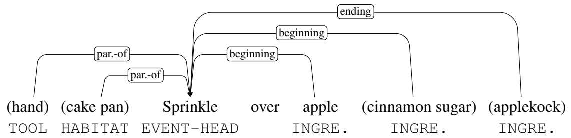


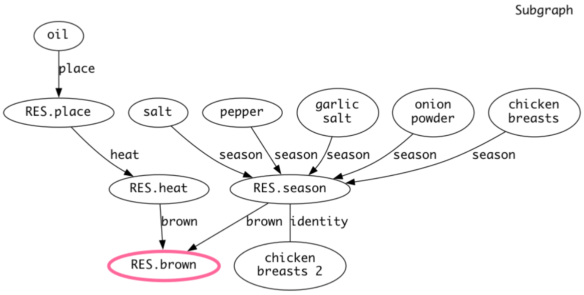
Figure 1: Annotated event example (combined R2VQ and CUTL annotations). Hidden entities are enclosed in parenthesis.

Avg. # of entities per recipe	Explicit	Hidden
EVENT-HEAD	10.6	N/A
TOOL	0.8	2.7
HABITAT	2.1	4.0
INGREDIENT (beginning)	12.0	9.4
INGREDIENT (ending)	1.0	10.4

Table 2: Statistics of the events in the CUTL dataset.

Text Realization PP is a heuristic method introduced by (Tu et al., 2022b) for question generation, which enriches the textual description of entities to reflect changes due to actions. We adopt this idea by first separating all the event predicates appearing in the data into three categories: TRANSFORMATION, LOCATION-CHANGE, and neither. For transformation events, the paraphrased entity has the format `eventPrefix + entity` (e.g. *boiled water, drained soaked peas*). For location change events or neither, the paraphrased entity has the same text form as the event input.

Given the graphical nature of the coreference graph from the DP events, we also use linearized graphs as the text realization, which has shown to be useful in various tasks such as syntactic parsing and AMR parsing (Vinyals et al., 2015; Bevilacqua et al., 2021). Specifically in our task, we extract the subgraph that is rooted in the hidden entity mention node, and then linearize it into a string literal. Examples from text realization methods are presented in Figure 2. PP converts transformation verbs into prefixes (e.g., *heated, seasoned*) and drops location change verbs (e.g., *place*). It also uses the identity link from the graph to find single entity texts that can substitute parts of the prefix-paraphrased text (e.g., `chicken breast 2` at the bottom of fig. 2 replaces the PP text for `RES.season` in the target realization.). Subgraph realization, on the other hand, records all the subevent state changes relevant to the target entity, and the events are also typed with the relations based on the verb sense and the number of beginning and ending ingredients that are connected to the verb.



Implicit mention provenance:
Brown the chicken breasts on each side. → `RES.Brown`

PP realization:

```
'browned seasoned chicken breasts and heated oil'
```

Subgraph realization:

```
['brown-AGG',
 [['season-AGG',
   ['chicken breasts', 'salt', 'pepper',
    'garlic salt', 'onion powder']],
  ['heat-TRANS', [['place-COL', ['oil']]]]]]
```

Figure 2: Text realization from PP and subgraph. Subgraph realization is wrapped and indented for readability. Event verbs are typed with: `AGG` (aggregation), `TRANS` (transformation), `COL` (change of location), etc.

Paraphrase Generation We prepare the paraphrasing data for evaluation by extracting all the ingredient mention nodes from the graph that satisfy: (1) the node is linked to a begin subevent, and to another end subevent; (2) the node has explicit text form. Such a node is connected to its placeholder text with the IDENTITY relation, as shown in Figure 2. Then we use the text of such nodes as the gold paraphrase to the hidden entity placeholder. In the end, we collected 273 gold paraphrase pairs from our dataset. Considering the scarcity of gold paraphrase in the dataset (2.7 pairs per recipe), we formalize the task as few-shot prompting and apply the GPT-3-davinci model to generate the paraphrases. Figure 3 shows the example prompts used in the GPT-3 paraphrasing methods. In each prompt, we use a single set of eight exemplars from the gold pairs and a human-created instruction on the task and how to interpret the input from different text realizations.

Paraphrase	BERTScore	Intrinsic
PREFIXP	81.15	3.08
PREFIXP-GPT	84.45 (±0.46)	3.97 (±0.08)
SUBGRAPH-GPT	**86.08** (±0.15)	**4.15** (±0.02)

Table 3: Paraphrase generation results on the gold paraphrase pairs. PREFIXP uses PP realization directly as the paraphrase; PREFIXP/SUBGRAPH-GPT uses DP/-subgraph realizations as exemplars in GPT-3 prompting.

Evaluation We use BERTScore (Zhang et al., 2019) for automatic evaluation and a 5-point Likert scale as intrinsic evaluation for the correctness, relevance, and appropriateness. For each type of realization, we perform two rounds of GPT-3 prompting with different sets of gold exemplars, and present the overall results in Table 3. While ROUGE (Lin, 2004) has been widely used in text-generation tasks, it is shown that these token-matching metrics do not align well with human annotation (Shen et al., 2022), and this finding aligns with what we observed in our experiments.

The BERTScore from all paraphrases is over 80, indicating the higher semantic similarity between the gold and model output. PREFIXP has the lowest BERTScore due to the text addition from verb prefixes and the lack of summarization ability over a list entities in the input. For intrinsic evaluation, SUBGRAPH-GPT performs better than PREFIXP-GPT, suggesting that the subgraph realization is a better resource for GPT-3 to recover and summarize the essential information in paraphrasing. PREFIXP performs the worst in the intrinsic evaluation. From the summary of annotators' feedback on the evaluation, we observe that the PP paraphrase of the entity from later steps tends to be lengthy and redundant without signaling the salient entity (average token numbers of PP paraphrase is 7.4, whereas it is 2.4 in GPT-generated paraphrases). In addition, PP paraphrase alone is less natural and less understandable to humans.[1] At the end, we validate the paraphrasing results from SUBGRAPH-GPT, and incorporate them into the following experiments.

[1] One low-scored example of the DP paraphrase: *stirred egg and water and black pepper and garlic granules.*

```
The task is to generate short and accurate paraphrase of the given noun phrases.
The input noun phrase describes the event state change of the food ingredients
through processing in a recipe, and the output paraphrase should summarize the
combination or state change of the ingredients.

input: stirred butter mixture and flour and cocoa and baking soda and salt
output: dough

[7 more exemplars]


input: squeezed horseradish
output:
```

```
The task is to generate short and accurate paraphrase of the given logical expression.
The input logical expression describes the cooking events and state change of the food ingredients
through processing in a recipe, and the output paraphrase should summarize the combination or state
change of the ingredients.

event types in the logical expression:
TRANSFORMATION: event that transforms the state, shape and etc. of an ingredient
AGGREGATION: event that combines multiple ingredients together
SEPARATION: event that separate an ingredient, or remove part of the ingredient
LOC: move the ingredient to another location

input: ['reserve-TRANSFORMATION', [['combine-AGGREGATION', ['onion', 'chilies', 'cilantro', 'salt']]]]
output: reserved onion mixture

[7 more exemplars]

input: ['squeeze-TRANSFORMATION', ['horseradish']]
output:
```

Figure 3: GPT-3 Prompt templates for the PREFIXP-GPT (top) and the SUBGRAPH-GPT (bottom).

5 Experiment 2: End-to-end DP

In this section, we present experiments of the task for automatic generation of the DP text. we explore baselines from language models and provide further insights on our data. We formalize DP generation as the task of identifying textual event mentions from cooking recipe text as well as their associated hidden entities or text mentions.

Experiment Setup We use the recent sequence-to-sequence generation model T5 (Raffel et al., 2020) as the baseline. We set the output sequence to be 'label-enclosed' text with special symbols to mark up the patterns that can be effectively processed by the models (Zhai et al., 2022). An example sequence is shown in Figure 4. We randomly sample 80 recipes for training and hold out 20 for testing. Model performance was evaluated using F1-score. We fine-tune the T5-base model on the training set, and leverage the effect from either using single sentence or aggregated sentences as the input sequence, and using additional recipe data for the augmentation.

Model Details We fine-tune the T5 text generation model (Raffel et al., 2020) to perform the task on the training set with a maximum of 512 input and out tokens. For each experiment run, we fine-tune T5-BASE model for 8 epoches on 4 NVIDIA Titan Xp GPUs. It took roughly an hour to finish the training [2]. For the augmentation setting, we map the ingredient entities that are linked with the PARTICIPANT-OF and RESULT-OF relations from the R2VQ dataset (Tu et al., 2022a) to the BEGINNING and ENDING subevents. R2VQ didn't assume the event participant/result is necessary so the mapping can only recover partial annotations under our subevent definition. In practce, we first use the entities and mapped relations from the 900 recipes as the "silver" data to pretrain the T5 model, and then fine-tune/train the pretrained T5 with the 80 recipes from the CUTL dataset.

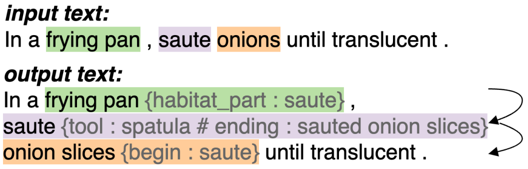

input text:
In a frying pan , saute onions until translucent .
output text:
In a frying pan {habitat_part : saute} ,
saute {tool : spatula # ending : sauted onion slices}
onion slices {begin : saute} until translucent .

Figure 4: Example of T5 model input and output for DP generation task. Each cooking role is wrapped by a pair of curly brackets ({...}). Cooking roles at the same position are separated by hashtags (#).

[2] training script adopted from https://huggingface.co/valhalla/t5-base-qa-qg-hl

	Single-T5		Agg.-T5		Agg.+Aug.-T5		
Label	E.	H.	E.	H.	E.	H.	Count
tool	71.42	60.28	72.63	61.59	**75.09**	**64.50**	73
habitat	73.62	64.28	73.87	64.69	**80.93**	**68.69**	129
ingredient (beginning)	81.33	31.92	82.18	**32.63**	**88.22**	32.13	405
ingredient (ending)	**60.03**	44.68	59.13	45.59	59.53	**46.19**	221
All	73.57	42.56	73.89	43.64	**78.27**	**44.43**	828

Table 4: DP generation results from T5 under different settings. F1 score is reported for both explicit (E.) and hidden (H.) entities. Single-T5 uses one sentence as single model input; Agg.-T5 aggregates every three continuous sentences as single input and only evaluates on the third sentence from each input; Agg.+Aug.-T5 uses the rest of 900 R2VQ recipes as augmented data for training.

Results Table 4 shows the model results on the DP generation task. Compared to Single-T5, Agg.-T5 gains a better performance (73.9/43.6 F1), suggesting the importance of contextual information from previous sentences in procedural text. Agg.+Aug.-T5 performs the best overall (78.3/44.3 F1 F1) due to the additional data from the R2VQ annotation. For individual labels, identifying hidden entities are still challenging to the baseline model, especially for the ingredient. Agg.+Aug.-T5 performs worse on hidden beginning ingredients than explicit ones by a large margin (53.1 F1). Compared to the hidden tool and habitat, hidden ingredient has more variants from the context of DP events (e.g., *onions, onion slices, sauted onions, etc*). In addition, each DP event can have multiple beginning or ending ingredients (e.g., *mix water and flour*), which also increases the difficulty of the task.

Overall, the above experiment shows that the inference and reasoning over all the hidden text remains a very challenging task to current large language models. For our data specifically, the higher ratio of the hidden entities and the entity variance from the dense paraphrasing makes it a challenging task to the model. Attempts to improve the results may include multi-task learning to generate entity types and values separately, and iterative training to utilize the data more efficiently. We further explore the DP method and data by showing the case study on out-of-domain DP text generation and GPT-3 paraphrasing.

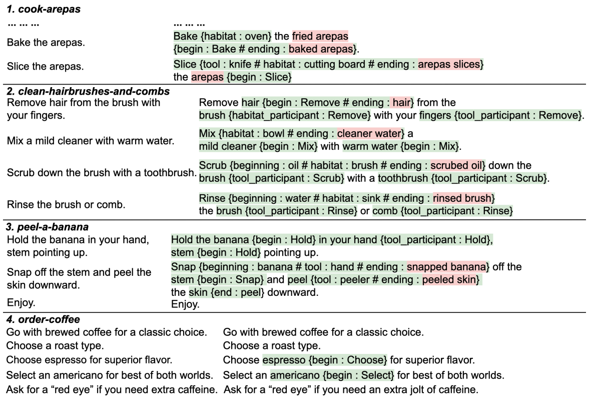
6 Case Study

6.1 Out-of-Domain DP Modeling

We explore the scenarios that the DP strategy and datasets can be adapted to raw data in the same style (e.g., procedural text) but out of the domain under a transfer learning setting. We show a case study of the results by applying the DP generation model that is fine-tuned on our training set to WikiHow articles. For this experiment, we use the articles from the WikiHow corpus curated by (Zhang et al., 2020) that is originally for the goal-step inference tasks. Specifically, we pick four articles from different domains and apply the fine-tuned DP generation model from §5 on these articles.

The generation results on the four unseen WikiHow articles are shown in Figure 5. The first article is an in-domain recipe (shortened in the Figure), so the model performs very well on identifying the relations and hidden entities. The ingredient entities also show the subevent state change through sentences (e.g., *fried arepas* to *baked arepas*). The results on the second article shows the effectiveness of the DP strategy being applied to out-of-domain data. Our defined DP event structure can be naturally transferred to text with clear steps and intermediate goals (e.g., *Mix a mild cleaner with warm water*). The model could mispredict the actual values of the hidden entities due to the limitations from the domain-specific vocabulary inventory. E.g., the predicted hidden entity is *oil* from the sentence *"Scrub down the brush ..."*. The subevent entity paraphrasing, however, is still effective. For example, the hidden result ingredient of the event *mix* is *cleaner water*. Similarly in the last sentence, we are able to generate *rinsed brush* that carries the subevent state effectively.

Compared to the first two, we find the last two articles to be more challenging to the model. Although the text is short, the third article involves rather complex spatial actions (e.g., *snap off, peel downward, etc.*) that may confuse the model. The part-whole relations of entities (e.g., *banana vs. skin vs. stem*) can also lead to semantically ambiguous subevent paraphrases such as *snapped stem / banana, peeled skin / banana*. The last article is



1. cook-arepas	
... ...	... ...
Bake the arepas.	Bake {habitat : oven} the fried arepas {begin : Bake # ending : baked arepas}.

Slice the arepas.	Slice {tool : knife # habitat : cutting board # ending : arepas slices}
	the arepas {begin : Slice}

2. clean-hairbrushes-and-combs	
Remove hair from the brush with your fingers.	Remove hair {begin : Remove # ending : hair} from the brush {habitat_participant : Remove} with your fingers {tool_participant : Remove}.
Mix a mild cleaner with warm water.	Mix {habitat : bowl # ending : cleaner water} a mild cleaner {begin : Mix} with warm water {begin : Mix}.
Scrub down the brush with a toothbrush.	Scrub {beginning : oil # habitat : brush # ending : scrubed oil} down the brush {tool_participant : Scrub} with a toothbrush {tool_participant : Scrub}.

Rinse the brush or comb.	Rinse {beginning : water # habitat : sink # ending : rinsed brush}
	the brush {tool_participant : Rinse} or comb {tool_participant : Rinse}

3. peel-a-banana	
Hold the banana in your hand, stem pointing up.	Hold the banana {begin : Hold} in your hand {tool_participant : Hold}, stem {begin : Hold} pointing up.
Snap off the stem and peel the skin downward.	Snap {beginning : banana # tool : hand # ending : snapped banana} off the stem {begin : Snap} and peel {tool : peeler # ending : peeled skin} the skin {end : peel} downward.
Enjoy.	Enjoy.
4. order-coffee	
Go with brewed coffee for a classic choice.	Go with brewed coffee for a classic choice.
Choose a roast type.	Choose a roast type.
Choose espresso for superior flavor.	Choose espresso {begin : Choose} for superior flavor.
Select an americano for best of both worlds.	Select an americano {begin : Select} for best of both worlds.
Ask for a "red eye" if you need extra caffeine.	Ask for a "red eye" if you need an extra jolt of caffeine.

Figure 5: DP generation on example WikiHow articles. The left shows the article title and steps; the right shows the model output. Green spans mark the entities and relations; red spans mark the paraphrased entities.

different from the others in the sense that it has a less clear step-goal structure and the events are not actions interacting with physical objects. These differences make texts of this type less suitable to the proposed method. In general, the case study shows the usefulness of the DP strategy and the dataset we created under a transfer learning scenario to procedural texts with the similar format. Future work includes expanding the DP evaluation on general procedural texts so that a quantitative study can be conducted.

6.2 Subgraph for GPT-3 Paraphrasing

We briefly characterize the common differences in the output paraphrases between PREFIXP-GPT and SUBGRAPH-GPT, and present several examples in Table 5. In comparison, PREFIXP-GPT tends to generate paraphrase as noun-noun components, while PREFIXP-GPT tends to generate an adjectival verb as the modifier to the entity. Score-wise, both output format are acceptable, but minor syntactic errors (mushroom[s] slices) and semantic ambiguity (meat [mixture]) are spotted from the NN components. PREFIXP-GPT also has a strong tendency to rewrite or hallucinate new text. This may be due to the fact that prefix-paraphrase has no special symbol or text structure to regulate the generation. Compared to SUBGRAPH-GPT which preserves the event type and structure in the model input, PREFIXP-GPT uses the 'flattened' text that may put extra weight on the local event that is closest to the entity to be paraphrased. Consider the gold *salad* from the table. Based on the event text ***season** with salt and pepper*, the PREFIXP-GPT generates the realization such as *seasoned pepper and salt and combined lemon juice and ...*, whic features the latest event and entities. A subgraph allows one to trace all the visited events and thus increase the model reasoning capability.

	GOLD	PREFIXP-GPT	SUBGRAPH-GPT
NN Comp.	mushrooms	mushrooms slices (4)	sliced mushrooms (5)
	cooked bacon	bacon bites (5)	chopped bacon (5)
	meat	meat mixture (4)	sauteed meat (5)
Hallucination	sewian	fried noodles (2)	fried sewian (5)
	meat	ground beef (4)	minced meat (4)
	soup	stew (3)	vegetable broth (4)
Locality	fish	marinated chunks (4)	marinated fish (5)
	salad	vinaigrette (2)	salad (5)

Table 5: Common difference between the output paraphrase from PREFIXP-GPT and SUBGRAPH-GPT, and their intrinsic scores.

6.3 Does GPT-4 solve everything?

We further explore the performance of different GPT models on the the task of paraphrase generation. In table 6, we select five examples which SUBGRAPH-GPT performs poorly on (with an intrinsic score of 3 or lower), and anecdotally compare the results with the paraphrases generated by the latest GPT-4 (SUBGRAPH-GPT4). In the first example, both GPT-3 and GPT-4 generate the same incorrect paraphrase. This might be due to the model has been trained biased towards a strong connection between the action *squeeze* and the *juice*. GPT-4 also doesn't generate ideal paraphrase on the second and the third example due to the noise from the context, e.g., *mussels* or *peas* is a more salient feature than the *water*. However, it is able to provide more details in the paraphrases (*cooked peas* v.s. *peas*). On the last two examples, GPT-4 performs better than GPT-3 by generating more accurate entities (e.g., *fillet* v.s. *fish*) and richer states (*seasoned. marinated*).

7 Conclusion

In this paper we define *Dense Paraphrasing (DP)*, the task of enriching a text fragment (lexeme, phrase, or sentence) such that contextual ambiguities are eliminated, contextual anchors or variables are supplied, and any implied arguments are made textually explicit. We outlined our DP procedure that can be applied to enrich the textual dataset, and provided insights on the transformer-based models as baselines for the DP text generation task. We presented the case study for generating DP under the out-of-domain setting, and the analysis on paraphrasing from event graphs, which show the feasibility of modeling DP and the challenges it

Context	Subgraph-GPT	Subgraph-GPT4	Gold
Prepared horseradish, squeeze it dry through a kitchen towel to get [SQUEEZE.RES]. Combine the horseradish, sour cream ...	horseradish juice	horseradish juice	horseradish
Wash mussels and de-beard, bring a pot of water to a boil to get [BOIL.RES] and cook the mussels ...	cooked mussels	de-bearded and cooked mussels	boiled water
In a large pot, bring peas and water to boil over high heat and reduce to simmer until tender to get [SIMMER.RES].	peas	cooked peas	soup
Add the salt and pepper ... place the fillets under the broiler, about 2 inches from the heat source and cook for 2 minutes to get [COOK.RES]	fish	seasoned cooked fillets	salmon fillets
Cut chicken thighs in half ...Combine the paste with the chicken and mix well; refrigerate several hours or overnight to get [REFRIGERATE.RES]	chicken patties	marinated chicken	chicken

Table 6: Output paraphrase comparison between different GPT models on five examples. Paraphrases are generated for entities represented as [VERB.RES].

poses to current large language models.

We believe that DP has the potential to help in a broad range of NLP applications. In particular, applications and tasks involving abstractive inferencing can benefit from the dynamic tracking and decontextualized redescriptions of entities appearing in a coreference chain. The notion of following an entity as it changes through a developing narrative or text can be computationally encoded using the technique described here, giving rise to a history or biographical model of an entity. We hope to extend the DP procedure to include creating vector representations of DP that can be fit into a broader range of computational models. We also intend to include reference to the "vertical typing" of an expression (type inheritance) from online resources with definitional texts, such as Wikipedia or WordNet (e.g., onion ∈ vegetable, poodles ∈ dogs). This would further enhance the utility of the resulting DP'ed data for logical inference tasks.

References

Regina Barzilay and Michael Elhadad. 1999. Using lexical chains for text summarization. *Advances in automatic text summarization*, pages 111–121.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *AAAI Conference on Artificial Intelligence*.

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*.

Michel Boyer and Guy Lapalme. 1985. Generating paraphrases from meaning-text semantic networks. *Computational Intelligence*, 1(1):103–117.

Susan Windisch Brown, Julia Bonn, Ghazaleh Kazeminejad, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2022. Semantic representations for nlp using verbnet and the generative lexicon. *Frontiers in artificial intelligence*, 5.

Susan Windisch Brown, James Pustejovsky, Annie Zaenen, and Martha Palmer. 2018. Integrating generative lexicon event structures into verbnet. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Haixia Chai, Nafise Sadat Moosavi, Iryna Gurevych, and Michael Strube. 2022. Evaluating coreference resolvers on community-based question answering: From rule-based to state of the art. In *CRAC*.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Peter W Culicover. 1968. Paraphrase generation and information retrieval from stored text. *Mech. Transl. Comput. Linguistics*, 11(3-4):78–88.

Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4496–4505, Hong Kong, China. Association for Computational Linguistics.

Kaustubh D. Dhole and Christopher D. Manning. 2021. Syn-qg: Syntactic and shallow semantic rules for question generation.

Jacob Eisenstein, Daniel Andor, Bernd Bohnet, Michael Collins, and David Mimno. 2022. Honest students from untrusted teachers: Learning an interpretable question-answering pipeline from a pretrained language model. *ArXiv*, abs/2210.02498.

Yanai Elazar, Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2021. Text-based np enrichment. *arXiv e-prints*, pages arXiv–2109.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618.

Biaoyan Fang, Timothy Baldwin, and Karin Verspoor. 2022. What does it take to bake a cake? the RecipeRef corpus and anaphora resolution in procedural text. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3481–3495, Dublin, Ireland. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, N. Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2022. Rarr: Researching and revising what language models say, using language models.

Neil M Goldman. 1977. Sentence paraphrasing from a conceptual base. *Sentence Paraphrasing from a Conceptual Base*, pages 481–507.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Zellig S Harris. 1957. Co-occurrence and transformation in linguistic structure. *Language*, 33(3):283–340.

Henry Hiż. 1964. The role of paraphrase in grammar. In *Monograph Series on Language and Linguistics 17*.

Seohyun Im and James Pustejovsky. 2009. Annotating event implicatures for textual inference tasks. In *The 5th Conference on Generative Approaches to the Lexicon*.

Seohyun Im and James Pustejovsky. 2010. Annotating lexically entailed subevents for textual inference tasks. In *Twenty-third international flairs conference*.

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2020. Recipe instruction semantics corpus (RISeC): Resolving semantic structure and zero anaphora in recipes. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 821–826, Suzhou, China. Association for Computational Linguistics.

Sylvain Kahane. 1984. The meaning-text theory.

Uri Katz, Mor Geva, and Jonathan Berant. 2022. Inferring implicit relations in complex questions with language models. *ArXiv*, abs/2204.13778.

Ghazaleh Kazeminejad, Martha Palmer, Tao Li, and Vivek Srikumar. 2021. Automatic entity state annotation using the VerbNet semantic parser. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 123–132, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kathleen McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10.

Igor Mel'cuk. 1995. Phrasemes in language and phraseology in linguistics. *Idioms: Structural and psychological perspectives*, pages 167–232.

Igor Mel'Čuk. 2012. Phraseology in the language, in the dictionary, and in the computer. *Yearbook of phraseology*, 3(1):31–56.

Kazunori Muraki. 1982. On a semantic model for multilingual paraphrasing. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, Marc Verhagen, Eben Holderness, and James Pustejovsky. 2023. The coreference under transformation labeling dataset: Entity tracking in procedural texts using event models. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation.

RM Smaby. 1971. Paraphrase grammars, volume 2 of formal linguistics series. dordrecht: D.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

Jingxuan Tu, Eben Holderness, Marco Maru, Simone Conia, Kyeongmin Rim, Kelley Lynch, Richard Brutti, Roberto Navigli, and James Pustejovsky. 2022a. SemEval-2022 Task 9: R2VQ – Competence-based Multimodal Question Answering. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1244–1255, Seattle, United States. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, and James Pustejovsky. 2022b. Competence-based question generation. In *International Conference on Computational Linguistics*.

Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Jack Weston, Raphael Lenain, Udeepa Meepegama, and Emil Fristed. 2021. Generative pretraining for paraphrase evaluation. *arXiv preprint arXiv:2107.08251*.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, and Gaurav Singh Tomar. 2021. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. *arXiv preprint arXiv:2112.08558*.

Yoko Yamakata, Shinsuke Mori, and John Carroll. 2020. English recipe flow graph corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5187–5194, Marseille, France. European Language Resources Association.

Bingyang Ye, Jingxuan Tu, Elisabetta Jezek, and James Pustejovsky. 2022. Interpreting logical metonymy through dense paraphrasing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Weihe Zhai, Mingqiang Feng, Arkaitz Zubiaga, and Bingquan Liu. 2022. Hit&qmul at semeval-2022 task 9: Label-enclosed generative question answering (leg-qa). In *SEMEVAL*.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Towards Unsupervised Compositional Entailment with Multi-Graph Embedding Models

Lorenzo Bertolini Julie Weeds and **David Weir**
University of Sussex
Brighton, UK
{l.bertolini, juliwe, d.j.weir}@sussex.ac.uk

Abstract

Compositionality and inference are essential features of human language, and should hence be simultaneously accessible to a model of meaning. Despite being theory-grounded, distributional models can only be directly tested on compositionality, usually through similarity judgements, while testing for inference requires external resources. Recent work has shown that knowledge graph embeddings (KGE) architectures can be used to train distributional models capable of learning syntax-aware compositional representations, by training on syntactic graphs. We propose to expand such work with Multi-Graphs embedding (MuG) models, a new set of models learning from syntactic and knowledge-graphs. Using a phrase-level inference task, we show how MuGs can simultaneously handle syntax-aware composition and inference, and remain competitive distributional models with respect to lexical and compositional similarity.

1 Introduction

Drawing an inference over structured text is considered to be a basic aspect of natural language understanding (Pavlick and Callison-Burch, 2016). To build structured meaning, humans rely on compositionality (Frege, 1892; Mollica et al., 2020). For this reason, much work has underlined the connection between composition, the construction of complex meaning from smaller units, and inference (MacCartney and Manning, 2008; Baroni et al., 2012; Pavlick and Callison-Burch, 2016; Pavlick and Kwiatkowski, 2019). With respect to recently popularised large language models (LLMs) like BERT (Devlin et al., 2019), the literature has produced contrasting evidence, both against (Keysers et al., 2020; Do and Pavlick, 2021; Bertolini et al., 2022) and in support (Brown et al., 2020; Nie et al., 2020) of these models being able to simultaneously handle composition and inferences with little to no supervision. However, most of the work has focused on sentence-level inference. Multiple pieces of evidence have shown that, when solving such tasks, models strongly rely on biases and spurious correlations in the benchmarks (Poliak et al., 2018; Dasgupta et al., 2018; McCoy et al., 2019). To address this issue, authors proposed to focus on phrase-level tasks (e.g., Yu and Ettinger (2020, 2021); Bertolini et al. (2022)). In particular, Bertolini et al. (2022) showed that LLMs learn to make robust compositional inferences regarding adjective-noun phrases only with direct supervision, and linked this ability to non-lexical subword units. While computationally effective, this solution is poorly grounded in linguistic and cognitive theories.

Recently, Bertolini et al. (2021) showed how training knowledge-graph embedding (KGE) architectures on syntactic graphs leads to distributional models able to learn syntax-aware compositional representations. While these models theoretically satisfy the compositionality principle (Frege, 1892; Partee et al., 1995), like LLMs, they still require external resources or training to be evaluated on inference. In this work, we propose to expand syntactic-graphs distributional models (SyG) with knowledge-graph, and propose Multi-Graph (MuG) models. We argue that, by training on both data sources, MuG could inherit compositional abilities from SyGs, and learn to manipulate the *hypernym* relation from KGE. Thus, MuG models should be able to handle both composition and inference simultaneously in the form of compositional entailment, in a fully unsupervised manner. Since previous results found rotation to better encode hierarchical relations (Chami et al., 2020) such as entailment, and reflection to be most suitable to represent syntactic information (Bertolini et al., 2021), we hypothesize that an attention-based hybrid model will be the best architecture to simulta-

Proceedings of the 15th International Conference on Computational Semantics pages 50–61
June 21–23, 2023. ©2023 Association for Computational Linguistics

neously handle compositionality and inference.

Our contributions are four-fold. First, we introduce Multi-Graph (MuG) models, a new set of embedding models trained on syntactic and knowledge-graphs. Second, we provide evidence that, under the correct combination of training method and architecture, MuG models can tackle compositional entailment, using a syntax-aware composition. Third, we propose a detailed analysis describing the behaviour of the best MuG model, clearly showing how the three macro classes of adjectives and the structure of the inference shape the behaviour of the model. Fourth, we investigate which abilities, in terms of distributional and knowledge-based, are inherited by MuGs. We show that MuGs are competitive distributional models, but struggle under a graph-related task, likely due to an incompatibility with respect to negative samples rate during training.

The paper is organised as follows. Section 2, reviews the related work on compositional entailment and different embedding models. In Section 3, we lay out the methodology behind MuG models, in terms of training methods, compositional entailment predictions, and model's parameters (such as the composition strategy). Section 4 describes training and evaluation datasets, and other implementation details. In Section 5, we present and analyse results on compositional entailment, graph completion and distributional similarity. Section 6 presents a discussion on the overall findings of the work, and how they fit in the current literature.

2 Background and Related Work

Compositional entailment A niche of work exists on phrase-level entailment, mostly focusing on adjective-noun (AN) phrases (e.g., *brown dog* entails (⊨) *animal*). Baroni et al. (2012) used non-intensional adjectives solely in the form of AN ⊨ N. Kober et al. (2021) used AN phrases as a data augmentation technique to improve lexical entailment classification. Recently, Bertolini et al. (2022) introduced PLANE, a benchmark to train and evaluate models on phrase-level adjective-noun entailment, which will be used in this work. All instances of the dataset are built out of true (noun (N), hypernym (h(N))) pairs, modified by an adjective (A). Items can take three entailment structures (or inference types (ITs)): AN ⊨ N, AN ⊨ h(N), and AN ⊨ Ah(N). Instances are then automatically annotated using rules defined by the three classes of English adjectives: intersective (I), subsective (S) and intensional (O). Table 1 summarises PLANE's instances, classes, and annotation schema. The work showed how LLMs struggle to solve PLANE without supervision, and that the mechanism supporting out-of-distribution generalisation is poorly linguistically grounded, as it notably depends on non-linguistic subword tokens.

Inference Type	Intersective	Subsective	Intensional
1 AN ⊨ N	✓	✓	✗
2 AN ⊨ h(N)	✓	✓	✗
3 AN ⊨ Ah(N)	✓	✗	✓

Table 1: PLANE annotation rules. Schema of how the interaction between each adjective class and inference type shapes the positive (✓) - negative (✗) value of a true noun (N) – hypernym (h(N))) entailment (⊨) pair.

Knowledge-graph Embedding (KGE) Models Multiple ways of encoding hypernymy and other entailment relationships with different transformations, including rotation and reflection, have been investigated (Balažević et al., 2019; Chami et al., 2020). Proposed models learn representations of entities and representations that encode a mapping of entities to their hypernyms. For example, we can learn representations of the entities *dog* and *animal* and the relationship *ISA* such that when the *ISA* transformation is applied to the representation of *dog*, we would expect to be close to the representation of *animal*. Among all, hierarchical relationships such as hypernymy were found to be best modelled by rotations (Chami et al., 2020).

Bertolini et al. (2021) showed that syntax-sensitive composition of adjective-noun phrases can be carried out by modelling syntactic relationships with geometric transformations. To form a phrase's encoding, such as *brown dog*, one or more of the constituent representations (according to the syntactic relationship between them) is transformed before combination. The work also tested multiple transformations, including attention, and found reflection to best model syntax.

Joint-Embedding models (JEM) Our work bears resemblances with work merging textual and KG data (Alsuhaibani et al., 2018; Roy and Pan, 2020; Wang et al., 2020). A more detailed survey of recent work in this area is provided in Roy and Pan (2020). Here, we note that Toutanova et al. (2015) add specific syntactic-triplets extracted from text, like $(Obama, \texttt{nsubj}, President)$ to the original

KG. These injected triplets are hence used only as a form of data augmentation. Alsuhaibani et al. (2018) expand `GloVe`'s (Pennington et al., 2014) loss to incorporate knowledge from a KG, creating a new joint objective function. In contrast to our work, the scope was to use KG data to enhance distributional embeddings. Wang et al. (2020) propose a robust attention-based model that incorporates textual and KG information in parallel, using one encoder for each source. A mutual attention component is then used to combine the outputs of the two encoders. In this case, similarly to our experimental setting, the scope was to improve the performance from the KGE perspective. Shwartz et al. (2016) propose to augment a hypernym classification model using a PathLSMT, based on syntactic relations. Vashishth et al. (2019) incorporated syntactic and semantic graphs using a large graph convolutional network. However, the two modalities were never mixed within the same architecture, since joint models produced poor results.

3 Methodology

3.1 Multi-Graph (MuG) Models

Most mixed-sourced approaches use different architectures or objectives to model each source of data. Here, we propose to use the same model to encode two types of graphs, syntactic and knowledge-based. Specifically, we propose the Multi-Graph (MuG) Model which will be used to simultaneously encode entailment relationships from knowledge-graphs and distributional relationships from syntactic corpus data. While previous work has shown that these relationship types can be encoded independently in models based on geometric transformations (Chami et al., 2020; Bertolini et al., 2021) we propose a training method which will allow a single model to encode both types of relationship and thus use each to generalise the other. For example, if a model knows that *vehicle* is a hypernym of *car*, can it learn from the syntactic relationships in parsed corpus data, what predictions to make about the hypernyms of *red car*, *small car* and *fake car*?

To investigate which architecture is better suited to learn a MuG model, we study the three KGE architectures introduced by Chami et al. (2020), namely RotE (rotation), RefE (reflections) and AttE (which uses attention to combine rotations and reflections). Since rotation was found to best encode KG relations (Chami et al., 2020), and reflection to better model syntax (Bertolini et al., 2021), we expect that an AttE combining both rotation and reflection will be the best architecture for a MuG model.

3.2 Training Methods

To train Multi-Graph models (MuGs), we propose two training methods, `static` and `altern`, using the same architecture and weights, yet separately considering the two data sources in the training phase.

`static` Straightforwardly, `static` trains a MuG model by feeding it first one complete data source and then the other. Specifically, a selected model is first trained with syntactic graphs and then with the knowledge-graph.

`altern` The `altern` method takes a dynamic approach to the training. Training is alternated at regular intervals between the two different data sources. This adds an extra hyperparameter to the model, *every*, which we have kept stable at 5 samples, that dictates the frequency with which the two training data sources alternate. All other model hyperparameters (e.g., total epochs) are kept stable and equally distributed across the data sources. Note that `static` could be considered as an extreme version of `altern` where *every* is set to the size of the first training data source multiplied by the number of epochs.

3.3 Predicting compositional entailment

Compositional entailment is framed as a binary classification task where models have to label (c_1, c_2) tuples such as *(red car, vehicle)*. To score these tuples we propose to make use of each architecture's scoring $s(head, relation, tail)$ and sigmoid $(\sigma(\cdot))$ functions. The proposed classification function is presented in Equation 1:

$$C(c_1, r, c_2) = \begin{cases} 1 & \text{if } \sigma(s(c_1, r, c_2)) >= 0.5 \\ 0 & \text{else} \end{cases} \tag{1}$$

$s(h, r, t)$ is the model-specific scoring function (see Chami et al. (2020); Bertolini et al. (2021)). r is always considered to be the $hyponym$ relation. Given the nature of the task, one or both of each (c_1, c_2) tuple components can contain a phrase. To generate these, we use the composition strategies from Bertolini et al. (2021) (adopting average instead of simple sum):

add simple addition: constructed by averaging the base representations of the constituent words

Rh Root-as-head: the syntactic root of the phrase is seen as the head of the dependency triple (e.g., <*dog*,`amod`,*brown*>) and is modified by the geometric transformation in the composition process

Rt Root-as-tail: the syntactic root of the phrase is seen as the tail of the dependency triple (e.g., <*brown*,$\overline{\texttt{amod}}$,*dog*>) and is not modified by the geometric transformation in the composition process

BiD Bi-directional: constructed by adding the representations obtained using **Rh** and **Rt**

3.4 Syntax Modelling

In contrast to Bertolini et al. (2021), we consider the composition strategy as another interchangeable aspect of a MuG model. The decision traces back to the difference between the two forms of the compositionality principle (Partee et al., 1995). If syntax is indeed a crucial feature of compositionality, then a model with a syntax-aware composition will yield better results. Otherwise, no differences should be observed.

4 Experimental Setup

Our investigation focuses on two main questions. First, can MuGs in fact manipulate both composition and inference? To test this, we will compare MuG and KGE models on a compositional entailment task. Second, what ability, if any, will be penalised or completely sacrificed, in order for a model to tackle compositional entailment? To answer this question, MuGs will be compared to KGE on a standard graph completion task, and to distributional models trained on syntactic graphs (SyGs) on multiple similarity benchmarks.

4.1 Tasks and Benchmarks

PLANE To test MuG and KGE on compositional entailment, we sample five validation and test splits from the portion of PLANE (Bertolini et al., 2022) that contains items also included in WN18RR (Bordes et al., 2013) and `text8`, available here[1]. Since preliminary experiments showed all models heuristically assigned a positive label to items with inference type 3 (e.g. *red car* $\models$ *red vehicle*), we only sampled items with inference types 1 (AN $\models$ N) and 2 (AN $\models$ h(N)). In each split, the ratio of positive and negative items is kept balanced, and so is each (noun, hypernym) tuple for every adjective.

KG and Similarity Judgements To compare MuG and KGE models on the uni-gram level, we adopt a standard filtered graph completion task (Chami et al., 2020). The performance of SyG and MuG models is compared using the same benchmarks from Bertolini et al. (2021). These include four lexical similarity tasks (Simlex (Hill et al., 2015), MEN (Bruni et al., 2014), WS353-sim, WS353-rel (Agirre et al., 2009)), and a compositional one (ML10) (Mitchell and Lapata, 2010), further divided into three syntactic classes (Adjective-Nouns, Verb-Objects, Noun-Nouns).

4.2 Implementation

We adopt the source code from Chami et al. (2020) to train each model. Using the hyperparameters from Chami et al. (2020) and Bertolini et al. (2021), we trained a set of three architectures: AttE, RefE, RotE. As training data for each MuG model, we follow Bertolini et al. (2021) and adopt a sense-stripped version of WN18RR as KG, and a parsed version of `text8` as syntactic graph. We use PLANE validation splits to tune hyperparameters for each combination of training method (KGE, MuG-`altern`, MuG-`static`), architecture (AttE, RefE and RotE), and composition strategy (add, Rh, Rt, BiD). Best hyperparameters are presented in Appendix A. Experiments are run on an NVIDIA GeForce RTX 3090 GPU.

5 Results

5.1 Compositional entailment

Table 2 reports average accuracies (± standard error) obtained by different models on the five test sets generated from PLANE. The close-to-random performance (50%) observed for KGE models — trained solely with the knowledge-graph — is to be expected, since the overlap between training data and PLANE is fairly scarce, especially with respect to adjectives. Furthermore, Bertolini et al. (2021) already showed how KGE models perform poorly on compositional benchmarks, especially with respect to adjective-noun phrases.

Overall, MuG models perform only marginally better than KGEs. The best-performing model is based on the attention architecture, trained with the `altern` method and makes use of a syntax-aware composition strategy (Rh). These results are in

[1] `https://github.com/lorenzoscottb/IWCS_2023`

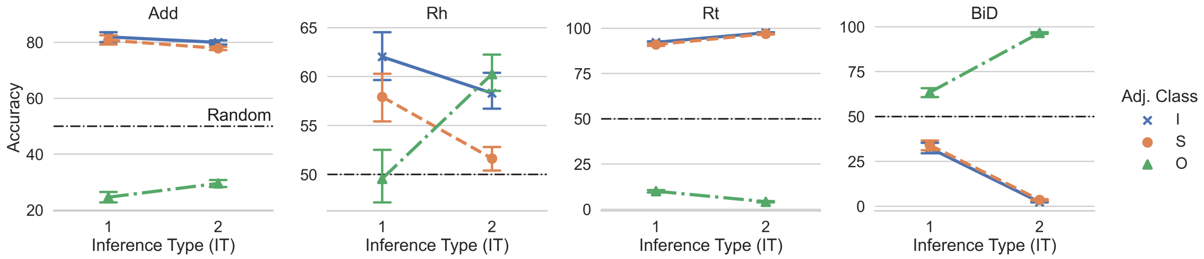


Figure 1: Models analysis. Break-down of the different AttE-altern models performance (mean accuracy ± standard error from different test splits), divided by adjective class (hue), composition strategy (columns) and inference type (IT, x-axis).

Method	Model	Composition	Accuracy
KGE	AttE	add	49.8 ± 0.2
	RefE	add	51.1 ± 0.2
	RotE	add	50.9 ± 0.2
MuG-altern	AttE	add	53.9 ± 0.4
		Rh	56.2 ± 0.5
		Rt	50.7 ± 0.1
		BiD	49.1 ± 0.2
	RefE	add	51.3 ± 0.5
		Rh	51.6 ± 0.4
		Rt	50.1 ± 0.0
		BiD	45.1 ± 0.4
	RotE	add	51.1 ± 0.3
		Rh	51.7 ± 0.4
		Rt	50.2 ± 0.0
		BiD	45.2 ± 0.7
MuG-static	AttE	add	51.9 ± 0.4
		Rh	53.3 ± 0.3
		Rt	51.5 ± 0.3
		BiD	47.1 ± 0.3
	RefE	add	53.3 ± 0.3
		Rh	53.4 ± 0.2
		Rt	50.9 ± 0.1
		BiD	47.7 ± 0.4
	RotE	add	53.6 ± 0.3
		Rh	54.2 ± 0.2
		Rt	50.7 ± 0.1
		BiD	47.4 ± 0.4

Table 2: Compositional entailment results. Accuracy scores (mean ± standard error) obtained by different combinations of training methods, architectures and composition strategies on different test–splits, generated from PLANE.

line with the two main hypotheses, suggesting that attention would better handle the two sources of data and that explicitly modelling syntax leads to more reliable compositional representations. Interestingly, the very same syntax-aware strategy (Rh) is also used by RotE-static, which seems to be the second-best performing model. However, aside from AttE-altern-Rh (and RotE-static-Rh) MuG models seem to generally struggle to correctly classify an item for compositional entailment. Hence, we now propose an in-depth analysis of what seems to be the best MuG model, comparing its behaviour to other AttE-altern models (i.e., tuned with different composition strategies), to better understand its prediction processes.

Model analysis Figure 1 breaks down the performance of AttE-altern models by composition strategies (columns), adjective class (hue) and inference type (x-axis). Overall, the figure shows that aside AttE-altern-Rh, models present a strongly heuristical behaviour, as suggested by the widespread lack of per-split variance (error bars). More specifically, add and Rt models produce almost exclusively positive predictions, as suggested by the very high performance with intersective (I) and subsective (S) adjectives. AttE-altern-BiD predictions seem to be slightly affected by the inference types (IT), fluctuating between random (under IT 1), and only-negative predictions (under IT 2). On the contrary, AttE-altern-Rh's results appear notably more complex, and suggest a strong interaction between inference type and adjective class, at least with respect to subsective and intensional adjectives. Recall that, since we have focused on IT 1 (AN $\models$ N) and 2 (AN $\models$ h(N)), intersective (I) and subsective (S) adjectives always have a positive label, while intensionals (O) are always associated with a negative label. As shown, when dealing with intersective (I) adjectives, the model is minimally impacted by the IT. The performance remains well above chance with items like *red car* $\models$ *car* (IT 1) and a *red car* $\models$ *vehicle* (IT 2).

On the other hand, the performance is significantly shaped by the inference type when instances contain subsective (S) or intensional (O) adjectives. More specifically, the performance of intensional (O) adjectives, always associated with negative la-

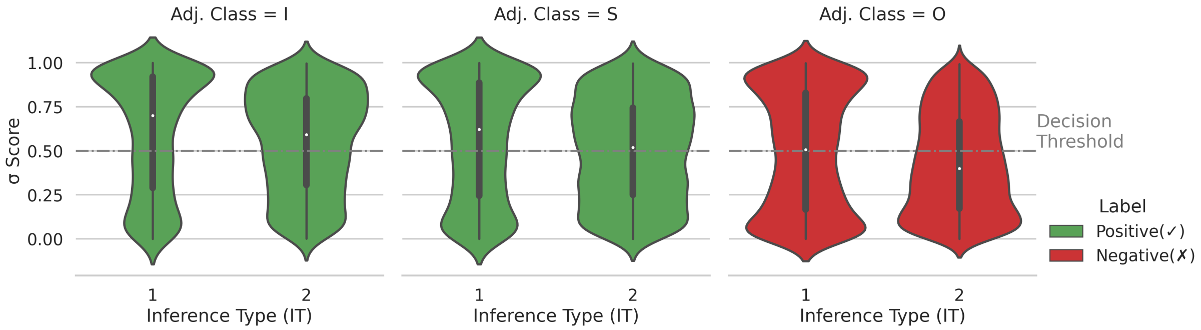


Figure 2: Distribution of the predicted scores of AttE-`altern`-Rh, divided by adjective class (columns), inference type (x-axis), and labels (hue). Dashed lines indicate the decision threshold, as in Equation 1.

bels, jumps from random to 60% moving from IT 1 to IT 2. In other words, the model better identifies scenarios like *fake car* $\not\models$ *vehicle* than *fake car* $\not\models$ *car*. The opposite happens for subsective (S) items. The model is better at classifying instances like *big car* $\models$ *car* than *big car* $\models$ *vehicle*. The fact that intersective (I) and intensional (O) adjectives produce virtually identical results on IT 2 instances, despite carrying opposite labels, while subsective's (S) performance drops to chance (although having the same label as intersective's adjectives) suggests two conclusions. First, the model's behaviour is not random or heuristical. Second, in contrast with previous evidence (Boleda et al., 2013), the theoretical distinction between adjective classes is likely reflected in the model's representations.

To understand if similar results derive from similar behaviours, Figure 2 summarises the model's prediction distribution after applying the sigmoid function in Equation 1. The plots of Figure 2 show two distinct patterns. Considering inference type 1 (i.e., AN $\models$ N), a large number of scores are towards the boundaries of the interval, generating peaky distributions. Distributions become increasingly bimodal moving through the three adjective classes (I, S and O). This suggests the model is often reasonably confident about the decision being made. On the other hand, the predictions under IT 2 (i.e., AN $\models$ h(N)) generate notably flatter distributions. Intersective (I) and intensional (O) adjectives do maintain a peak towards one of the boundaries, but the model is much less confident about decisions on IT 2 for all adjective classes.

We will now investigate if MuG models in general (i.e., not just AttE-`altern`-Rh) remain competitive with their KGE and SyG counterparts, starting with graph completion (Section 5.2), followed by distributional similarity benchmarks (Sections 5.3 and 5.4).

5.2 Graph completion

Figure 3 compares the performance of KGE and MuG models on the graph completion task. Error bars report the standard error obtained from collapsing MuG models tuned on different composition strategies. Overall, KGEs always outperform MuG models, while MuGs trained with the `static` method appear to notably outperform the ones trained with the `altern` method. This suggests that the recency of the KG training (i.e., the `static` method) is indeed influential in obtaining good results in the graph completion task. Figure 6 further breaks down the results, and compares the performance of MuGs against the amount of negative samples used in training. For comparison with the main main results (Figure 3), a dashed grey line reports the best score obtained by a KGE model.

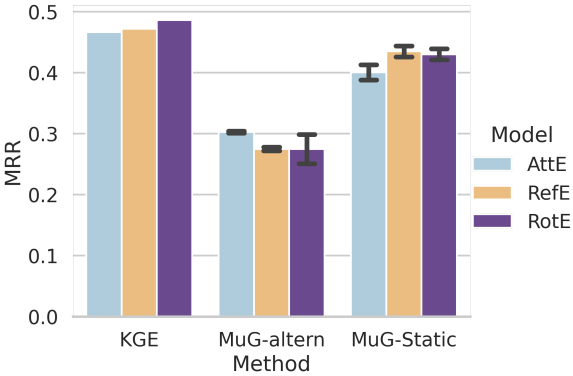


Figure 3: Mean reciprocal rank (MRR) scores of KGE and MuG models on the graph completion task. Error bars report standard error, obtained collapsing models trained with different composition strategies.

The figure shows how the optimal performance of the two training methods is reached at very different amounts of negative samples. Mug-`static`

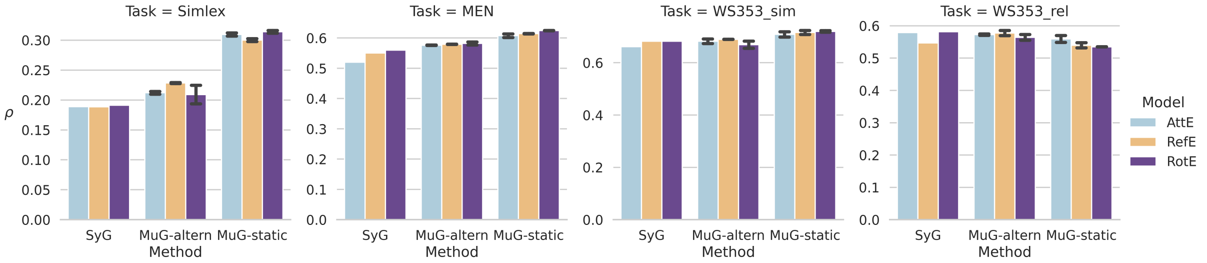


Figure 4: Spearman ρ for SyG and MuG models on word–level similarity judgement tasks. Error bars report standard error obtained collapsing models tuned on different composition strategies.

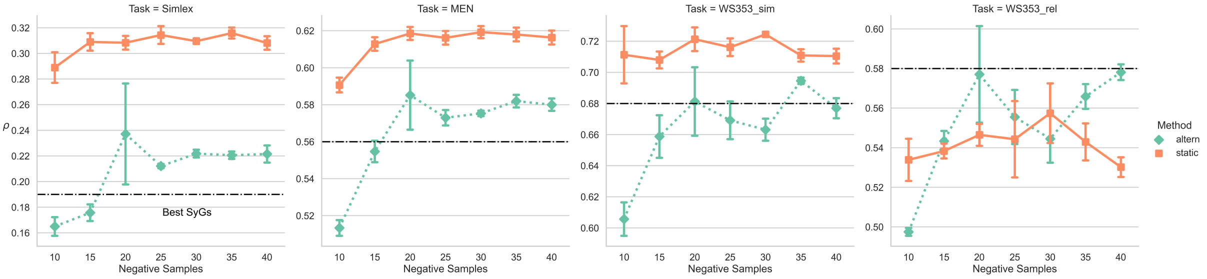


Figure 5: Lexical similarity with respect to the negative samples used during training. For comparison with Figure 4, a dashed black line outlines best results obtained by a SyG model.

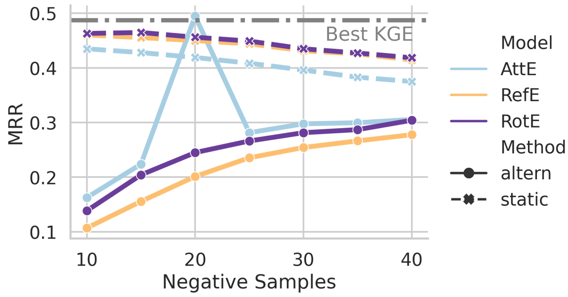


Figure 6: Graph completion analysis with respect to negative samples used during training.

models require few negative samples and are negatively affected by increasing amounts. On the other hand, the performance of models trained with the altern method increases with the number of negative samples used for training. However, other than a seemingly spurious peak at 20 negative samples, the performance obtained by altern models remains far from competitive. In line with results from Chami et al. (2020), RotE and RefE outperform AttE with the static method. Lastly, we note that the best AttE model from Table 1 is not the outlier observed in Figure 6.

5.3 Lexical similarity

We now consider whether MuGs remain competitive with SyGs (i.e., distributional models trained with syntactic graphs). Spearman's ρ is used to measure the correlation between model's predictions and human judgements on similairty benchmarks. The first comparison is presented in Figure 4. Results are divided with respect to training methods (x-axis) and trained models (hue). Error bars reflect the standard error produced by MuG models tuned with different composition strategies. Overall, MuGs tend to outperform SyGs, especially MuG-static, suggesting that KG data helps with lexical similarity tasks. A notable exception is WS353-rel, which uses relatedness (e.g., *journey* is related to *car*) rather than similarity. The KG training data is taken from WordNet, thus including many examples of hypernym/hyponym pairs which one might expect to help more with similarity. However, Bertolini et al. (2021) found a generally poor performance of KGE models on lexical similarity tasks. Altogether, these results suggest that, even in the static training, the KG data and distributional information were successfully merged, leading to a performance which cannot be achieved by one of the data sources on their own.

Similarly to Section 5.2, Figure 5 shows how the number of negative samples impacts the performance. In this case, both training methods are positively impacted by higher negative samples, although the effect remains more marked for altern models. WS353-rel aside, static models appear to consistently outperform SyGs and

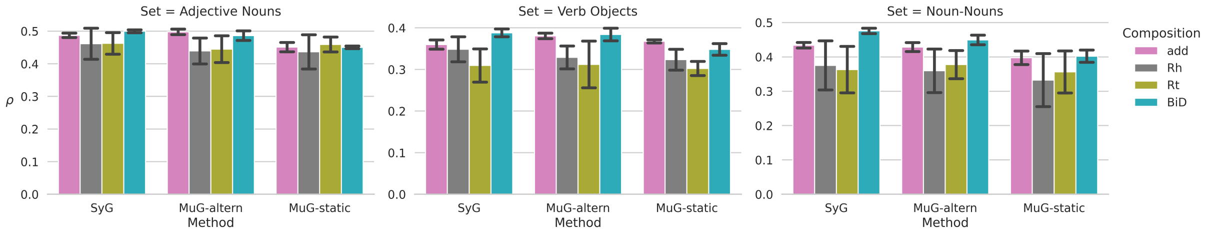


Figure 7: Spearman ρ for SyG and MuGs (divided by composition strategy) on the different subsets of the ML10 dataset. Error bars report standard errors obtained collapsing results from different architectures.

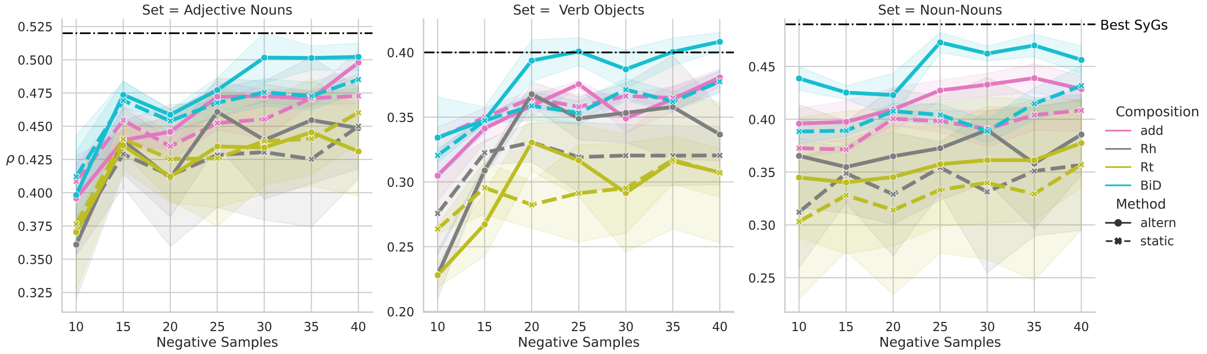


Figure 8: Compositional similarity with respect to number of negative samples used during training. For comparison with Figure 7, a dashed black line outlines best results obtained by a SyG model.

seem notably more robust, while `altern` models require a high number of negative samples.

5.4 Compositional similarity

A similar analysis is proposed for compositional similarity. Figure 7 summarises the results of best-performing models on PLANE against SyG models on the ML10 datasets. Results are split between the adjective-noun (AN), verb-object (VO), and noun-noun (NN) subsets. In this case, the focus is on training methods (x-axis) and composition strategy (hue). Compared to lexical similarity results, MuG models don't seem to outperform SyG models, but they remain a competitive alternative. Contrary to the previous results, the best performance with respect to MuG models is achieved via the `altern` training method. With respect to composition strategy, results seem to support Bertolini et al. (2021) findings: add and BiD are the best and most stable composition strategies, with BiD outperforming add. It is interesting to note that ML10 is based on bi-gram instances (e.g., how similar *hot tea* is to *cold water*), which is comparable to PLANE instance having inference type 3 (e.g., *hot tea* $\models$ *hot liquid*), that no model could solve in the compositional entailment task (see Section 4.1). The fact that MuG-`altern` models remain competitive to SyGs on ML10 suggests that their issue under IT 3 is more related to the manipulation of the hypernym relation, rather than a systematic problem of each model.

Figure 8 presents a last negative samples analysis. For comparison, a dotted line signals the best SyGs' results. As for lexical similarity, results indicate once more that performance improves with the negative samples' rate. Note that, contrary to the results on compositional entailment, Rh's performance is fairly poor across the board.

6 Discussion

Our work introduced MuGs, a set of embedding models learnt from multiple graph-based sources. Under specific and predicted conditions (i.e., using an attention-based model and syntax-aware composition), MuGs can be shown to simultaneously tackle compositionality and inference with some success. Experiments revealed that MuGs tuned for compositional entailment are competitive distributional models, with respect to both lexical and compositional similarity, yet struggle with graph completion. Our analysis suggests that a considerable part of the trade-off can be explained by the negative samples rate used for training. The best MuG model at compositional entailment, AttE-`altern`-Rh, was tuned on validation data to 35 negative samples. As summarised by the analysis in Figures 6, 5, and 8, whilst similarity tasks, especially compositional ones, also benefit from high negative samples rate, graph completion tends to require low negative samples (and the `static` training method) to achieve the best performance.

Of note, the compositional entailment experiment presented in this work can also be interpreted with respect to knowledge-graphs. Despite a different evaluation method (accuracy instead of rank), the proposed task is a type of graph completion. The evaluation is still binary and requires the manipulation of hierarchical structures through the hypernym relation. Hence, MuGs can be interpreted as compositional KGE models.

Indeed, an LLM like BERT can achieve better results on compositional entailment as defined by PLANE. However, it can only do so with direct supervision, and relying on an effective yet not theoretically-sound mechanism (Bertolini et al., 2022). Since MuG are trained only with uni-grams, our approach to phrase-level inference (i.e., compositional entailment) is fully unsupervised, requires significantly less training data, and has a deeper connection with the principle of compositionality (Partee et al., 1995). On each compositional task, linguistically-sound word encodings composed with a syntax-aware non-linear composition strategy yielded the best performance. Moreover, when a model does not present a strongly heuristical behaviour, we found that the three adjective classes pose as many different challenges to the models, similarly to what already observed in Bertolini et al. (2022).

7 Conclusions and Future Work

In this work, we introduced Multi-Graph embedding models (MuGs), a set of models trained on syntactic and knowledge-graphs. Under specific conditions, MuGs can partially tackle compositional entailment, making use of syntax-aware composition, based on attention. We provided evidence that MuGs are competitive with distributional counterparts on lexical and compositional similarity benchmarks. Our analysis suggested that compositionality is supported by a higher number of negative samples, and connected this evidence to the low performance of MuGs on graph completion. Future work will have to primarily focus on developing a training strategy to overcome the negative samples issue, able to obtain a better integration of the two sources of data and produce a more stable performance across tasks. Lastly, MuG models will have to be tested on other types of compositional entailment (e.g., noun-noun or verb-object phrases), as well as full sentences.

Acknowledgements

This research was supported by the EPSRC project *Composition and Entailment in Distributed Word Representations* (grant no. 2129720), and the EU Horizon 2020 project HumanE-AI (grant no. 952026). We thank the anonymous reviewers for their helpful comments and suggestions.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Mohammed Alsuhaibani, Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2018. Jointly learning word embeddings using a corpus and a knowledge base. *PLOS ONE*, 13(3):1–26.

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. Multi-relational poincaré graph embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, page 4463–4473, Red Hook, NY, USA. Curran Associates Inc.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France. Association for Computational Linguistics.

Lorenzo Bertolini, Julie Weeds, and David Weir. 2022. Testing large language models on compositionality and inference with phrase-level adjective-noun entailment. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4084–4100, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Lorenzo Bertolini, Julie Weeds, David Weir, and Qiwei Peng. 2021. Representing syntax and composition with geometric transformations. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3343–3353, Online. Association for Computational Linguistics.

Gemma Boleda, Marco Baroni, The Nghia Pham, and Louise McNally. 2013. Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 35–46, Potsdam, Germany. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

E. Bruni, N. K. Tran, and M. Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online. Association for Computational Linguistics.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 1596–1601, Madison, WI.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nam Do and Ellie Pavlick. 2021. Are rotten apples edible? challenging commonsense inference ability with exceptions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2061–2073, Online. Association for Computational Linguistics.

Gottlob Frege. 1892. Über sinn und bedeutung. *Zeitschrift für Philosophie Und Philosophische Kritik*, 100(1):25–50.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Thomas Kober, Julie Weeds, Lorenzo Bertolini, and David Weir. 2021. Data augmentation for hypernymy detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1034–1048, Online. Association for Computational Linguistics.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK. Coling 2008 Organizing Committee.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Francis Mollica, Matthew Siegelman, Evgeniia Diachek, Steven T. Piantadosi, Zachary Mineroff, Richard Futrell, Hope Kean, Peng Qian, and Evelina Fedorenko. 2020. Composition is the core driver of the language-selective network. *Neurobiology of Language*, 1(1):104–134.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Barbara Partee et al. 1995. Lexical semantics and compositionality. *An Invitation to Cognitive Science: Language*, page 311–360.

Ellie Pavlick and Chris Callison-Burch. 2016. Most "babies" are "little" and most "problems" are "huge":

Compositional entailment in adjective-nouns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2164–2173, Berlin, Germany. Association for Computational Linguistics.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Arpita Roy and Shimei Pan. 2020. Incorporating extra knowledge to enhance word embedding. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4929–4935. International Joint Conferences on Artificial Intelligence Organization. Survey track.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany. Association for Computational Linguistics.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal. Association for Computational Linguistics.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3308–3318, Florence, Italy. Association for Computational Linguistics.

Yashen Wang, Huanhuan Zhang, Ge Shi, Zhirun Liu, and Qiang Zhou. 2020. A model of text-enhanced knowledge graph representation learning with mutual attention. *IEEE Access*, 8:52895–52905.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics.

Lang Yu and Allyson Ettinger. 2021. On the interplay between fine-tuning and composition in transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2279–2293, Online. Association for Computational Linguistics.

A Hyperparameters

Method	Architecture	Composition	Negative Samples
KGE	RefE	add	10
KGE	RotE	add	10
KGE	AttE	add	10
static	RefE	add	20
static	RefE	Rh	20
static	RefE	Rt	40
static	RefE	BiD	35
static	RotE	add	40
static	RotE	Rh	40
static	RotE	Rt	35
static	RotE	BiD	20
static	AttE	add	30
static	AttE	Rh	30
static	AttE	Rt	40
static	AttE	BiD	10
altern	RefE	add	40
altern	RefE	Rh	40
altern	RefE	Rt	35
altern	RefE	BiD	40
altern	RotE	add	40
altern	RotE	Rh	40
altern	RotE	Rt	35
altern	RotE	BiD	15
altern	AttE	add	40
altern	AttE	Rh	35
altern	AttE	Rt	40
altern	AttE	BiD	35

Table 3: Final hyperparameters for each model.

Gender-tailored Semantic Role Profiling for German

Manfred Klenner, Anne Göhring, Alison Yong-Ju Kim, Dylan Massey
Department of Computational Linguistics
University of Zurich
{klenner,goehring}@cl.uzh.ch

Abstract

In this short paper, we combine the semantic perspective of particular verbs as casting a positive or negative relationship between their role fillers with a pragmatic examination of how the distribution of particular vulnerable role filler *subtypes* (children, migrants, etc.) looks like. We focus on the *gender* subtype and strive to extract gender-specific semantic role profiles: who are the predominant sources and targets of which polar events - men or women[1]? Such profiles might reveal gender stereotypes or biases (of the media), but as well could be indicative of our social reality.

1 Introduction

Some verbs express a positive or negative relationship (a polar relation) between the fillers of their semantic roles. For example, we can infer from the sentence "He offended her," an even, reciprocally holding, negative relation (e.g. *against(he,her)*. Moreover, such semantic roles might bear a polar (i.e. positive or negative) load, e.g. the agent of cheating might be regarded as a negative actor, a villain. From a pragmatic point of view, it might be interesting to take a closer look at the distribution of particular role fillers or role filler groups of such verbs indicating a polar relation, namely vulnerable groups such as children (pedophilia), migrants (xenophobia), people of color (racist bias), and certain gender identities (gender bias). This could reveal interesting facts about the conceptualization and contextualization of these filler groups in the real world. Such an approach could be useful for various kinds of monitoring processes (e.g. discrimination motoring). In this short paper, we focus on gender. Our goal is to enable gender-tailored semantic profiling. On a micro level, semantic profiling strives to identify the roles that gender denoting nouns occupy, e.g. that female nouns occur quite often as patients (targets) of physical violence, while male denoting nouns often are filler of the patient role of torture or accusation. On the macro level, a general, cross-verb inventory of semantic roles like *villain, victim, benefactor, beneficiary* could be used to aggregate gender-specific conceptualization. Here, we focus on the micro level.

We introduce a classifier that determines the grammatical gender of human-denoting German nouns. We combine this with our rule-based sentiment inference system[2] (Klenner et al., 2017) which assigns two types of relations between entities: in favor of, against. Each verb of our verb lexicon expresses such a polar relation and has a source (the agent) and a target (the patient, recipient or theme) role. We filtered the output of our system for cases in which the gender classifier labeled at least one of the verb roles as male- or female-denoting[3]. With such data, we were able to filter for polar events in which men are sources and women targets (and vice versa). On the basis of statistical tests, cases are found in which female or male denoting nouns are significantly over- or underrepresented.

2 Related Work

Currently, gender classification is primarily restricted to predicting the gender of text authors of blogs, see Mukherjee and Liu (2010), or to find out whether a headline is about a man or a woman, see Campa et al. (2019).

Sun and Peng (2021) observe a gender-specific tendency to combine personal and professional events in the Wikipedia pages of celebrities, an

[1] Certainly, we do not claim that gender is a binary category; but gender-denoting nouns without explicit indications (e.g. '*') do have a binary reference that we cannot overcome.

[2] The online version can be found here: https://pub.cl.uzh.ch/demo/stancer/index.py.

[3] Thus, there is no need to assign semantic roles explicitly.

Proceedings of the 15th International Conference on Computational Semantics pages 62–66
June 21–23, 2023. ©2023 Association for Computational Linguistics

asymmetric association where e.g. women's personal events appear more often in the career section than for men. They also establish higher efficiency when extracting events (verb denotations) over analyzing raw text for detecting this gender bias.

Bias detection and debiasing, in general, are important research topics (see Stanczak and Augenstein (2021) for a survey). Researchers use metrics such as pointwise mutual information (PMI) to measure the association of words with gender (Stanczak et al., 2021). We look into cases in which both grammatical genders co-occur with a verb, i.e. when PMI cannot be used.

3 Grammatical Gender Classification

The basis for our gender classifier is the freely available resource (Klenner and Göhring, 2022) of 13,000 German nouns which were manually classified as denoting either animate or inanimate entities[4]. In order to create a gold standard for grammatical gender classification, we took a subset containing animate singular nouns and manually[5] selected those that can be used to refer to women or men (altogether 4,320). Examples of female-denoting nouns include *Schwester, Gastgeberin, Schauspielerin* (Eng. *sister, hostess, actress*, respectively). We then saw that the data was imbalanced, namely that there were more male-denoting nouns (2,830) than female-denoting ones (1,490). As such a dataset would have produced a biased classifier with better classification for male-denoting nouns, we searched for more female-denoting nouns, ultimately expanding this set to 3,700. In German, this can generally be carried out by adding the suffix *in* to the end of male-denoting nouns, e.g. *Helfer* → *Helfer**in*** (Eng. *helper*). If such a variant is found in a corpus, it is added to the female list. Since we found that female nouns in news texts are underrepresented, we decided to keep the whole list of 3,700 female nouns for learning.

In Klenner and Göhring (2022) we tested various word embeddings (GloVe, BERT,FastText) for the training of the animacy classifier (MLP, SVM, LR) and found FastText with logistic regression (LR) to perform best. Therefore, we used only FastText embeddings to train a LR model for gender-aware animacy classification. There was no need to carry out extensive experiments, since our initial model achieved a high accuracy of 97.29%. Table 1 shows the results of a random 75/25 train/test split. Female-denoting noun identification with a precision of 98.3% and a recall of 99.3% might help us to mitigate gender imbalance in news texts.

	non-actors	female	male
precision	0.967	0.983	0.973
recall	0.984	0.993	0.927
f1	0.975	0.988	0.949

Table 1: Performance of our three-way, gender-aware animacy classification model.

Note: Not all German female-denoting nouns possess the "in" ending. In fact, in our list of female-denoting nouns, 50 have endings other than "in" (e.g. *Frisöse*, Eng. hairdresser). A rather simple (rule-based) method was to classify a word with an "in" ending as a female-denoting noun. But that would produce quite some error. In a corpus of 25 million nouns, we found 67,823 words (tokens) ending with "in". For 36,247 cases of these "in"-words our classifier predicted *female*. The remaining 31,576 "in"-nouns correspond to 4,035 types. We manually classified 1,000 and found only 5 female-denoting words. Thus, the classifier does not base its decision on the suffix, though this would be a legitimate approach since FastText uses sub-word splitting. The performance of our classifier with respect to the non-"in" female-denoting nouns cannot reliably be evaluated at the moment. We leave it to future work to train models able to deal with these rarer cases.

4 Statistical Setting

Our question of interest was that of identifying an imbalance, if any, between men and women, or some gender-specific behavioral semantic profile, as portrayed in newspaper texts. We focused on men and women's roles as positive or negative actors (sources) or as being positively or negatively affected patients (targets). In particular, we looked at all polar verb instantiations, with male- and female-denoting nouns occupying the source and target roles. Then, we gathered statistics on how often a positive or negative relation between two gender-denoting nouns (e.g. a female- and a male-denoting noun) was found. We performed this for all gender permutations at the level of a single verb, but we also accumulated this over all verbs. To evaluate whether a verb is more biased

[4] download: `https://zenodo.org/record/7630043#.Y-aCU9LMJH4`

[5] The annotation task is straightforward for a native speaker; thus, only one annotator was needed.

towards male or female roles, the (prior) gender distribution in the whole data must be taken into account. In our text corpus, we found a ratio of male- (1,290,415) to female- (283,952) denoting nouns (according to the gender classifier) of about 4:1. That is, the maximum likelihood estimated probability of male denoting nouns is 0.815, that of female 0.185.

The data is binomially distributed for each role of a verb frame. For instance, if a transitive (active voice) verb has $n = 200$ instantiations (and thus 200 sources), of which 20 are female, then we determine the cumulative probability of up to 20 cases given 200 trials with $p = 0.185$ as $\sum_{i=1}^{20} binom(i, 200, 0.185)$. If this value is below $\alpha = 0.05$, then we reject H_0 and adopt H_1, i.e. we can conclude that the verb (usage) is biased, and similarly for the $1 - 0.95\%$ interval. Spelled out, H_0 claims that female (male) denoting nouns occupy source (target) verb roles according to their prior probability. If this is for some verbs rather unlikely, than H_1 is adopted saying there is a verb-role specific bias, for instance that female denoting nouns are significantly more often targets of (verbs of) physical violence than male denoting nouns.

We only looked into verbs for which a normal distribution could be approximately assumed, which is given if $n * p \geq 5$ and $n(1 - p) \geq 5$, where n is the number of cases. In our setting, this amounts to a frequency threshold of $n = 5/0.185 = 27$. For each verb above this frequency threshold, we tested the null hypothesis H_0 that male- and female-denoting nouns occupy the role of a verb according to their respective distributions in the whole corpus.

5 Empirical Results

We use data from 3 Swiss newspapers published between 2004 and 2014. Despite the medium corpus size, the cases in which a verb has 2 animate role fillers (singular male or female[6]) at the source and target positions of that verb are relatively infrequent. This low frequency can be attributed to (1) the abundance of cases written in passive voice (for which there is quite often no source indicating PP) and (2) cases in which the source or holder is a personal pronoun (which, in German, leaves the animacy status of the referent open). In German, inanimate objects might have non-neutral grammatical gender, e.g. German *Brücke* (Eng. *bridge*) is feminine. This reduces the number of instantiations, e.g. for the verb *töten* (Eng. *to kill*) the counts shrink from 26,200 to 1,110 (21,000 passive cases, 4,100 pronouns). As gender classification is done after sentiment inference, another 800 cases disappear since no or only one gender-denoting noun was found, ultimately leaving 302 cases of *töten*.

relation	source	target	#
+	male	male	30
+	male	female	5
+	female	male	6
+	female	female	2
-	male	male	1273
-	male	female	**707**
-	female	male	**221**
-	female	female	63

Table 2: Overview: number of positive (+) and negative (-) relations between the gender referring nouns.

5.1 1st Experiment: Source Imbalance

From the output of the sentiment inference system for these texts, 132 verb types display cases of an animate source *and* target. Only 20 verbs pass the strict threshold (≥ 27), and of these, 10 have a gender-specific imbalance. Table 2 shows the overall statistics. We can see that negative relations from a male-denoting noun (as source) to a female-denoting noun (as target) occur about 3 times as often as the other way around (in bold).

If we observe the most frequent verbs of these two bidirectional cases, it turns out that they are gender-specific. Among verbs whose sources are female-denoting nouns, the most frequent are (in ascending order) *coerce, deceive, threaten, accuse*; for male-denoting noun sources: *attack, kill, rape*.

Table 3 shows the list of 10 verbs with gender-specific source-role imbalance. For 7 of these verbs, male-denoting nouns take on the source role significantly more often than expected (the error risk α is 5%). A letter f (m) in a column $\leq$ means that the probability of #f (#m) female (male) sources for the verb is less than or equal to α.

In order to quantify the noise in our empirical analysis, we manually inspected all cases from Table 3. We looked for gender classification and sentiment relation errors. The last column (#e) in Table

[6] We did not take plural nouns into account since German plural male nouns for a long time have been regarded as being generic, denoting all genders. The gender reference of such a noun, thus, cannot be reliably fixed.

verb	$\leq$	$\geq$	#f+m	#f	#m	#e
attack	f	m	62	5	57	7
harass	m	f	76	31	45	1
fire	f	m	157	17	140	5
shot dead	f	m	194	25	169	1
critizice	f	m	33	1	32	3
kill (töten)	f	m	302	23	279	20
kill	f	-	62	6	56	1
rape	f	m	46	2	44	3
indict	-	f	30	9	21	0
assault	f	m	60	5	55	6

Table 3: Gender specific source role imbalance (f=female, m=male, $\leq$ means $\leq \alpha$, $\geq$ means $\geq 1 - \alpha$, e=prediction error

3 shows the error counts. E.g. *attack* was associated with 5 cases of animals in the source role and 2 cases of generic male plural nouns, which can also be used as a feminine singular noun (*Unbekannte*, Eng. unknown females). A manual analysis revealed an error rate of 4.6% (47 out of 1022).

5.2 2nd Experiment: Target Imbalance

As stated, the low frequencies shown in Table 2 are partly due to the high number of passive sentences, in which typically only a target can be found. However, we can also perform statistical tests with targets only, which would help us determine whether men and women are significantly less or more often targets than their respective distributions suggest. We found 793,246 instantiations of 233 verbs in passive voice, 66 for which we found gender-specific patterns. For instance, men are more often target of torture (line *foltern* in the appendix), *verwunden* (Eng. *injury*), *verdächtigen* (Eng. *suspect*), and *anklagen* (Eng. *accuse*) than women, who more often are targets of *vergewaltigen* (Eng. *rape*), *zwingen* (Eng. *coerce*), *benachteiligen* (Eng. *discriminate*), and *erniedrigen* (Eng. *humiliate*).

5.3 3rd Experiment: Inanimate Targets

We also tried to identify the inanimate objects (targets) toward which men and women hold a favorable or opposing view (e.g. *lies* in *She detests lies*). At the token level, we have: 3,180 +f, 1,477 -f, 22,689 +m and 9,935 -m (e.g. 9,935 negative attitudes of male towards something). At the type level: 1,857 +f, 1,030 -f, 7,258 +m, 4,564 -m. Still, the ratio of men:women is imbalanced: there are far more male- than female-denoting sources. Table 4 shows the word-level intersection percentage of the target topics that we found. The intersection is not high. A close inspection might reveal interesting differences; we leave this for future work.

	f	m	∩	%
+	1857	7258	944	10.3
-	1030	4564	500	8.9

Table 4: Men and women: likes (+) and dislikes (-) (∩ =intersection)

5.4 4th Experiment: Polar Targets

One final experiment again deals with inanimate targets, but this time we look how often men and women are *in favor* of something positive or negative, and correspondingly for the *against* relation. For this task, we use our polarity lexicon Clematide and Klenner (2010)[7], albeit without NP sentiment composition; only words are used. Table 5 shows the results. For instance, there are 1,242 cases in which men are against something positive (− →pos), e.g. decriminalization or democracy. In this paper, we have discussed the methods to generate these candidates, future work is devoted to a fine-grained qualitative analysis.

gender	+ →pos	+ →neg	- →pos	- →neg
female	149	144	178	35
male	944	896	1242	214

Table 5: In favour of + and -, against + and -, gender-specific, where pos/neg is a positive/negative word

6 Conclusion and Outlook

We introduced gender-tailored semantic role profiling on the basis of grammatical gender detection and sentiment relation extraction. Our model combines the first classifier for the detection of the grammatical gender of German nouns with an existing rule-based sentiment relation extractor. In a case study, we were able to carve out the different semantic role profiles of male and female denoting expressions in news texts from 2004 to 2014. In more recent work, we have compared the analysis of the data from 2004 to 2014 to results from the same newspapers from 2015 to present-day (Klenner, 2023), in order to see whether semantic profiles have changed or not.

[7] See under: “PolArt”-Lexicon from https://sites.google.com/site/iggsahome/downloads.

7 Discussion of Limitations

Our method detects gender imbalance by using an existing rule-based system and a new grammatical gender classifier. Neither performs perfectly, and we cannot claim that our sampling methods produce representative data drawn from the whole population. Rather, we work with a subset that can be identified by our tools. Generalizing from the subset to the population is not our intention; rather, our approach is a first step toward gender-tailored sentiment analysis. Finally, we do not claim to find biases in the data, but instead speak of imbalance and propose that a qualitative analysis of the results is needed.

8 Appendix: Table of Target Imbalance

verb	#	#f	#m	tendency
anklagen	753	89	664	≥ m ≤ f
anzeigen	324	42	282	≥ m ≤ f
bedrängen	96	33	63	≤ m ≥ f
belästigen	83	38	45	≤ m ≥ f
benachteiligen	65	21	44	≤ m ≥ f
beschuldigen	492	60	432	≥ m ≤ f
beschützen	33	10	23	≥ f
bestehlen	49	17	32	≤ m ≥ f
bestrafen	1093	153	940	≥ m ≤ f
betrügen	136	40	96	≤ m ≥ f
demütigen	57	16	41	≤ m ≥ f
deportieren	76	24	52	≤ m ≥ f
diffamieren	29	2	27	≥ m
diskriminieren	65	24	41	≤ m ≥ f
drohen	611	136	475	≤ m ≥ f
einschüchtern	28	9	19	≤ m ≥ f
foltern	224	30	194	≥ m ≤ f
kritisieren	378	45	333	≥ m ≤ f
misshandeln	108	31	77	≤ m ≥ f
nötigen	84	25	59	≤ m ≥ f
töten	2385	383	2002	≥ m ≤ f
umbringen	329	71	258	≤ m ≥ f
unterdrücken	41	17	24	≤ m ≥ f
verdächtigen	580	70	510	≥ m ≤ f
vergewaltigen	286	213	73	≤ m ≥ f
verwunden	100	10	90	≥ m ≤ f
vorwerfen	1154	142	1012	≥ m ≤ f
widerlegen	55	17	38	≤ m ≥ f
zwingen	985	232	753	≤ m ≥ f
überfallen	174	52	122	≤ m ≥ f

Table 6: Statistically significant target role imbalance: $\leq m \geq f$ means: men are significantly fewer target (then their distribution in the data suggests), women significantly more often. Other case accordingly.

Acknowledgements

Our work was supported by the Swiss National Foundation (SNF) under the project number 105215_179302 from 2018 to 2022.

References

Stephanie Campa, Maggie Davis, and Daniela Gonzalez. 2019. Deep and machine learning approaches to analyzing gender representations in journalism. Online.

Simon Clematide and Manfred Klenner. 2010. Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 7–13.

Manfred Klenner. 2023. Sentiment inference for gender profiling. In *Proceedings of the 4th Conference on Language, Data and Knowledge*. in press.

Manfred Klenner and Anne Göhring. 2022. Animacy denoting german nouns: Annotation and classification. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1360–1364, Marseille, France. European Language Resources Association (ELRA).

Manfred Klenner, Don Tuggener, and Simon Clematide. 2017. Stance detection in Facebook posts of a German right-wing party. In *LSDSem 2017/LSD-Sem Linking Models of Lexical, Sentential and Discourse-level Semantics*.

Arjun Mukherjee and Bing Liu. 2010. Improving gender classification of blog authors. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 207–217, Cambridge, MA. Association for Computational Linguistics.

Karolina Stanczak and Isabelle Augenstein. 2021. A survey on gender bias in natural language processing. *CoRR*, abs/2112.14168.

Karolina Stanczak, Sagnik Ray Choudhury, Tiago Pimentel, Ryan Cotterell, and Isabelle Augenstein. 2021. Quantifying gender bias towards politicians in cross-lingual language models. *CoRR*, abs/2104.07505.

Jiao Sun and Nanyun Peng. 2021. Men are elected, women are married: Events gender bias on Wikipedia. In *Proceedings of the 59th Annual Meeting of the ACL and the 11th International Joint Conference on Natural Language Processing*, pages 350–360, Online. Association for Computational Linguistics.

Implicit causality in GPT-2: A case study

Minh Hien Huynh and **Tomas O. Lentz** and **Emiel van Miltenburg**
Department of Communication and Cognition
Tilburg University
hienhuynh.tdn@gmail.com
{T.O.Lentz, C.W.J.vanMiltenburg}@tilburguniversity.edu

Abstract

This case study investigates the extent to which a language model (GPT-2) is able to capture native speakers' intuitions about *implicit causality* in a sentence completion task. Study 1 reproduces earlier results (showing that the model's surprisal values correlate with the implicit causality bias of the verb; Davis and van Schijndel 2021), and then examine the effects of gender and verb frequency on model performance. Study 2 examines the reasoning ability of GPT-2: Is the model able to produce more sensible motivations for why the subject VERBed the object if the verbs have stronger causality biases? For this study we took care to avoid human raters being biased by obscenities and disfluencies generated by the model.

1 Introduction

This paper is a case study, highlighting different ways to analyse the linguistic abilities of a language model, with respect to an established linguistic phenomenon, namely Implicit causality (IC) bias (Hartshorne, 2014). Speakers associate either the subject or the object of a verb with the cause of the state or event described by that verb. For example, the verb *frighten* is a *subject-biased* verb because native speakers of English tend to see the subject as the cause of the frightening event. Thus, given a main clause like in (1a), participants in a sentence completion task would tend to provide a reason referring to the subject, as in (1b).

(1) a. [MAIN CLAUSE John scared Mary because . . .]

b. [REASON he put on a Halloween costume.]

Earlier work by Upadhye et al. (2020) and Davis and van Schijndel (2020, 2021) investigated the extent to which language models are able to capture native speakers' IC biases. This paper aims to reproduce some of their earlier results, using GPT-2 (Radford et al., 2019) as an example. Using the same sentence completion task as Davis and van Schijndel (2020), we further investigate how bias, subject gender, and verb frequency influence the behavior of this model (§3). Next, we will more thoroughly assess the quality of the completions generated by GPT-2 (§4), asking: Does the model's performance hold up to further scrutiny?

Why GPT-2? Although it is neither the most recent, nor the best performing open-source language model around (see Black et al. 2022 for alternatives), GPT-2 is still a very popular choice for many researchers and practitioners. (See https://huggingface.co/gpt2 for statistics.) This popularity is at least partly due to its size, as the model can be run and fine-tuned on consumer hardware. The model's popularity means that studying its capabilities and limitations may be more impactful (at least in the short term) than studying the capabilities and limitations of larger but less accessible systems. For us, GPT-2 offers the right balance of complexity and efficiency; as shown by Upadhye et al. (2020), its outputs are good enough to have a meaningful discussion about the assessment of language model performance, without requiring a large (and expensive) computational infrastructure.

Contributions Next to the value of reproducing earlier work, and providing further details on the IC-related behaviour of GPT-2, the main innovation of this paper is the controlled assessment of output quality. We took great care to separate issues with the fluency and offensiveness of the output from the content of the generated continuations. (See Section 6 for the limitations of this study.) This not only makes the task less harmful to our participants, but it also increases their focus on our construct of interest: The reasoning abilities of language models. Our code and data from this paper is available at https://github.com/hienhuynhtdn/GPT2andImplicitCausality/.

Proceedings of the 15th International Conference on Computational Semantics pages 67–77
June 21–23, 2023. ©2023 Association for Computational Linguistics

2 Data

We present two studies. Study 1 investigates next-word surprisal values, and Study 2 looks at continuations that are generated based on a prompt. For both our studies, we provide the model with input sentences of the following form:

(2) SUBJECT VERB-*ed* OBJECT *because* ....

The verbs are derived from a list of 246 IC verbs compiled by Ferstl et al. (2011), who also provide a *bias score* for each of these verbs. This score is derived from a human experiment, where participants were asked to complete sentences like (1a). The human bias scores range from -100 (i.e., all valid continuations produced by respondents in Ferstl et al.'s experiment uniquely referred to objects of the preceding clauses) to 100 (i.e., all valid continuations referred to subjects of the preceding clauses).

The subjects and objects are provided by Davis and van Schijndel (2020), who produced a list of 14 noun pairs that are grammatically male and female (e.g. *man, woman* or *brother, sister*). A combination of the nouns and verbs, our set of stimuli consists of 6888 examples (246 verbs × 14 pairs of gender-mismatched nouns × 2 subject genders).

3 Study 1: IC-bias and pronoun use

First, we investigate whether verbs in GPT-2 possess the same subject/object bias as in the human experiment described above.

Set-up To test the hypothesis, we use the approach from Davis and van Schijndel (2021) and checked for each prompt whether the model assigned a lower surprisal to a male or female pronoun (i.e. *he* or *she*).[1] If GPT-2 captures the IC bias, then subject-biased verbs with a female subject should lead the model to produce lower surprisal values for *she*. Since each noun pair is used in both orders (either a male or a female noun in subject position), we have a perfectly balanced dataset.

Results Figure 1 shows the results split by subject gender and bias scores from Ferstl et al. (2011). GPT-2 generally picks up on the subject or object bias of the verb. The gender produces more subject-based explanations if the verb's IC bias is more subject-biased. There is only one exception: Performance for sentences with both subject-biased verbs and female subjects is at chance level for all bias scores.

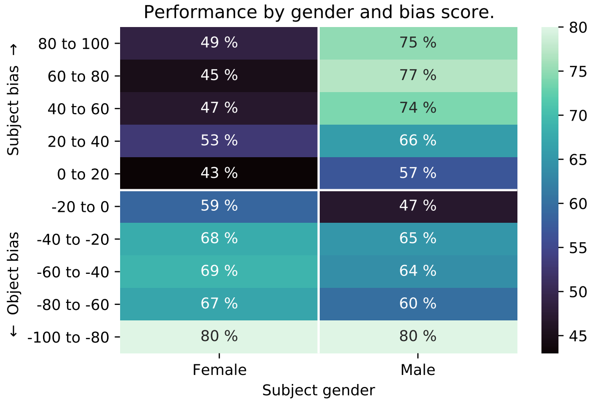


Figure 1: Heatmap table showing the percentage of outputs matching the subject or object bias of the verb, with scores separated by subject gender.

IC Bias and Gender. We used the lme4 (Bates et al., 2015) and lmerTest package in R (Kuznetsova et al., 2017) to carry out a mixed effects regression analysis of the relationship between the bias scores and GPT-2's subject-preference scores, corresponding to the difference between the surprisal for the object-congruent pronoun and the surprisal of the subject-congruent pronoun. Details about our statistical analyses are provided in Appendix B. There were significant fixed effects of human bias score ($b = 0.003, \text{SE} = 0.0001, p < .001$) and of the interaction between human bias score and subject gender ($b = 0.0014, \text{SE} = 0.0002, p < .001$). This confirms our observations from Figure 1. The effect of subject gender was found to be not significant ($b = -0.0103, \text{SE} = 0.012, p = .374$).[2]

Verb frequency and model performance. We then investigated whether verb frequency is positively correlated with the language model performance. For more frequent verbs, we hypothesize that the model more closely matches the human subject/object bias. To test this hypothesis, we regressed the squared errors from the previous mixed effects model to the log-transformed word frequencies of verbs used in the stimuli. As a proxy for verb frequency in the training corpus, we used the

[1] Next-word surprisal is estimated for a target upcoming word by taking the inverse log of this word's probability: $surprisal(w_t) = -logP(w_t|w_1 \dots w_{t-1})$

[2] In addition, the coefficient of determination of the model was calculated based on the method developed by Nakagawa and Schielzeth (2013) using the MuMIn library in R (Bartoń, 2022). Approximately 14% of the variance in the GPT-2's subject-preference score could be explained by the fixed effects alone (marginal R2 = .142) while 23% of the variance in the subject-preference score could be explained by both fixed and random effects (conditional $R^2 = .230$).

SUBTLEX-US word frequencies from Brysbaert and New (2009). We found a significant fixed effect of log-transformed word frequencies of the verbs in our materials ($b = -0.049, \text{SE} = 0.012, p < .01$). This supports our hypothesis: More data leads to a better approximation of human IC bias, in terms of the preference for either male or female pronouns.

4 Study 2: Assessing continuations

We now turn to continuations generated by GPT-2 based on the prompts described in Section 2. Our results above suggest that GPT-2 can generate continuations according to the causality bias pattern, given enough data. Based on this result, we now hypothesize that such continuations are better when the IC pattern is clearly present in the data. The stronger the IC bias is, the clearer the pattern of continuations in the training set. Of course, this task is much more difficult than simply generating the right pronouns. We now want to see whether the continuations actually *make sense* in the eyes of human judges. To this end, we collected human ratings for a carefully controlled subset of the continuations generated by GPT-2 (see Appendix C for details). IRB approval was obtained prior to this study. (Ethical considerations in Appendix A.)

Participants We used the Prolific participant pool to recruit 75 participants for the sentence rating task. Our items were spread across 25 different lists, and each participant was only allowed to provide ratings for one list. We restricted potential participants to native speakers of English, either from the UK or from the USA. After assessing response quality, we recruited five additional participants, to obtain three reliable judgments per item.

Task and target construct Each continuation was assessed by three different participants, who judged whether the continuations were reasonable, given the prompt. We set up our experiment as a rating task, where each participant indicated for a list of 40 items whether they agreed with the statement that the continuation was 'reasonable.' Participants could indicate their agreement on a five-point Likert scale, ranging from 'Strongly Agree' to 'Strongly Disagree.' With the addition of some examples in our task description (see Appendix G), we targeted our participants' intuitions for what makes a good reason to do something. As we will discuss below, we aimed to avoid any influence of the form of the output as much as possible.[3]

Prompt selection Due to financial limitations, we were not able to obtain ratings for all 6888 continuations. Following the recommendations from van Miltenburg et al. (2021), we used a stratified sampling approach. We selected the 5 most frequent noun pairs, and the 10 most frequent verbs for each of the 10 different bias levels (as illustrated in Figure 1). This selection gives us a sense of the upper bound performance with respect to continuation quality. Frequency was again determined using the SUBTLEX-US data (Brysbaert and New, 2009). Prompts were constructed in the same way as before (see Ex. 2), with each noun pair being presented in both orders. This yields 10 verbs × 10 bias levels × 5 nouns × 2 orders = 1000 prompts.

Data preparation We used GPT-2 to generate continuations for each prompt. As shown earlier, the problem with these continuations is that they may be offensive or contain disfluencies (most notably repetition, see Fu et al. 2021). This creates two problems: (i) Offensive output may cause psychological harm for our participants, and (ii) offensiveness and disfluencies may lower the reliability of the ratings, if participants consistently provide lower scores for offensive/disfluent outputs (even though they may be consistent with the prompt). To prevent harm, and to avoid noise in the ratings, we took the following approach:

1. If the output is offensive, select a different noun pair from the 9 remaining pairs to generate a non-offensive alternative continuation.
2. If the output contains repetition, manually remove repeated elements from the sentence, so that the core content of the continuation remains largely unchanged.

Reliability To assess annotator reliability, we used a leave-one-out approach to correlate each participant's scores with the mean scores of the two other participants who rated the same responses. Initial correlations ranged between 0.17 and 0.84. Five annotators scored below our cutoff of 0.4, and thus we recruited five more participants. After recomputing the reliability scores, we kept only the

[3] Given the terminological confusion in the field of Natural Language Generation (Howcroft et al., 2020), Belz et al. (2020) developed a categorization system for evaluation criteria in NLG. In terms of their taxonomy, our informal notion of 'reasonable continuation' clearly focuses on the content of the output, but the frame of reference is harder to define. If we look at the completed sentence as a whole, the sentence is evaluated in its own right, and so it is a question of *Coherence*.

Rating	1	2	3	4	5
Raw counts	689	571	333	782	625
% of ratings	23	19	11	26	21
% avg rating >= rating	100	79	55	32	7

Table 1: Distribution of the ratings. Ratings correspond to a Likert scale, where 1=Strongly disagree, 2=Somewhat disagree, 3=Neither agree nor disagree, 4=Somewhat agree, 5=Strongly agree.

three highest-scoring participants per task. This way, we obtained a mean score of 0.64, with a standard deviation of 0.11. This is a strong correlation, considering the subjective nature of the task.

Continuation quality Table 1 shows the ratings in three different ways. The top row shows the (bimodal) raw score distribution: Ratings tend to be either negative or positive, with relatively few ratings in the middle of the scale. The second row provides the same values as percentages, as a guide for the third row. The third row shows the percentage of continuations for which the average rating is greater than or equal to a given rating. For example: Only 32% of all the continuations have an average rating greater than or equal to 4.

A total of 72 sentences was only rated at the lowest level. Table 2 (next page) provides examples of low-quality categorisations, with a rough categorisation. The three error categories are:

1. Non-sensical: Continuations that do not make sense as a reason for anything.
2. Invalid reason: Continuations that provide a reason, but the reason is not applicable.
3. Subject-object reversal: Continuations that would have made sense if subject and object were reversed.

These categories are not mutually exclusive (and fairly subjective) because it is hard to pin down what makes for a good/bad continuation. Nevertheless, it is clear that the example continuations in Table 2 provide poor reasons indeed. Future research could carry out a more systematic error analysis (along the lines of van Miltenburg et al. 2021), and present the distribution of the different kinds of erroneous continuations.

There were 69 continuations that always received the highest rating seem to conform to the human IC bias. One random example is 'The woman thanked the man because he was a good man.'; the verb 'to thank' is strongly object-biased (raw subject bias score -92).

Continuation diversity Besides continuation quality, we also observe low continuation diversity. Table 3 shows the five most frequent continuations for our prompts, split by subject gender. It is clear from the table that the continuations generated by GPT-2 are very repetitive, and tend to be generic without any specific details. So there are no examples like (3) in the generated continuations:

(3) The woman thanked the man because he gave her a nice book for her thirty-seventh birthday.

Explaining model performance We again used a mixed effects model to analyze our results. Our aim is to explain GPT-2's performance (i.e., how reasonable the continuation is) in terms of verb frequency and absolute IC bias (which only looks at strength of the bias).[4] Full details about the model and model fitting are provided in Appendix D.

Subject gender and its interactions were considered, but did not significantly improve model fit, and were dropped. Absolute IC bias has a small but positive effect on the rating ($b = 0.003, \text{SE} = 0.001, p = 0.015$). The effect of frequency is negative ($b = -0.121, \text{SE} = 0.038, p = 0.002$). There is also a negative interaction of bias and frequency ($b = -0.006, \text{SE} = 0.001, p < .001$), indicating that the positive effect of bias diminishes for higher frequency verbs. Thus, while we concluded in Study 1 that the accuracy of GPT-2 with respect to subject/object bias increases as the verb becomes more frequent, we now find that higher frequency does not give us more reasonable continuations. The (small) positive effect of absolute IC bias on the ratings can be explained by the intuition that the IC bias pattern is likely more clearly present in the training data for verbs that are more strongly biased. This makes it easier to pick up the pattern.

5 Discussion

The difference between Study 1 and Study 2 indicates a qualitative difference between generating pronouns and providing explanations: The latter requires higher-level reasoning which may not be present in language models like GPT-2 (Bender and Koller, 2020). Though the fact that GPT-2

[4] IC bias and log-transformed frequency values were determined as for Study 1. IC bias scores were made absolute; the mean absolute IC bias is 49.3 (SD 27.5, range [2, 92]). The mean (untransformed) frequency per million is 45.4 (SD 90.7, range [0.02, 502.27]). The correlation between IC bias and frequency was not significant and low (Pearson's $r = -0.02, p = .35$).

Error category	Examples
Non-sensical	The girl hit the boy because he was too young to be a boy. The girl chased the boy because he was wearing a black hoodie and a black hoodie with a black hoodie on.
Invalid reason	The mother approached the father because she was afraid of him. The mother scared the father because he was a little too big for her.
Subject-object reversal	The woman affected the man because she was afraid of him. The woman surprised the man because he was wearing a black suit and a black tie.

Table 2: Rough categorisation of examples of low-quality continuations

Male subject		**Female subject**	
Continuations	Frequency	Continuations	Frequency
he was afraid of her	521	she was afraid of him.	476
she was a woman	102	he was a good man and he was a good man	166
he was a good man and he was a good man	68	he was a good man	124
she was a good girl	64	he was a good boy	92
she was a woman, and he was a man	60	she was a woman	57

Table 3: Most frequent continuations for the prompts in Study 2, split by subject.

follows the implicit causality bias in pronoun selection is at least compatible with knowledge of causality, the quality of the continuations suggests the pronoun selection is based on superficial heuristics rather than a deep understanding of language (also discussed as *fast* versus *slow*; see Choudhury et al. 2022; Kahneman 2011). Although existing suites for LM evaluation (e.g. Ettinger 2020) are useful, slower forms of assessment (such as human evaluation) are helpful to tease out this difference.

6 Limitations

The main limitation of our paper is that we focused on only one language model (GPT-2), and only in one language (English). So while our findings provide insights into the capacities of the English GPT-2, they cannot be generalised to other language models or other languages (which is also illustrated by Davis and van Schijndel (2021)). Our main contribution is methodological, namely exploring how to assess the linguistic capacities of language models. Because our approach treats GPT-2 (mostly) as a black box, our analysis can easily be applied to other models as well.

For Study 2, a negative effect of verb frequency on the quality of generated continuations was found. As the materials used for the rating study were not gathered to cover the full range of frequencies, this pattern should not be generalized and may reflect a hidden effect. The negative interaction with IC bias suggests the same. The model fit shows an imperfect fit, but without one clear deviation from linearity. Hence, our findings may well be explained better with more appropriate independent variables.

7 Conclusion

This paper showed two different ways to assess the linguistic capacity of a language model (GPT-2), using implicit causality as a case study. The techniques used above can be applied in a black box setting, without the need to look at the internals of the model. We hope that this paper is useful for others aiming to assess the ability of other language models to capture different linguistic phenomena as well. Our findings also showed that automatic assessment methods may not be enough to determine whether semantic phenomena like implicit causality are learned by a language model. Human evaluation remains a necessary complement to automatic evaluation (van der Lee et al., 2021). Our paper shows one way to do this without participants being influenced by factors like grammaticality and offensiveness of the output.

Acknowledgments

We thank all participants of our crowdsourcing experiment. This study was approved by the Research Ethics and Data Management committee at the Tilburg School for Humanities and Digital Sciences (TSHD) (code: 2019.40ab). We thank the committee for their feedback. The second author acknowledges the support of the Dutch Research Council (NWO) as part of the InDeep consortium (NWA.1292.19.399).

References

Kamil Bartoń. 2022. *MuMIn: Multi-Model Inference*. R package version 1.46.0.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*, 41(4):977–990.

Sagnik Ray Choudhury, Anna Rogers, and Isabelle Augenstein. 2022. Machine reading, fast and slow: When do models "understand" language?

Forrest Davis and Marten van Schijndel. 2020. Discourse structure interacts with reference but not syntax in neural language models. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 396–407, Online. Association for Computational Linguistics.

Forrest Davis and Marten van Schijndel. 2021. Uncovering constraint-based behavior in neural models via targeted fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1159–1171, Online. Association for Computational Linguistics.

Leon Derczynski, Hannah Rose Kirk, Abeba Birhane, and Bertie Vidgen. 2022. Handling and Presenting Harmful Text. *arXiv e-prints*, page arXiv:2204.14256.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Evelyn C Ferstl, Alan Garnham, and Christina Manouilidou. 2011. Implicit causality bias in english: A corpus of 300 verbs. *Behavior Research Methods*, 43(1):124–135.

Karën Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Last words: Amazon Mechanical Turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2021. A theoretical analysis of the repetition problem in text generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12848–12856.

Florian Hartig. 2022. *DHARMa: Residual Diagnostics for Hierarchical (Multi-Level / Mixed) Regression Models*. R package version 0.4.5.

Joshua K. Hartshorne. 2014. What is implicit causality? *Language, Cognition and Neuroscience*, 29(7):804–824.

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701, Minneapolis, Minnesota. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York.

Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. 2017. lmertest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13):1–26.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Emiel van Miltenburg, Desmond Elliott, and Piek Vossen. 2018. Measuring the diversity of automatic image descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1730–1741, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shinichi Nakagawa and Holger Schielzeth. 2013. A general and simple method for obtaining r2 from generalized linear mixed-effects models. *Methods in ecology and evolution*, 4(2):133–142.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Shiva Upadhye, Leon Bergen, and Andrew Kehler. 2020. Predicting reference: What do language models learn about discourse models? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 977–982, Online. Association for Computational Linguistics.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William S. Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *CoRR*, abs/2112.04359.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

A Ethical considerations

Because our study deals with human subjects, we first obtained ethical approval from our IRB. We describe our considerations below.

A.1 Information letter and informed consent

Our IRB mandates the use of a separate information letter (Appendix E) and informed consent form (Appendix F). With the information letter, we give a general description of the study, and provide an indication of potential risks and benefits of the study. The informed consent form is provided separately to prevent information overload, and to ensure that participants know what they are agreeing to, if they decide to take part in our study.

A.2 Crowdsourcing and payment

Crowdsourcing has been criticized for its potentially exploitative nature (Fort et al., 2011). We explicitly frame our task as an *experiment* with *human participants*, rather than a *human intelligence task* with *crowdworkers*, and apply the same considerations and protections as for lab experiments. Nevertheless, it is still work, and work needs to be paid. Based on experience, we expected participants to spend roughly 15 minutes on the task, and set the compensation to £2.40, which amounts to £9.60/hour; 10 cents above the current UK minimum wage.[5] In the end, the vast majority of our participants spent less time than expected on our task (time in mm:ss format: range: 03:06–16:18, median: 06:18, mean: 06:59, SD: 02:55.).

All participants were compensated for their time, including those providing low-quality responses.

A.3 Offensive material

We wanted to avoid confronting our participants with profanity or otherwise potentially harmful language. We manually identified potentially offensive continuations generated by the model, and replaced harmful outputs with alternative continuations generated for different prompts. We considered continuations potentially harmful if they contained profanity or made reference to religion, violence, or sexual acts. All originally generated sentences and their replacements can be found in the GitHub repository associated with our paper.

A.4 Language model-related harms

Language models are associated with several different harms (Bender et al., 2021; Weidinger et al., 2021), but these harms also depend on the task at hand. For example, since we used a pretrained model, our study did not incur any additional training costs. And as described above, since no one other than the authors were directly exposed to the model's output, we could prevent our participants from seeing harmful or toxic content. Thus we are mostly left with inference costs, which are relatively low, since GPT-2 can run on a personal computer.

A.5 Intended use of this work

This study serves two purposes: (i) To explore the ability of a language model (GPT-2) to capture native speakers' intuitions about implicit causality, and (ii) to develop an evaluation methodology that isolates coherence of the responses from other factors like offensiveness and (un)wellformedness. We do not wish to make any claims about the cognitive capacity of language models in general, nor do we want to claim that GPT-2 can somehow reason about the world. We just want to see whether the model can generate outputs that follow earlier observations about implicit causality, and that are internally consistent. Follow-up studies in the spirit of this work are encouraged.

A.6 Licensing

All resources used for this study were developed for research purposes, but not all materials have a clearly indicated license. GPT-2 is provided under the MIT license.[6] The work by Davis and van Schijndel (2020, 2021) is provided on GitHub without any license[7] and both Brysbaert and New (2009) and Ferstl et al. (2011) published their work in the *Behavior Research Methods* journal without a clear license, but with a clear intention for their work to be used for research purposes. We thus conclude that academic use of these resources is warranted.

B Study 1: Statistical analysis

As noted in Section 3, we used the lme4 (Bates et al., 2015) and lmerTest packages in R (Kuznetsova et al., 2017) to carry out a mixed

[5] See https://www.gov.uk/government/publications/the-national-minimum-wage-in-2022

[6] https://huggingface.co/gpt2

[7] https://github.com/forrestdavis/ImplicitCausality

effects regression analysis of the relationship between the model's subject-preference score and human bias score of every IC verb, reducing the complexity by removing terms that do not significantly improve fit.

As fixed effects, human bias score, subject gender of the stimulus sentence-fragments (with two levels, namely male or female) and their interaction were included. Moreover, we included item, which indicated the pairs of antonymous nouns used in the stimulus sentences as subjects and objects, as a random effect, and added by-item random slopes for the effects of human bias score, subject gender and their interaction. However, the fitting of the full model including all fixed and random effects failed to converge. Therefore, the by-item random slopes for the effects of human bias, subject gender and their interaction were removed. Our final model is the following in R notation:

(4) subject-preference score ~ human IC bias score * subject gender + (1|item)

We regressed the squared errors obtained from the linear mixed effects regression model we performed in the previous step on the log-transformed word frequencies of IC verbs used in the stimuli. As random effects, item was also entered into the model, and by-item random slope for the effect of log- transformed frequency was added.

C Study 2: Continuations

While Study 1 looked into the surprisal values for the generation of male/female pronouns, this study looks at continuations themselves.

C.1 Consistency check

We first checked whether the IC bias pattern continues to be present in the full continuations. In all but 176 cases, the model generated a gendered pronoun. These cases were coded manually, to determine whether the continuation referred to the subject or the object. Figure 2 shows the proportion of continuations referring to either the subject or the object of the prompt (Y-axis), split by the bias score (X-axis). We see the same trends as in the first study: Overall we see that references to the subject increase as the bias score increases. This trend also holds when we split the prompts by subject gender, but for the female subjects the proportion of references to the subject never exceeds 50%.

C.2 Patterns

We then inspected the frequencies of the continuations. We observe that more than 20% of the outputs is repeated more than 100 times. This lack of diversity is a common issue in (neural) Natural Language Generation (e.g. van Miltenburg et al. 2018; Hashimoto et al. 2019; Zhang et al. 2021). It is not necessarily a problem at the individual level (the continuation may be a bit generic but still appropriate for the given context), but at the corpus level these 'one size fits all' continuations are shortcuts that prevent more varied outputs.[8]

C.3 Offensive output

Looking over the generated outputs, there were several occasions where the model generated offensive continuations, including instances of sexism, racism, and misogyny. On top of this, the model outputs also contained sexually explicit words, and some continuations described acts of violence. Following the recommendations from Derczynski et al. (2022), we do not provide any examples in this paper. As discussed in Section 4, we also removed (potentially) offensive continuations from our human rating experiment. For transparency reasons, we do provide those sentences in our GitHub repository.

D Study 2: Statistical analysis

A full linear model was built with the factors subject gender, absolute IC bias (centered), log-transformed word frequency (z-scored) and their interactions as fixed effects, and random effects for participant (intercept, and slopes for the three main effects), sentence (intercept), and subject/object pair (intercept). The model was then subjected to the step function of lmerTest, which removes insignificant components. Though the full model and some of the first reductions of it did not converge, this procedure is still appropriate (a model that does not converge should not be used, and as more data is not available, the model should be simplified). The reduced final model, that did converge, is, in R syntax:

[8] We might also question whether it is possible at all to properly assess the cognitive capacity of a model that keeps using such shortcuts (which may be seen as cheating). As an alternative, we can imagine a two-stage process where researchers first generate an unrestricted set of continuations, and then force the model to avoid common continuations.

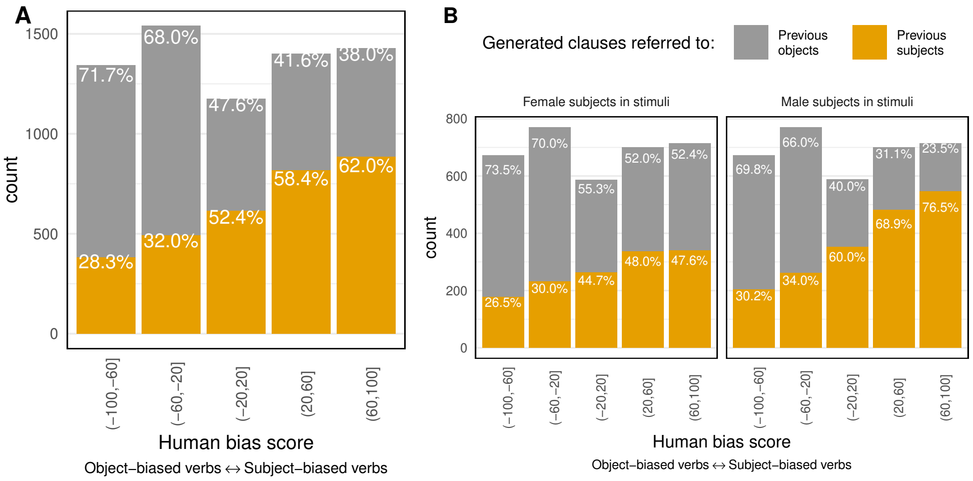


Figure 2: Percentage of continuations referring to subjects or objects by bias scores of verbs. Panel A shows overall proportion, Panel B shows proportion split by subject gender

(5) rating ~ abs(IC bias_abs_c) * frequency + (1 | sentence_ID) + (1|participant) + (1|item)

The DHARMa package (Hartig, 2022) was used to assess model fit. Though the residuals are not normally distributed, the deviations show no clear pattern. To avoid spurious conclusions, we corroborated the significant estimates by checking if their 95% confidence intervals included 0, which they did not. Hence, treating ratings as a continuous numeric variable was not problematic.

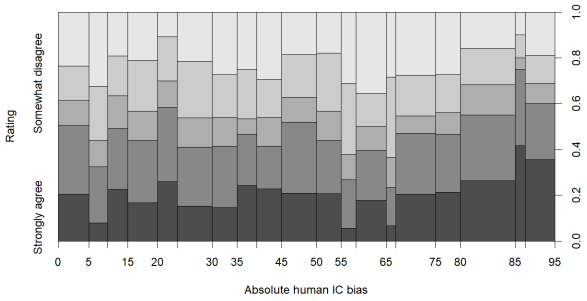


Figure 3: Ratings by absolute value of human IC bias.

E Information letter

Title: Assessing computer-generated texts

Introduction

We invite you to take part in our study to assess computer-generated texts. This study is part of a larger project to see how good or bad computers are at producing or understanding human language,such as English. In this study you will be asked to rate the quality of computer-generated texts/sentences. We will use this information to see what the computer is good at, and to see where it can still be improved.

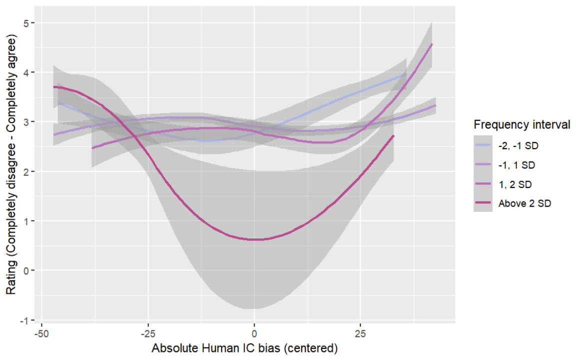


Figure 4: Ratings by absolute value of human IC bias, separated for four different levels of verb frequency.

What do I have to do?

As mentioned above, you will be asked to rate the quality of computer-generated texts. In this study, you will be asked to read 40 short sentences, and to provide your judgment. We are interested in your intuition as a native speaker of English, so you don't need to think too long about it.

Expected duration

We expect this study to take about fifteen minutes of your time. Other than this, we do not foresee any risks associated with this study. On the positive side, your participation will improve our understanding of the language capacity of modern computer models.

Ethics and rights
This study was approved by the Research Ethics and Data Management Committee (REDC) at Tilburg University.

Your participation is completely voluntary. Your consent to participate generally applies for the duration of this study. However, you have the right to decline to participate and withdraw from the research once participation has begun, without any negative consequences, and without providing any explanation.

Your participation is completely anonymous. We will not store any identifying information, so your answers cannot be traced back to you. We only see the demographic information that Prolific provides. Do let us know if you would like to have a copy of your responses, and we will try to obtain them based on the Prolific ID.

Use of data
Your responses will be used for the current study, and possible follow-up studies in the future. This means that the data will be presented in research articles, that are publicly available. For full transparency, we will also publicly share the anonymised individual responses. As such, they will be stored indefinitely.

Contact
If you have any questions, or if would like to learn more about this study, please contact C.W.J.vanMiltenburg@tilburguniversity.edu for any of your questions.

Ethics approval
If you have any remarks or complaints regarding this research, you may also contact the "Research Ethics and Data Management Committee" of Tilburg School of Humanities and Digital Sciences via tshd.redc@tilburguniversity.edu.

F Informed consent

By agreeing to this consent form, you confirm that you have read the study description and that you have been offered the opportunity to ask questions (via email). Remember that your participation is voluntary, and that you have the right to decline to participate and withdraw from the research once participation has begun, without any negative consequences, and without providing any explanation.

I hereby give permission to:

- Store my anonymised responses to this survey.
- Analyse the anonymised data (both manually and automatically through statistical software).
- Make the responses to this survey publicly available upon completion of the study.

Yes ⇒ continue to the survey.
No ⇒ continue to the end of the survey.

G Task instructions

Task instructions
All questions below are of have the same form. You will see the start of a sentence on the first line, and a continuation generated by a computer model on the second line. Your job is to assess the quality of the continuation on the second line.

Example of a reasonable continuation:
For the following sentence: *The clown startled the girl because*
A reasonable continuation would be: *his make-up was scary.*

- Strongly agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Strongly disagree

This is a reasonable continuation because scary make-up can cause someone to be startled. So here you would answer *Strongly agree*.

Example of a less reasonable continuation:
For the following sentence: *The clown startled the girl because*
A reasonable continuation would be: *she liked him.*

- Strongly agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Strongly disagree

This is a less reasonable continuation, because being liked by someone is generally not a reason to startle them. So here you would answer one of the disagree options.

Multi-purpose neural network for French categorial grammars

Gaëtan Margueritte[1], **Koji Mineshima**[2], **Daisuke Bekki**[3]
[1]ENSEIRB-Matmeca Engineering School, [2]Keio University, [3]Ochanomizu University
gamargueritte@gmail.com, mineshima@abelard.flet.keio.ac.jp,
bekki@is.ocha.ac.jp

Abstract

Categorial grammar (CG) is a lexicalized grammar formalism that can be used to identify and extract the semantics of natural language sentences. However, despite being used actively to solve natural language understanding tasks such as natural language inference or recognizing textual entailment, most of the tools exploiting the capacities of CG are available in a limited set of languages. This paper proposes a first step toward developing a set of tools enabling the use of CG for the French language by proposing a neural network tailored for part-of-speech and type-logical-grammar supertagging, located at the frontier between computational linguistics and artificial intelligence. Experiments show that our model can compete with state-of-the art models while retaining a simple architecture.

1 Introduction

Categorial grammar (CG) is a formalism whose foundations come from Ajdukiewicz (1935) and Bar-Hillel (1953). From there, we can find two major lines of research that were created, namely, combinatory CG (CCG) (Steedman, 2000) and type-logical grammar (TLG) (Moortgat, 1997; Morrill, 1994) which itself can be divided into two subtheories, namely, displacement calculus (Morrill et al., 2011) and multi-modal CG (Moortgat, 1997). Other theories that build upon those theories also exist, such as hybrid TLCG (Kubota and Levine, 2020) and abstract CG (de Groote, 2001).

Using these syntactic theories offers knowledge about each word passed in an input sentence. Using the appropriate resources, the great amount of information provided by a *supertag* (Bangalore and Joshi, 1999) attributed to a given word in a sentence can be parsed efficiently to solve natural language understanding tasks such as natural language inference or recognizing textual entailment. This syntax-semantic interface can then be used by machines in order to answer various kinds of challenges, such as question answering and text summarization.

The continuous development of CCG and TLG led to the progressive appearance of several annotated corpora in various languages, such as German (Hockenmaier, 2006), Italian (Bos et al., 2009), Japanese (Uematsu et al., 2013), and of course English (Hockenmaier and Steedman, 2007). However, the number of treebanks and tools is very limited for the French language. Because CG has a close affinity to lambda calculus, logic, and natural deduction proofs, we are motivated to develop the current state-of-the-art in this field for the French language.

In this work, we propose a simple supertagger for part of speech (POS) and TLG tagging by exploiting the capacities of deep bidirectional encoder representation from transformers (BERT) (Devlin et al., 2018) for unlabeled input sentences. We demonstrate that integrating into our architecture a small long short-term memory (LSTM)-based variational autoencoder (VAE) while adapting the training pipeline allows us to increase the word-wise supertagging accuracy of our model. We also show experimentally that joining the training of both POS and TLG supertagging offers slightly increased overall accuracy while reducing the accuracy of tags seen rarely during training.

2 Related works

French TLG and POS supertagging The TLGbank (Moot, 2015) is a type-logical treebank for French, developed from the French Treebank, a lexical and syntactic resource by Abeillé et al. (2003). Because both corpora have been manually verified and rectified by their respective authors, they can be considered as the gold standard for French CG. Alongside his TLGbank, Moot pre-

Proceedings of the 15th International Conference on Computational Semantics pages 78–82
June 21–23, 2023. ©2023 Association for Computational Linguistics

sented the supertagger DeepGrail,[1] which is an LSTM layer that uses ELMo (embeddings from language models) vector embeddings of the unlabeled input data. This model successfully assigns 93.2 percent of words their correct TLG formula and presents an accuracy of 99.1 percent of correct POS supertags.

Since then, state-of-the-art TLG supertagging of this treebank has been achieved by Kogkalidis and Moortgat (2022) with an accuracy of 95.92 percent. Their approach revisits traditional models by proposing a framework based on heterogeneous dynamic graph convolutions and by decomposing the structure of the supertags. By doing so, they presented novel accuracy results on supertags that were rarely seen during the training phase. This generalization effort motivated us to explore different ways to regularize our architecture without losing overall model accuracy.

CamemBERT Our approach is built around the use of CamemBERT (Martin et al., 2020), which is a fine-tuned RoBERTa model (Liu et al., 2019) for French, which itself is based on BERT (Devlin et al., 2018). This model is attractive for the French language because it uses a subword tokenization where each word is divided, so it can exploit the numerous inflections that appear in the French language. In the study reported herein, we found only a few differences between the experimental results of CamemBERT$_{BASE}$ and CamemBERT$_{LARGE}$ models. Therefore, for the sake of computing speed and efficiency, we used only CamemBERT$_{BASE}$ in our model because its architecture is three times smaller than its other version.

3 TLG and POS supertagger model

In this section, we describe the training data and procedure and present the different modules of our model.

3.1 Training data

We manually split the TLGbank with a fixed seed into train/dev/test splits at a ratio of 80:10:10 to have comparable results with the network proposed by Kogkalidis and Moortgat (2022). For each word, the corpus presents its TLG and French POS supertags, allowing us to test several versions of our network using solely the 14,521 parsed sentences of the treebank (411,520 words).

Class	Frequency	Number of words
Frequent	$n \geq 100$	43,861
Uncommon	$100 > n \geq 10$	761
Rare	$10 > n \geq 1$	139
Unseen	$n = 0$	21

Table 1: Supertag classes statistics of the TLGbank.

CGs such as TLG often suffer from a large number of possible supertags. To evaluate the regularization power of our architecture, we group the tags into four classes based on their frequency of appearance in our training split. Table 1 shows the supertag class names, frequency of tags in the train split, and number of different words whose supertag is in this class.

Because POS supertags do not share the same sparsity as TLG supertags (<30 different tags for the French MElt POS tagset), we report only the overall accuracy on this task.

3.2 Model architecture

We develop each part of the model presented in Figure 1 before presenting how the different modules were combined and evaluated. For simplicity, we call the model VAEoTL (variational autoencoder over transfer learning).

CamemBERT CamemBERT was trained originally on the masked language modeling task. Thus, we fine-tune CamemBERT$_{BASE}$ during the training phase while only removing its original head in a classical transfer-learning fashion. For each phase of training described in Section 3.3, the learning rate of CamemBERT is 10 times lower than for the rest of the model in order not to waste its pre-training. CememBERT's subword tokenization requires us to adapt the output size. Because we attribute only one supertag per word (and not per subword), we adapt the training data by attributing the supertag to the first subpart of each word and by padding the other subparts. Accuracy is thus evaluated using a simple mask removing this padding.

BiLSTM A single-layered bi-directional LSTM (BiLSTM) is used after the CamemBERT layer. It is a recurrent network that combines two LSTMs: one reading the sentence from left to right, and one reading the sentence from right to left, thus extracting for each input information coming from

[1] https://richardmoot.github.io/DeepGrail/

its neighbors on both sides.

VAE With the goal of regularizing our network in mind, we tried to add a VAE to our architecture. This module allows us to approximate the output distribution of the BiLSTM by encoding it to a latent space, before decoding it to reconstruct the aforementioned outputs. Doing so allows us to regularize the BiLSTM outputs and to increase the supertagging accuracy, specifically over rare tags. Internally, the encoder and decoder of the VAE module are both composed of BiLSTM linked by dense layers to the latent space. In our case, a latent space of size 200 was the best compromise between speed and efficiency in the final model.

However, integrating this module requires adapting the training procedure because it requires the previous layers to be pre-trained. We differentiate our procedure into three distinct phases as described later in Section 3.3.

Dense+CRF heads The final output of our neural network is tagged by a simple dense layer mapping the hidden dimensions to tagset space in order to produce probability emission for each possible supertag. However, applying a simple softmax activation function to such emissions would imply that each tag is conditionally independent of its neighbor, which is in sharp contrast to the nature of CGs.

While the softmax activation allows us to distribute the probability for each supertag to be chosen given an input word, it sometimes fails to modelize the relationship between adjacent supertags. Instead, we use a conditional-random-field layer (Lafferty et al., 2001), a discriminative model that finds the Viterbi path maximizing the probability of a sequence of possible supertags given an input sequence. This effectively considers the context around each supertag while allowing us to use a simple forward-backward algorithm to compute the negative log-likelihood between network emissions and target outputs.

Two different heads are required because we want to evaluate both the TLG and French POS supertagging tasks. We experimented on two possible applications of this model: *single-headed* or *multi-headed*. In the former, we train only a single head at once, thus dedicating the whole architecture to a single task. In the latter, we share the training of the previous layers between each task, on the hypothesis that overall accuracy should improve because only the most relevant features will be learned, thereby effectively preventing overfitting.

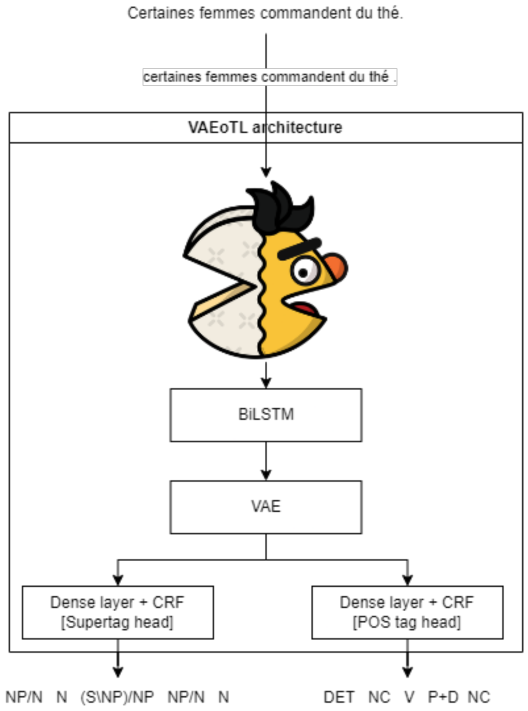

Certaines femmes commandent du thé.
certaines femmes commandent du thé .
VAEoTL architecture
BiLSTM
VAE
Dense layer + CRF [Supertag head]
Dense layer + CRF [POS tag head]
NP/N N (S\NP)/NP NP/N N
DET NC V P+D NC

Figure 1: Architecture of the network

3.3 Training procedure

For its training, the VAE module requires an adapted negative log-likelihood with regularizer and to have its previous layers sufficiently trained. Accordingly, we define three distinct phases to our training. The first phase (20 epochs) does not use the VAE module at all, because we do not wish to approximate the outputs of an untrained model. In the second phase, we remove the heads of the model and freeze the training of CamemBERT and the BiLSTM layers in order to train the VAE for 10 epochs, using the mean squared error as a reconstruction criterion added to the Kullback–Leibler divergence in order to compute the loss. In the final phase, we unfreeze all layers and fine-tune the whole model for 10 epochs.

3.4 Implementation

We implement our model using PyTorch,[2] which provides an easy-to-use-and-adapt interface to construct our model, alongside Huggingface,[3] from which we accessed the CamemBERT model.

[2] https://pytorch.org/

[3] https://huggingface.co/

Model	Overall	Frequent	Uncommon	Rare	Unseen
ELMo & LSTM (Moot, 2015)[1]	93.20	95.10	75.19	25.85	0.0
Phase 1 Single-head	95.47	95.90	81.20	41.30	0.0
Phase 1 Multi-head	95.57	96.00	83.57	28.78	0.0
Final Single-head	95.58	96.00	81.20	45.19	0.0
Final Multi-head	95.66	96.13	83.04	28.78	0.0
HDC (Kogkalidis and Moortgat, 2022)[1]	95.92	96.40	81.48	55.37	7.26

Table 2: Model performance in percent for each category of tags (average over five runs). HDC stands for heterogeneous dynamic convolutions. [1]Reported results from the cited paper.

For each phase, we use a different Adam optimizer with $\beta = (0.9, 0.999)$, no weight decay, and a learning rate of 10^{-4} fading to zero with polynomial decay. To regularize the outputs, 40 percent dropout is added during training.

4 Results

In Table 2, we present the wordwise supertagging accuracy compared to the state-of-the-art results published by Kogkalidis and Moortgat (2022) in TLG supertagging. Although our model did not surpass the state of the art, we proved its efficiency despite its simplicity.[4] The first training phase is enough to reach high accuracy, but we observe that adding a VAE module still allows us to improve our accuracy, specifically over rare tags.

We observe that sharing the training between TLG and POS supertagging allows us to improve overall accuracy while sacrificing rare-tags accuracy. This is because the model will learn the underlying correlation between both types of supertags, thus reducing the probability of picking rare TLG supertags knowing the POS supertag of the same word.

Further investigations using this architecture are needed in future work to prove the efficiency of this model. However, its simple nature offers the opportunity to manipulate and adapt it easily, whether by modifying its structure or by simply adding new heads tailored to specific tasks.

Table 3 presents our results on the POS supertagging task compared to MElt tagger results reported by Denis and Sagot (2012). We observe that the model achieves state-of-the-art results, demonstrating that it can learn features relevant for both TLG and POS supertagging.

Model	Accuracy
MElt tagger (Denis and Sagot, 2012)	97.70
Phase 1 Single-head model	99.53
Phase 1 Multi-head model	99.57
Final Single-head VAEoTL	99.55
Final Multi-head VAEoTL	99.56

Table 3: Model performance in percent for French POS tagging on the TLGbank.

5 Contributions and limitations

With the goal in mind to provide a tool allowing to properly represent the syntax of input sentences formulated in natural language, we hope that future works will be able to extend the capacities of this architecture in order to exploit this syntax-semantic interface. While our model has not improved the state of the art of French TLG supertagging, it presents an accessible and simple fine-tuning of existing transformer-based models. Its modular architecture eases the adaptation of other existing techniques such as beam search to obtain more than a single prediction per word.

However, this model fails to modelize the internal structure of the syntactic types in the sense that it does not learn to create new composed types (N/N, S\NP) by assembling atomic types (N, NP, S). The current state of the art presented by Kogkalidis and Moortgat (2022) solves this problem by using a graph-theoretic perspective.

6 Conclusion

In this work, we investigated the different ways to regularize and fine-tune a supertagger for the French language, exploiting pre-trained unlabeled word embedding and a customized procedure utilizing a VAE architecture. We used a gold-standard annotated corpus, TLGbank, to train a simple and adaptable model able to compete with the current state of the art of supertaggers. We have shown experimentally that a VAE can be used

[4] The software used is available at the following github page for reproducibility of results: `https://github.com/gaetanmargueritte/ftlgsupertagger`

to improve model regularization and that overall accuracy can be improved by using a multi-headed architecture.

Acknowledgments

We thank both anonymous reviewers for their helpful comments and suggestions. We also want to thank Michael Moortgat for presenting his work and for his insightful comments that allowed us to gain a novel point of view on the supertagging task, and Yuta Takahashi for his continuous support that made easier the researches that led to this paper. This work was partially supported by JST CREST Grant Number JPMJCR20D2, Japan.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. *Building a Treebank for French*, pages 165–187. Springer.

Kasimir Ajdukiewicz. 1935. Die syntaktische konnexität. *Studia Philosophica*, 1:1–27.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Comput. Linguist.*, 25(2):237–265.

Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29(1):47–58.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a dependency treebank to a categorial grammar treebank for Italian. In *Eighth International Workshop on Treebanks and Linguistic Theories*.

Pascal Denis and Benoît Sagot. 2012. Coupling an annotated corpus and a lexicon for state-of-the-art pos tagging. *Language Resources and Evaluation*, 46(4):721–736.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding.

Philippe de Groote. 2001. Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 252–259, Toulouse, France. Association for Computational Linguistics.

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 505–512, Sydney, Australia. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Konstantinos Kogkalidis and Michael Moortgat. 2022. Geometry-aware supertagging with heterogeneous dynamic convolutions.

Yusuke Kubota and Robert D Levine. 2020. *Type-Logical Syntax*. MIT Press.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Michael Moortgat. 1997. Categorial type logics. In *Handbook of Logic and Language*.

Richard Moot. 2015. A type-logical treebank for French. *Journal of Language Modelling*, 3(1).

Glyn Morrill. 1994. *Type Logical Grammar: Categorial Logic of Signs*. Springer.

Glyn Morrill, Oriol Valentín, and Mario Fadda. 2011. The displacement calculus. *Journal of Logic, Language and Information*, 20:1–48.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. 2013. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1042–1051, Sofia, Bulgaria. Association for Computational Linguistics.

Experiments in training transformer sequence-to-sequence DRS parsers

Ahmet Yıldırım and Dag Trygve Truslew Haug
Department of Linguistics and Scandinavian Studies
University of Oslo
{ahmetyi,daghaug}@uio.no

Abstract

This work experiments with various configurations of transformer-based sequence-to-sequence neural networks in training a Discourse Representation Structure (DRS) parser, and presents the results along with the code to reproduce our experiments for use by the community working on DRS parsing. These are configurations that have not been tested in prior work on this task. The Parallel Meaning Bank (PMB) English data sets are used to train the models. The results are evaluated on the PMB test sets using Counter, the standard evaluation tool for DRSs. We show that the performance improves upon the previous state of the art by 0.5 (F_1%) for PMB 2.2.0 and 1.02 (F_1%) for PMB 3.0.0 test sets. We also present results on PMB 4.0.0, which has not been evaluated using Counter in previous research.

1 Introduction

Discourse representation structures (DRSs) are a way of representing meaning based on Discourse Representation Theory (Kamp and Reyle, 1993; Kamp et al., 2011). In addition to predicate-argument structures, DRSs express temporal relations, anaphora, modals, negation, and presuppositions, and can be further employed by other automatic processes to understand natural language.

The task of mapping sentences to their DRS meaning representations is called DRS parsing. There now exists a large dataset with DRSs for corpus examples, the Groningen Parallel Meaning Bank (PMB, Abzianidze et al. 2017), which makes it possible to train deep neural networks of the kinds that provide state-of-the-art performance on a variety of NLP tasks these days.

Recent work has explored a variety of neural network architectures for this task, but curiously, little work has been done using the otherwise widely utilized transformer-based encoder-decoder architecture. In this paper, we report on such experiments using Wordpiece (Wu et al., 2016) to tokenize the input and output, and train a sequence-to-sequence model where the encoder is a pre-trained BERT model (Devlin et al., 2018) and the decoder consists of randomly initialized transformer layers with cross attention. We experiment with different hyperparameter settings and achieve higher performance than in previous work.

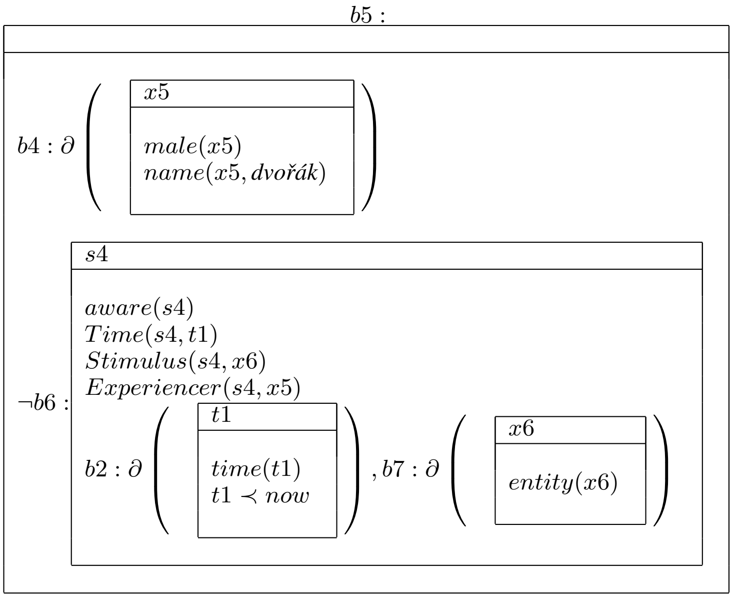
In the remainder of this paper, we briefly introduce DRSs and the PMB dataset in Section 2. We then survey previous work on DRS parsing in Section 3. Section 4 provides the machine learning configurations we used. Section 5 presents the results and a comparison with prior work. Section 6 remarks on our overall takeaways from this work.

2 Data

Historically, DRSs are represented in a box notation designed for human readability. The left-hand side of Figure 1 shows the representation of *Dvořák was not aware of it*. The negated content *was not aware of it* is represented as a separate embedded box labeled $b6$. Moreover, the sentence contains three presuppositions that must be resolved: these are the boxes $b2$, $b4$, $b7$ (shown inside a presupposition operator ∂), corresponding to the referents $t1$ (time at which the sentence holds), $x5$ (the referent of the proper name *Dvořák*), and $x6$ (the entity to which *it* refers). The latter two referents appear inside the negated box, because they are syntactically in the scope of negation, but they must in fact be interpreted in a wider context (i.e. the text entails that there exists a reference for *it* and a time at which the state of Dvořák not being aware of it held). For more details about DRSs, we refer to (Kamp and Reyle, 1993; Kamp et al., 2011).

The release of the PMB offered for the first time relatively large amounts of text annotated with deep

Proceedings of the 15th International Conference on Computational Semantics pages 83–88
June 21–23, 2023. ©2023 Association for Computational Linguistics



b4 REF x5
b4 Name x5 ”dvořák”
b4 PRESUPPOSITION b5
b4 male ”n.02” x5
b2 PRESUPPOSITION b6
b6 Time s4 t1
b2 REF t1
b2 TPR t1 ”now”
b2 time ”n.08” t1
b5 NEGATION b6
b6 REF s4
b6 Experiencer s4 x5
b6 aware ”a.01” s4
b6 Stimulus s4 x6
b7 REF x6
b7 PRESUPPOSITION b6
b7 entity ”n.01” x6

Figure 1: Box and clause notation of the DRS for *Dvořák was not aware of it*

semantic representations in the form of DRSs. In PMB, DRSs are given in a more machine-friendly clause format shown on the right-hand side of Figure 1. We refer to Liu et al. (2021) for more details on the conversion. Notice that the clause format also contains references to WordNet synsets (”n.02” etc.). Parsing to PMB representations therefore also involves word sense disambiguation.

There are several releases of the PMB, differing in size and also in some choices of representation. Previous work has focused on version 2.2.0, which contains 5929 DRSs for English sentences, and version 3.0.0, which has 8403 English DRSs. The latest release, version 4.0.0, has 10715 English DRSs. All versions also have data in Dutch, German, and Italian, which we ignore here. Each release has various data files available at the website (Parallel Meaning Bank, 2020), but also provides a separate download that contains only the data relevant for experiments in semantic parsing (“exp_data”).

The annotations are done automatically and then manually corrected. The representations are labeled with bronze, silver, or gold status. Bronze sentences have no manual correction, silver sentences have a partial manual correction and gold sentences have a full manual correction. The dev, test, and eval datasets consist of gold sentences only.

3 Related work

Before the advancement of machine learning systems, rule-based approaches were proposed as a solution for the DRS parsing task. Work within this research track mainly tried to resolve anaphora (Johnson and Klein, 1986; Wada and Asher, 1986), scope ambiguities, and presuppositions (Bos, 2001) on short English text. Later, the Boxer Software (Bos, 2008) used syntactic parses from a Combinatory Categorial Grammar (Clark and Curran, 2004) to produce DRSs. In another line of work, DRSs were represented as graphs obtained from dependency structures of sentences (Le and Zuidema, 2012) and ranked according to their probabilities of representing the sentence where the probabilities are obtained from a corpus by computing word-to-word alignments using an external tool (Och and Ney, 2003).

System	Model	Input
Liu et al. (2019)	transformer	characters
van Noord et al. (2018)	seq2seq	characters
van Noord (2019)	seq2seq	characters
Evang (2019)	stack LSTMs	word embeddings
Fancellu et al.[1]	bi-LSTM	word embeddings

Table 1: Systems in the shared task on DRS parsing

With the advent of language models and a data set like the PMB which is large enough for fine-tuning such models, it became possible to employ neural nets for DRS parsing. All systems in the recent shared task on DRS parsing (Abzianidze et al., 2019b) used neural architectures, as shown in Table 1 adapted from Abzianidze et al. (2019a).

Most systems used a variety of a sequence-to-

[1] No system description was submitted to the proceedings but the system is described in Abzianidze et al. (2019a).

sequence LSTM (Hochreiter and Schmidhuber, 1997), though Liu et al. (2019) used a transformer model (Vaswani et al., 2017). The system input was either character-level representations or word embeddings obtained from one of the widely utilized BERT language models. Later, van Noord et al. (2020) combined these two inputs to their LSTM sequence-to-sequence system, which also used an attention mechanism (Vaswani et al., 2017), arguing that this improved results even when added to the rich BERT embeddings. They also report results using a transformer model but were unable to beat the LSTM sequence-to-sequence model in this way. Their work reported state-of-the-art results for PMB 2.2.0 and PMB 3.0.0 English datasets. Later, Liu et al. (2021) used BERT word embeddings and position embeddings as input and expression of DRSs as trees as output to train a transformer sequence-to-sequence model. They reported a slight improvement (0.4%) upon the state of the art for PMB 2.2.0 dataset. As far as we know these are the only attempts at using the transformers architecture which is the default approach across many NLP tasks today.

4 Machine learning configurations

We use sequence-to-sequence modeling with two main components: an encoder and a decoder. HuggingFace transformers library (Wolf et al., 2020) provides the class EncoderDecoderModel to configure such models. The models are trained with various configurations of this class to test the performance.[2] For the encoder side, 7 configurations are tested. The first two options test different sizes of random initialization (No-PT). One configuration is 6 layers of 768 hidden layer size (No-PT, 6x768), and the other is 8 layers of 512 hidden layer size (No-PT, 8x512). The rest of the encoders are pre-trained models: bert_base_cased, bert_base_uncased, bert_large_cased, and bert_large_uncased. For the decoder side, we use the size of 6x768 with the 6x768 sized No-PT encoder, and 8x512 with both a No-PT encoder setup and the pre-trained encoders. All decoder side weights are randomly initialized. The 12x768 networks have 12 and 8x512 networks have 8 attention heads per layer. For the 6-layer setups, two configurations are used: 6 and 12 attention heads per layer. All decoders include cross-attention layers as it is effective in sequence-to-sequence training (Gheini et al., 2021).

The work by van Noord et al. (2020) reports that updating pre-trained encoder weights always resulted in poor performance. Therefore, a similar approach is followed and the encoder side weights are frozen whenever we use the pre-trained encoders. When the 12x768 decoder is used with No-PT encoders, the number of parameters to be trained gets too high and a model cannot be trained. Thus, the 12x768 decoder configuration is only used together with the frozen pre-trained encoders.

Our configurations get inputs as sub-word tokens derived from the widely utilized Wordpiece tokenizer (Wu et al., 2016). With the pre-trained encoders, the tokenizer used to train that pre-trained model is used as the input tokenizer. For No-PT encoders and for the decoder side output, we train custom Wordpiece tokenizers for each dataset. Since the output of DRS parsing is a DRS, the serializations of DRSs are tokenized using the relative clause notation introduced in van Noord (2021). All custom tokenizers are trained with: a vocabulary size of 25000, the minimum frequency for consideration of a token is set to 3, and the maximum tokenization length (maximum number of tokens for one sentence) is set to 512 tokens.

To test the effect of using different parameters introduced in this section, the other hyperparameters are fixed such as the optimizer, learning rate, and loss function. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001, and use the negative log-likelihood loss (Yao et al., 2020) to compute the loss in each batch between the model output and the expected output. We set the batch size to 16 sentences as this is the amount the graphic cards could handle. For any other parameter, the default value defined by version 4.17.0 of the Transformers library is used for the objects of types BertConfig, EncoderDecoderModel, EncoderDecoderConfig, and BertModel.

We use four Nvidia V100 32GB GPUs to train the models. The training time depends on the number of parameters and the number of attention heads. For one configuration, training for PMB 2.2.0 sets takes around one day, and training for PMB 3.0.0 and PMB 4.0.0 sets takes around 2 days. When four GPUs are used, it takes around one week to train all models in all configurations. We train for each configuration only once.

[2] The replication code is published under GitHub: `https://github.com/textlab/seq2seqDRSparser`

number of			PMB 2.2.0		PMB 3.0.0		PMB 4.0.0		
parameters	**Encoder**	**Decoder**	**dev**	**test**	**dev**	**test**	**dev**	**test**	**eval**
139,636,648	No-PT, 6x768-6	6x768-6	86.65	87.65	89.78	89.16	89.05	89.48	87.32
139,636,648	No-PT, 6x768-12	6x768-12	86.45	87.8	89.64	89.48	89.03	89.45	87.51
102,389,160	No-PT, 8x512-8	8x512-8	86.87	87.26	89.46	89.64	89.06	89.61	87.38
55,775,144	bert_base_uncased	8x512-8	87.17	88.45	89.69	89.78	89.1	89.79	87.2
55,775,144	bert_base_cased	8x512-8	87.51	88.23	**89.96**	89.89	89.19	89.9	**88.18**
133,633,960	bert_base_uncased	12x768-12	87.41	88.18	89.57	89.66	**89.4**	**90.26**	87.36
133,633,960	bert_base_cased	12x768-12	**87.53**	**89.23**	89.78	**90.32**	88.07	89.04	86.9
134,421,160	bert_large_uncased	12x768-12	86.93	88.56	89.08	88.65	88.71	89.6	87.29
134,421,160	bert_large_cased	12x768-12	86.9	88.27	89.39	90.03	88.81	90.12	87.42
≈106 million	van Noord et al. (2020)		86.1	88.3	88.4	89.3			
≈106 million	Liu et al. (2021)			88.7					

Table 2: F1% scores of various models. Prior works by van Noord et al. (2020) and Liu et al. (2021) use similar hyperparameter settings. No-PT: No pre-training. AxB-C: A hidden layers of size B and C attention heads per layer.

5 Results

The performance scores are computed for *dev*, *test*, and *eval*[3] sets for each dataset. To compute the scores, we used the Counter tool provided by van Noord (2022). To make the results comparable with the previous work, the version of Counter with the same version tag for each release of the datasets is used. For the 4.0.0 release of the datasets, we use the latest version of the code as 4.0.0 is the newest release. The models are trained for at least 80 epochs for all datasets and stopped if there is no increase in performance for the last five epochs. Table 2 presents the results obtained for the configurations mentioned in the previous section.

Previous work used gold and silver data for finetuning. Our work uses the train sets as is and does not prioritize gold, silver, or bronze sentences. Therefore, one training epoch consists of using each sentence only once, and, the learning rate is not changed throughout the training. Even with this setup, we observe that two configurations with randomly initialized encoders and decoders (No-PT) outperform the previous state of the art for PMB 3.0.0. Moreover, using pre-trained encoders performed even better. For the PMB 2.2.0 test set, our setup slightly improved upon the previous state-of-the-art. For PMB 4.0.0, to the best of our knowledge, this is the first time model performances are reported using Counter.[4] For all configurations, using the larger BERT pre-trained models bert_large cased and uncased do not perform better than the smaller bert_base cased and uncased. We observe that using cased pre-trained models generally performed better.

Table 3 presents detailed performances for different kinds of DRS clauses in the clause notation. The results are in line with what van Noord et al. (2020, Table 10) report. *DRS operators* have the highest performance which indicates that structural features of a DRS is captured better than the other features. One reason may be that the test set of all releases of PMB represent relatively short sentences that have structurally simple DRSs. Roles (i.e. binary predicates like `Agent`, `Theme`, `MadeOf` etc.) and concepts (which includes word sense disambiguation because each concept is a WordNet synset) are harder to capture, especially verbal concepts. Performance for adjective and adverbs increase with each release of the datasets, probably reflecting improving standards of annotation.

van Noord et al. (2020) observe that parsing performance decreases with sentence length. In Haug et al. (2023) we show that the same holds for our system. Nevertheless, the PMB test set with its uniformly quite short sentences (the large majority is ¡ 10 tokens) does not lend itself to study the effect of sentence length, and in Haug et al. (2023) we test the system on more realistic sentence lengths.

6 Conclusions

Our work presents the effect of using various sizes of transformer-based encoders and decoders in sequence-to-sequence neural networks with the

[3] PMB publishes the *eval* dataset only for the 4.0.0 release

[4] Poelman et al. (2022) reports using the SMATCH (Cai and Knight, 2013) tool by comparing Discourse Representation Graphs (DRG), a simpler form of DRSs, on PMB 4.0.0.

	PMB 2.2.0	**PMB 3.0.0**	**PMB 4.0.0**
Operators	95.58	96.55	95.78
Roles	88.2	89.88	89.01
Concepts	85.35	86.99	87.95
Nouns	90.68	91.35	92.28
Verbs	73.45	75.81	73.83
Adjectives	67.43	78.98	82.53
Adverbs	50.0	73.85	85.5
Events	72.37	76.46	75.96

Table 3: F1% scores in different PMB versions' test sets for different types of DRS clauses in the clause notation. The configuration is bert_base_cased encoder with 12x768-12 decoder that is trained for each PMB version separately.

subword tokenizer Wordpiece on the task of DRS parsing. The performances of the use of various sizes and pre-trained encoder configurations are reported. This work shows that the performance of DRS parsing increases with some of these configurations. We believe that applying our setup could improve the performance of other related tasks. For example, Liu et al. (2021) explores multilingual DRS parsing based on transfer from English translations which, as we have shown here, could be better parsed with our approach.

Our results provide a new state-of-the-art of what can be achieved in a vanilla setup of transformer networks with raw text input and clause format DRS output. While it is likely that the results can be improved with better language models, or by fine-tuning strategies similar to those of van Noord et al. (2020) (prioritizing gold data over silver and bronze), we think more substantial improvements can come from working on the input and output representations. On the output side, we plan to experiment with other ways of expressing DRSs such as the format introduced by Liu et al. (2021). On the input side, we believe that syntactic dependency parses contain much information that is useful to DRS parsing, such as predicate argument structures. We are currently experimenting with rule-based extraction of relevant information from UD trees and ways of adding this information to the input.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos. 2019a. The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos, editors. 2019b. *Proceedings of the IWCS Shared Task on Semantic Parsing*. Association for Computational Linguistics, Gothenburg, Sweden.

Johan Bos. 2001. Doris 2001: Underspecification, resolution and inference for discourse representation structures. *ICoS-3, Inference in Computational Semantics*, pages 117–124.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 277–286. College Publications.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 103–es, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. 2021. Cross-attention is all you need: Adapting pretrained Transformers for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1765, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dag Trygve Truslew Haug, Jamie Y. Findlay, and Ahmet Yıldırım. 2023. Ethe long and the short of it: DRASTIC, a semantically annotated dataset containing sentences of more natural length. In *Proceedings of the 4th International Workshop on Designing Meaning Representation(DMR 2023)*, Nancy, France. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mark Johnson and Ewan Klein. 1986. Discourse, anaphora and parsing. In *Coling 1986 Volume 1: The 11th International Conference on Computational Linguistics*.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer Academic Publishers.

Hans Kamp, Josef van Genabith, and Uwe Reyle. 2011. Discourse Representation Theory. In D. M. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 15, pages 125–394. Springer.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Phong Le and Willem Zuidema. 2012. Learning compositional semantics for open domain semantic parsing. In *Proceedings of COLING 2012*, pages 1535–1552, Mumbai, India. The COLING 2012 Organizing Committee.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019. Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, Mirella Lapata, and Johan Bos. 2021. Universal discourse representation structure parsing. *Computational Linguistics*, 47(2):445–476.

Ri van Noord. 2019. Neural boxer at the IWCS shared task on DRS parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord. 2022. Rikvn/drs_parsing: Scripts to evaluate scoped meaning representations. `https://github.com/RikVN/DRS_parsing`. Accessed: 2022-07-19.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring Neural Methods for Parsing Discourse Representation Structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51.

Parallel Meaning Bank. 2020. Index of /releases. `https://pmb.let.rug.nl/releases/`. Accessed: 2022-07-19.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022. Transparent semantic parsing with Universal Dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Rik van Noord. 2021. *Character-based Neural Semantic Parsing*. Ph.D. thesis, University of Groningen.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hajime Wada and Nicholas Asher. 1986. BUILDRS: An implementation of DR theory and LFG. In *Proceedings of the 11th Coference on Computational Linguistics*, COLING '86, page 540–545, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Hengshuai Yao, Dong-lai Zhu, Bei Jiang, and Peng Yu. 2020. Negative log likelihood ratio loss for deep neural network classification. In *Proceedings of the Future Technologies Conference (FTC) 2019*, pages 276–282. Springer International Publishing.

Unsupervised Semantic Frame Induction Revisited

Younes Samih **Laura Kallmeyer**
Department of Computational Linguistics
Heinrich Heine University Düsseldorf,
Düsseldorf, Germany
{samih,kallmeyer}@hhu.de

Abstract

This paper addresses the task of semantic frame induction based on pre-trained language models (LMs). The current state of the art is to directly use contextualized embeddings from models such as BERT and to cluster them in a two step clustering process (first lemma-internal, then over all verb tokens in the data set). We propose not to use the LM's embeddings as such but rather to refine them via some transformer-based denoising autoencoder. The resulting embeddings allow to obtain competitive results while clustering them in a single pass. This shows clearly that the autoendocer allows to already concentrate on the information that is relevant for distinguishing event types.

1 Introduction

In natural language processing, Semantic Frame Induction refers to the task of clustering target word instances, specifically verbs, in a corpus according to their semantic frames in a given context. For example, in the sentences:

(a) *The price of LNG is rising, which makes the European economy unstable.*

(b) *Gold value fell 2% in January after climbing 5% in August.*

(c) *Adam climbs dangerous cliffs.*

We would like to cluster the verbs in (a) and (b) in one group and (c) in another. The problem of verb semantic frames induction has received its share of attention, particularly in the SemEval 2019 shared task (Subtask-A) (QasemiZadeh et al., 2019a), in which the gold labels are annotated according to the FrameNet (Baker et al., 1998) frames inventory. Frame-semantic resources are prohibitively expensive and time-consuming to construct due to difficulties in the frame definitions, as well as the complexity of the construction and annotation tasks, that require expert knowledge in lexical event semantics. To overcome these issues, researchers proposed to automate the process of FrameNet construction through unsupervised techniques (Titov and Klementiev, 2011; Modi et al., 2012; Ustalov et al., 2018). Unsupervised semantic frame induction methods help to automatically build high-coverage frame-semantic resources. Up until recently, state-of-the-art results for semantic frame induction were dominated by a series of models leveraging contextualised pretrained language model representations to cluster instances of verbs according to the frames they evoke (Arefyev et al., 2019; Anwar et al., 2019). In recent work, Ribeiro et al. (2020) achieve state-of-the-art results by applying a graph-clustering algorithm based on Chinese whispers (Biemann, 2006) by using contextualized representations of frame-evoking verbs from BERT (Devlin et al., 2019)). Another approach has been proposed by Yamada et al. (2021b), who also use masked word embeddings and two-step clustering: each target instance is represented by three contextualized embeddings in a text, clustering is performed first over instances of the same verb and then across all verbs. However, these previous methods have one crucial shortcoming. As transformers based contextual words embeddings are originally designed to be fine-tuned on each downstream task to attain their optimal performance, it is unclear how best to extract representations of frame-evoking verbs from them, which are broadly applicable across diverse word-related tasks. In this paper, we further explore the use of LM representations by leveraging transformer-based Sequential Denoising Auto-Encoder (Wang et al., 2021) (TSDAE) embeddings to tackle the aforementioned problem. The proposed method achieves state-of-the-art performance on the frame induction task.

Proceedings of the 15th International Conference on Computational Semantics pages 89–93
June 21–23, 2023. ©2023 Association for Computational Linguistics

The contributions of this paper are three-fold:

- To the best of our knowledge, we are the first to adapt Transformer-based Sequential Denoising Auto-Encoder for semantic frame induction.
- Our method does not require two step-clustering, which is essential in most recent semantic frame induction models (Arefyev et al., 2019; Yamada et al., 2021a).
- Our clustering model outperforms recent state of the art systems for semantic frame induction on the SemEval 2019 shared task (Subtask-A) benchmark.

2 Method

In this section, we provide a brief description of TSDAE, and introduce the different components of our semantic frames induction model. The proposed model works in three stages:

- We train TSDAE on unlabeled sentences from the target task,
- then use its encoder to extract embeddings of the frame-evoking verb, associating each target verb instance with its representative vector.
- Finally, We perform clustering on these representative vectors.

TSDAE based word embeddings TSDAE, as shown in Figure 1, is a popular unsupervised learning algorithm based on an encoder decoder architecture. The model is a modified encoder-decoder Transformer where the key and value of the cross-attention are both restricted to yield sentence embedding only (Wang et al., 2021). The encoder maps the original input vector to a hidden representation, and the decoder maps the hidden representation back to the original input space. During training, noise is added to each input text by deleting or swapping a fraction of all tokens (we delete 60% of words in our experiments[1]), encoding the noisy text and reconstructing the embedding using the decoder module. The autoencoder minimizes the reconstruction error by approximating an identity function (Ng et al., 2011). A good reconstruction quality means that the semantics must be well captured in the word embeddings by the encoder. After training, the decoder module is discarded and the encoder is used to extract word representations. We re-implemented the TSDAE algorithm based on Huggingface's Transformers. [2] The algorithm is a simplified version of methods described in (Wang et al., 2021).

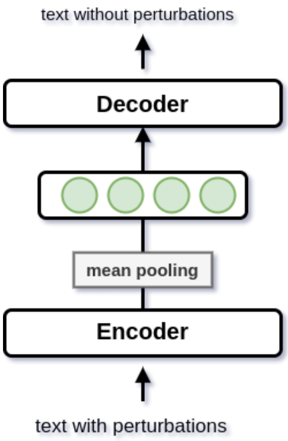


Figure 1: Architecture of TSDAE

Clustering After training TSDAE on unlabeled sentences from the target task, contextualised vectors for the frame evoking verbs in the sentences are calculated, and then clustered using agglomerative clustering with average linkage and cosine distance. The number of clusters is defined based on clusters maximizing the average silhouette score of all frame evoking verbs.

3 Experiments

Data	#Verbs	#Frames	#Examples
Dev.	600	41	588
Test.	4620	149	3346
All.	5220	190	3934

Table 1: Statistics of the dataset from the SemEval 2019 shared task

In this section, we introduce the datasets and experiment settings used for semantic frame induction. We also present the evaluation results of each model and compare them against existing semantic frame induction systems.

[1] We performed several auxiliary experiments on the development dataset to determine the optimal noise type and its ratio. It was discovered that removing the verb that evokes the semantic frame did not produce the most favorable outcome. Instead, setting the deletion ratio to 0.6 resulted in the most effective performance.

[2] https://github.com/huggingface/transformers

Model	Embeddings	#C	Pu	Ipu	Fpu	Bcp	Bcr	Bcf
1-cluster-per-verb	-	273	82.16	66.95	73.78	75.98	57.33	65.35
Anwar et al. (2019)	Elmo	150	72.4	81.49	76.68	62.17	75.27	68.1
Arefyev et al. (2019)	Bert	272	78.68	77.62	78.15	70.86	70.54	70.7
Ribeiro et al. (2019)	Bert	222	72.84	77.84	75.25	61.25	69.96	65.32
Ribeiro et al. (2020)	Bert	-	-	-	79.97	-	-	73.07
GA	Bert	227	80.26	79.05	79.65	73.52	71.88	72.69
	RoBERTa	192	80.35	81.9	81.12	73.61	75.7	**74.64**
TSDAE+GA	Bert	208	79.87	79.87	79.87	72.89	73.41	73.15
	RoBERTa	160	80.17	**84.33**	**82.2**	**73.62**	**78.67**	**76.06**

Table 2: Experimental results. #C denotes the number of frame clusters.Scores in bold denote significant improvements over the baseline. GA designates group average clustering.

3.1 Dataset

We use the SemEval 2019s Task 2 (QasemiZadeh et al., 2019b) as the benchmark datasets to evaluate our models and to facilitate comparison with related work. This dataset contains a subset of sentences extracted from the Penn Treebank 3.0 (Marcus et al., 1993) annotated with FrameNet Frames and tagged with morphosyntactic information in the CoNLL-U format (Buchholz and Marsi, 2006). Table 1 lists the statistics of the dataset.

3.2 Evaluation Measures

We evalute our approach using the six evaluation metrics[3] employed on the SemEval's task: Purity (Pu), inverse-Purity (Ipu) and their harmonic mean (Fpu) as proposed in (Steinbach et al., 2000), as well as The harmonic mean of BCubed's precision and recall (denoted by Bcp, Bcr, and Bcf respectively) (Bagga and Baldwin, 1998).

3.3 Experimental Settings

For all experiment we use BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). In all cases we use the implementations from the HuggingFace Transformers toolkit (Wolf et al., 2019).

Baselines A very competitive baseline for frame-semantic induction is the SemEval'19 shared task 2 winning system by Arefyev et al. (2019). They use a two-step agglomerative clustering model. First, it groups examples to a relatively small number of large clusters, exploiting dense vector representations of the target word in a context obtained from hidden layers of BERT model. It merges verbs that evoke the same frame together while not taking into account homonyms. Then splits each of them into smaller clusters using the TF-IDF representations from substitutes generated for the target word by BERT masked LM to disambiguate all homonyms. An even stronger baseline is the system by Ribeiro et al. (2020) who apply Chinese whisper (Biemann, 2006), a graph-clustering algorithm, to a graph using contextualised representations of frame-evoking verbs as its nodes. Anwar et al. (2019) introduced a simpler system based on the agglomerative clustering of contextualised representations extracted from hidden layers of ELMo (Peters et al., 2018). Finally, we also evaluated one additional, simpler baseline (1-cluster-per-head) that treats all instances of one verb as one cluster.

3.4 Results

We evaluate the performance of each model and report the BCubed F1-scores in Table 2, along with the results from other semantic frame induction systems. Our model (TSDAE+GA) based on TSDAE and group average clustering outperforms the other methods on both Fpu and Bcf by a large margin. It achieves the highest Fpu score of 82.2 and also got the highest Bcf score of 76.06. The graph-based clustering by Ribeiro et al. (2020) proved to be the most competitive baseline, yielding decent scores according to all six measures. Finally, our RoBERTa group average model (GA) relying on hard clustering algorithms showed a slight increase in performance when compared to that of the graph-based model, justifying the more elaborate (TSDAE+GA) method. It is also worth noting that the Bert based group average model obtain a slightly worse or identical results.

[3] We use the standard evaluation script from the SemEval'19 shared task to calculate all the results. http://pars.ie/lr/semeval2019-task2/semeval-2019-task2-scorer.zip

4 Analysis

We extracted the cluster signatures and manually inspected all of the semantic frame clusters produced by TSDAE+GA, our best system, along with their associated verbs in order to scrutinize the emerging semantic classes and gain insight into annotator decisions. We found that the most prominent reason for incorrect clustering was due to the hard partitioning output, while the evaluation dataset contained fuzzy clusters. We also observed that the semantic distinctions that are easier for humans to make often elude representation models, and that discriminating between similar and highly associated but dissimilar verbs remains a challenge for most systems. Moreover, we noticed that the effectiveness of the models differ depending on the semantic Frames, indicating discrepancies in the quality of representations for verbs from diverse domains. Interestingly, we found that many clusters included an incoherent mix of multiple semantic frames along with an incoherent set of verbs. This suggests that frame induction should not be treated solely as a verb clustering task as it requires a distinct and separate approach.

5 Conclusion

In this paper, we introduced the first implementation of TSDAE for unsupervised frame induction and demonstrated that our method outperforms previous approaches in SemEval'19 shared task 2, setting a new state-of-the-art. Our error analysis revealed that a major source of incorrect clustering stemmed from the hard partitioning output, while the evaluation dataset consisted of fuzzy clusters. For future work, we aim to extend our work to the multi-lingual setup in future studies.

Acknowledgments

The work presented in this paper was partly financed by the Deutsche Forschungsgemeinschaft (DFG) within the project "Unsupervised Frame Induction (FInd)". We wish to thank three anonymous reviewers for their constructive feedback and helpful comments.

References

Saba Anwar, Dmitry Ustalov, Nikolay Arefyev, Simone Paolo Ponzetto, Chris Biemann, and Alexander Panchenko. 2019. HHMM at SemEval-2019 task 2: Unsupervised frame induction using contextualized word embeddings. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 125–129, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Nikolay Arefyev, Boris Sheludko, Adis Davletov, Dmitry Kharchev, Alex Nevidomsky, and Alexander Panchenko. 2019. Neural GRANNy at SemEval-2019 task 2: A combined approach for better modeling of semantic relationships in semantic frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 31–38, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Chris Biemann. 2006. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, New York City. Association for Computational Linguistics.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ashutosh Modi, Ivan Titov, and Alexandre Klementiev. 2012. Unsupervised induction of frame-semantic representations. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 1–7, Montréal, Canada. Association for Computational Linguistics.

Andrew Ng et al. 2011. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Behrang QasemiZadeh, Miriam R. L. Petruck, Regina Stodden, Laura Kallmeyer, and Marie Candito. 2019a. SemEval-2019 task 2: Unsupervised lexical frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 16–30, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Behrang QasemiZadeh, Miriam R. L. Petruck, Regina Stodden, Laura Kallmeyer, and Marie Candito. 2019b. SemEval-2019 task 2: Unsupervised lexical frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 16–30, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Eugénio Ribeiro, Vânia Mendonça, Ricardo Ribeiro, David Martins de Matos, Alberto Sardinha, Ana Lúcia Santos, and Luísa Coheur. 2019. L2F/INESC-ID at SemEval-2019 task 2: Unsupervised lexical semantic frame induction using contextualized word representations. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 130–136, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Eugénio Ribeiro, Andreia Sofia Teixeira, Ricardo Ribeiro, and David Martins de Matos. 2020. Semantic frame induction through the detection of communities of verbs and their arguments. *Applied Network Science*, 5(1):1–32.

Michael Steinbach, George Karypis, and Vipin Kumar. 2000. A comparison of document clustering techniques.

Ivan Titov and Alexandre Klementiev. 2011. A Bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1445–1455, Portland, Oregon, USA. Association for Computational Linguistics.

Dmitry Ustalov, Alexander Panchenko, Andrey Kutuzov, Chris Biemann, and Simone Paolo Ponzetto. 2018. Unsupervised semantic frame induction using triclustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 55–62, Melbourne, Australia. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Kosuke Yamada, Ryohei Sasano, and Koichi Takeda. 2021a. Semantic frame induction using masked word embeddings and two-step clustering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 811–816, Online. Association for Computational Linguistics.

Kosuke Yamada, Ryohei Sasano, and Koichi Takeda. 2021b. Verb sense clustering using contextualized word representations for semantic frame induction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4353–4362, Online. Association for Computational Linguistics.

Towards Ontologically Grounded and Language-Agnostic Knowledge Graphs

Walid S. Saba
Institute for Experiential AI
Northeastern University
w.saba@northeastern.edu

Abstract

Knowledge graphs (KGs) have become the standard technology for the representation of factual information in applications such as recommendation engines, search, and question-answering systems. However, the continual updating of KGs, as well as the integration of KGs from different domains and KGs in different languages, remains to be a major challenge. What we suggest here is that by a reification of abstract objects and by acknowledging the ontological distinction between concepts and types, we arrive at an ontologically grounded and language-agnostic representation that can alleviate the difficulties in KG integration.

1 Introduction

Knowledge graphs are by now the standard representation of knowledge repositories that are used in various applications, such as search, recommendation engines, and question-answering systems. While there are powerful KG tools, the semantic and conceptual side of KG technology is still partially ad-hoc. In particular, the continuous update and KG integration remain to be a challenge.

A Knowledge graph (KG) is a graph structure that can be viewed as a set of triples $\langle e_1, r, e_2 \rangle$ relating real-world entities e_1 and e_2 by a relation r to represent a real-world fact, as in the following examples:

$$\langle \mathit{RogerWaters}, \mathit{BornOn}, 01/08/1955 \rangle \quad (1)$$
$$\langle \mathit{PinkFloyd}, \mathit{StartedIn}, \mathit{London} \rangle \quad (2)$$
$$\langle \mathit{BarakObama}, \mathit{LivesIn}, \mathit{WhiteHouse} \rangle \quad (3)$$

From the triples above that we might have in some knowledge graph KG_1 we can immediately point to several issues that pose major challenges in constructing and maintaining KGs. We discuss these issues next.

2 Alignment and Continuous Change

Here are the main issues in the triples (1) through (3) above: First, in another knowledge graph KG_2 that we might want to integrate with KG_1 there might be another *Roger Waters* where the two entities might or might not be the same and thus an entity alignment must occur with the triple in (1). Another issue here is that the triple in (2) uses "*StartedIn*" to represent the fact that the Pink Floyd band started in London. Another KG might, instead, use the relation "*FormedIn*" and a match and an alignment between the two relations is needed. Finally, the integration of KG_1 with another KG might reveal that the triple in (3) is no longer valid and must thus be fused with new and updated information. At a minimum, then, the process of fusing together two or more KGs will first of all involve a tedious process of entity alignment (EA) (Zhang et. al., 2022), but more generally it will involve a process of continuous updating of information (Wang, et. al., 2022). Note that updating information and entity alignment both involve identifying if entities are the same (or not), where in one case we will perform a 'merge' and in the second an update.

Clearly then entity alignment is the most basic operation in any KG integration, and as such it has received the most attention. To match an entity e_1 in KG_1 with an entity e_2 in KG_2 embeddings in low dimensional space for both entities are constructed using neighboring information: related entities, immediate relations,

Proceedings of the 15th International Conference on Computational Semantics pages 94–98
June 21–23, 2023. ©2023 Association for Computational Linguistics

and attributes. Entities e_1 and e_2 are considered to have a match if their vector similarity is above a certain threshold. As such, different alignment techniques mainly differ in how the embeddings are constructed. In particular, they differ in what information is bundled in the embedding, and how far in the graph are other entities, relations and attributes are still considered to be in the "neighborhood". Zhu et. al. (2021), for example, report that spreading entity information across all relations, gathering information, and bringing it back to an entity's embedding, improves on embedding similarity and entity alignment. In (Lin, Y. and Liu, Z. et. al., 2016) it is further suggested that including all attributes and their values will also improve on an entity's embedding. Other approaches (e.g., Zhu et. al., 2023) will also include, besides attribute values, all string information corresponding to entity, relation, and attribute names. In all these approaches the ultimate goal is to improve on the construction of entity embeddings, in the hope of improving on the accuracy of entity alignment (i.e., entity matching). See (Zhang, R. et. al., 2022) for a good survey of various alignment techniques.

3 Reifying Abstract Objects

Regardless of the novelty and the progress made by various entity alignment algorithms, the accuracy of merging different knowledge graphs, especially ones that are continuously updated, will remain to be less than desired. In this section we will argue that the problem is to be handled not with constructing ever more reliable embeddings leading to more accurate alignments, but with how knowledge graphs are constructed in the first place. Specifically, we suggest that the answer lies in proposals that have been made in the study of semantics and formal ontology. In particular, we will appeal to conceptualism and the conceptual realism of Cocchiarella (2001), where we reify (or 'object-ify') abstract concepts in a manner that is consistent with our basic "cognitive capacities that underlie our use of language". This is essentially an extension of Davidsonian semantics (Davidson, 1967; Larson, 1998) where events are treated as entities, and is also in line with Moltmann's (2013) arguments that the ontology of natural language admits references to "tropes", which are particular instances of properties.

Let us make all of this clear with an example. Consider the knowledge graphs in figure 1 where we are representing the facts expressed by "*The musician Roger Waters was born in Great Bookham on 01/08/1955*". The knowledge graph in figure 1b has the same facts expressed in figure 1a but in an ontologically grounded and linguistically agnostic representation. First, note that instead of the ad-hoc naming of relations in 1a (e.g, **bornIn** and **bornOn**), in 1b we have primitive and language-agnostic relations where events are entities (e.g., "Birth") that have two essential properties, a time and a location and where these properties have specific values of specific *types*[1]. Note also that we are assuming here that these canonical names are done in the process of KG construction, and thus a 'Birth' event, regardless how it was named, will in the end translate to the same event.

In our representation, therefore, everything is an entity and the relations come from a fixed set of primitive and linguistically agnostic set of relations (the set of primitive relations are shown in figure 2). How we come up with these relations is beyond the scope of this short paper but see Smith (2005) for a discussion.

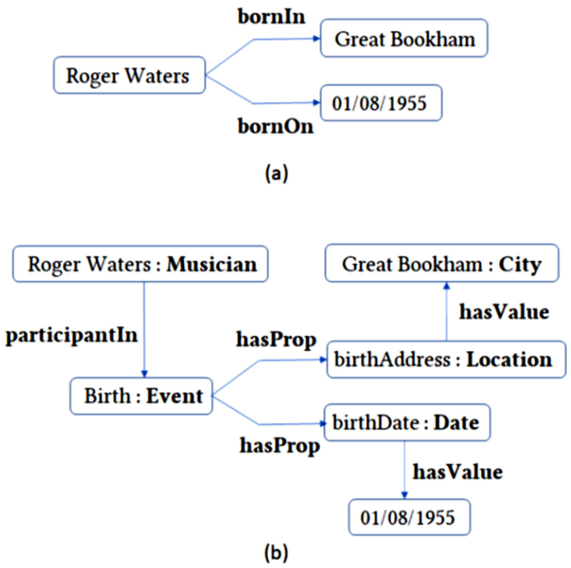


Figure 1: (a) A KG representing the facts expressed in "*The musician Roger Waters was born in Great Bokham on 01/08/1955*"; and (b) a language-agnostic KG representing the same facts.

[1] While both 'human' and 'teacher' are concepts, a **human** is a type, while a teacher is not. In fact, a 'teacher' is (ontologically, or metaphysically an object of type human that we call (or label as) teacher when it is the agent of a teaching **activity**.

Besides the primitive and linguistically agnostic representation, entities and attribute values in the knowledge graph of figure 1b are strongly-typed, where the types are assumed to exist in a strongly-typed hierarchy along the lines suggested in Saba (2020). Note that by making all entities typed we resolve the issue of separating knowledge graphs into two parts, one that has continuously updated information (⟨*RogerWaters*, *LivesIn*, *London*⟩) and one that has more static conceptual information such as ⟨*RogerWaters*, *IsA*, *Musician*⟩ (see Hao et. al., 2019 for a discussion on this issue).

eq(x, y)	x is identical to y
isPartOf(x, y)	x is part of y
inst(x, y)	x instantiates y
hasProp(x, y)	x inheres in y
exemp(x, y)	x exemplifies y
dep(x, y)	x depends on y
isA(x, y)	x is a subtype of y
precedes(x, y)	x precedes process y
participantIn(x, y)	y participates in occurrent x
hasAgent(x, y)	y is agent of occurrent x
hasObject(x, y)	y is object of occurrent x
hasValue(x, y)	attribute x has value y
realizes(x, y)	process x realizes y

Figure 2: The set of primitive and linguistically agnostic relations that are used in the knowledge graph. These are the only relations used and all other abstractions are entities (e.g., events, properties, states, etc. all of which are reified/object-ified),

Moreover, entity alignment will now be more accurate since the embedding of [*RogerWaters*: **Musician**] will only match the same musician in another knowledge graph, even if the entity was labeled differently, e.g. [*GeorgeRogerWaters*: **Musician**]. Besides adding semantic constraints that will improve knowledge integration, types are language agnostic and thus, like primitive relations, are easy to translate across languages. In figure 3 we show the isomorphic Arabic and French equivalents of the KG in figure 1b above.

4 Evaluation

Aside from the simple alignment of knowledge graphs written in different languages or different domains, we show here how the ontologically grounded and linguistically agnostic representation helps in the problem of entity alignment. First, we construct embeddings for triples where a change is made in one of the entities or in the relation:

e_1 = EMBED(⟨*RogerWaters*, *LivesIn*, *London*⟩)
e_2 = EMBED(⟨*RogerWaters*, *PlaceOfResidence*, *London* ⟩)
e_3 = EMBED(⟨*RogerWaters*, *LivesIn*, *Chelsea* ⟩)
e_4 = EMBED(⟨*RogerWaters*, *PlaceOfResidence*, *Chelsea* ⟩)

EMBED(⟨e_1, r, e_2⟩) returns an embedding that is the sum of the vectors of e_1, r, and e_2. In table 1 below we show the cosine similarity **cosim**(e_i, e_j) for i, j = 1,2,3,4 and for i ≠ j. The triples with a different entity (a different real-world fact) matched better than those with slightly different but semantically similar relation (i.e., same real-world fact).

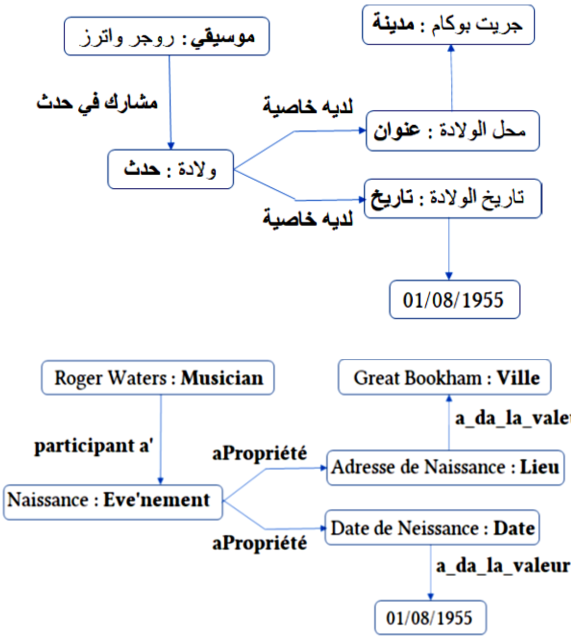
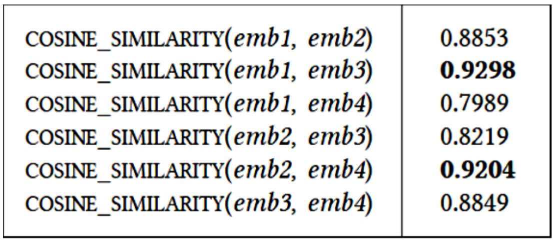


Figure 3: Since entity names, types, attribute values, and primitive relations are language agnostic, there's a straightforward automatic translation of the KG in figure 1b into isomorphic Arabic and French KGs.

Similar results were obtained by changing various semantically similar relations (e.g., **bornIn** vs. **placeOfBirth**, etc.)

The above shows that entity alignments across knowledge graphs would fail simply because of the ad-hoc labeling of relations in the knowledge graph. On the other hand, changing the location in the knowledge graph in 1b amounts to changing one embedding out of several that remain constant. In the example of figure 1b, a change in the location would result in a similarity of 0.688 only, and the alignment would clearly fail, as it should.



COSINE_SIMILARITY(*emb1*, *emb2*)	0.8853
COSINE_SIMILARITY(*emb1*, *emb3*)	**0.9298**
COSINE_SIMILARITY(*emb1*, *emb4*)	0.7989
COSINE_SIMILARITY(*emb2*, *emb3*)	0.8219
COSINE_SIMILARITY(*emb2*, *emb4*)	**0.9204**
COSINE_SIMILARITY(*emb3*, *emb4*)	0.8849

Table 1: Triples with different facts (locations) matched better than triples with the same facts (locations) but a relation that is worded slightly.

That is, an entity that is a participant in a birth event that happened in London should not match with an entity that is a participant in a birth event that happened in Chelsea, regardless of the entity name. Note that this true even in knowledge graphs in different languages (see figure 3), assuming, of course, that the embeddings of [London : **City**] and [**مدينة** : لندن] have a good cosine similarity, as one would expect.

5 Discussion

One important aspect to the representation we are suggesting is that it is language agnostic. This we claim is based on the fact that our representation has entities and primitive relations between them and that both of these are language agnostic. Thus the claim of universality is based two assumptions: (i) we are assuming that entities, including abstract entities such as those corresponding to properties, events, states, etc. are language-agnostic; (ii) we are assuming that our primitive relations (see figure 2) are also language agnostic. If both of these assumptions are correct, then our representation is language-agnostic, and the only remaining question would be "how universal are the primitive relations in figure 2?" A final answer to this question requires further experimentation.

Another important issue we could not discuss here for lack of space are the types that are associated with every entity and attribute value. These types are assumed to exist in a hierarchy of types that must also be language agnostic (that is, we are assuming that "the types of things we talk about/express facts about" are the same across languages). Admittedly, however, this claim might not be uncontroversial and further work needs to be done in this regard, although we believe the work of Saba (2020) is a step in the right direction. Another issue that should also be addressed is related to the mapping from natural language to our representation. As noted to us by one of anonymous reviewers, a fact such as "John sold the car to Bill" should, in theory, translate into the same sets of relations in the KG as the fact "Bill bought the car from John". While in both cases we will have a language agnostic representation with reified abstract objects for the 'buying' and 'selling' events where Bill and John are participants, these two facts will only be equivalent if there were some meaning postulate that relates the 'selling' and 'buying' events.

6 Concluding Remarks

In this short paper we suggested an ontologically grounded and linguistically agnostic representation for knowledge graphs. This representation, we believe will solve the major challenges facing knowledge graphs today, namely the difficulty in continuous updating of factual information (which requires static conceptual information to be separated from the more dynamic information), and the difficulty of knowledge graph integration which requires very accurate entity and relation alignment. We argued that our representation offers a solution to these (essentially semantic) problems.

A final remark we would like to make is related to an excellent point made by one the anonymous reviewers, name that the representation and the method we propose will work if the construction of every KG follows our methodology. This is true, and so in essence the representation we are suggesting can be thought of as a new standard for a semantically rigorous knowledge graph methodology. Although this is part of future work, this will entail building a natural language interpreter that will ensure the translation of every KG into the canonical and language agnostic representation suggested in this paper.

Acknowledgement

The feedback of colleagues at the Institute for Experiential AI is greatly appreciated.

References

Nino B. Cocchiarella. 2001. Logic and Ontology, *Axiomathes*, 12: 117-150.

Donald Davidson. 1967. The Logical Form of Action Sentences, in N. Rescher (ed.) *The Logic of Decision and Action* (pp. 81-120) University of Pittsburgh Press.

Junheng Hao, Muhao Chen, et. al. 2019. Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts, In *25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19)*

Richard K. Larson. 1998. Events and Modification in Nominals, *Semantics and Linguistic Theory*, Vol. 8, 145-168.

Yankai Lin, Zhiyuan Liu, Maosong Sun. 2016. Knowledge Representation Learning with Entities, Attributes and Relations, *In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (IJCAI-16).

Friederike Moltmann. 2013. *Abstract Objects and the Semantics of Natural Language*, Oxford Uni Press.

Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-Scale Knowledge Graphs: Lessons and Challenges, *Communications of The ACM* | August 2019, Vol. 62, No. 8, 36-43.

Walid Saba. 2020. Language and its commonsense: Where Formal Semantics Went Wrong, and Where it Can (and Should) Go, *Journal of Knowledge Structures and Systems* (JKSS), 1 (1):40-62

Barry Smith. 2005. Against Fantology, In Johann C. Marek & Maria E. Reicher (eds.), *Experience and Analysis*. Vienna: HPT: 153-170.

Yuxin Wang, Yuanning Cui, et. al. 2022. Facing Changes: Continual Entity Alignment for Growing Knowledge Graphs, In U. Sattler et al. (Eds.): *ISWC 2022, LNCS 13489*, pp. 196–213, 2022.

Rui Zhang, Bayu Distiawan Trisedya, Miao Li, Yong Jiang and Jianzhong Qi. 2022. A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning, *The VLDB Journal*, 31: 143–1168.

Beibi Zhu, Tie Bao, Ridong Han, Hai Cui, Jiayu Han, Lu Liu and Tao Peng. 2023. An effective knowledge graph entity alignment model based on multiple information, *Neural Networks* 162:83–98

Renbo Zhu, Meng Ma and Ping Wang. 2021. RAGA: Relation-aware Graph Attention Networks for Global Entity Alignment, In *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

The Universe of Utterances According to BERT

Dmitry Nikolaev **Sebastian Padó**
Institute for Natural Language Processing, University of Stuttgart
dnikolaev@fastmail.com pado@ims.uni-stuttgart.de

Abstract

It has been argued that BERT "rediscovers the traditional NLP pipeline", with lower layers extracting morphosyntactic features and higher layers creating holistic sentence-level representations. In this paper, we critically examine this assumption through a principle-component-guided analysis, extracing sets of inputs that correspond to specific activation patterns in BERT sentence representations. We find that even in higher layers, the model mostly picks up on a variegated bunch of low-level features, many related to sentence complexity, that presumably arise from its specific pre-training objectives.

1 Introduction

The Transformer architecture of neural networks (Vaswani et al., 2017) shows state-of-the-art performance on a range of NLP tasks (Wang et al., 2018, 2019). At the same time, the question of what Transformer models learn exactly has motivated a number of studies into the representations that they construct (Rogers et al., 2020; Chi et al., 2020; Papadimitriou et al., 2021), with an increasingly popular answer being that they recreate the classical NLP pipeline of incremental abstraction, from morphosyntax to semantics (Tenney et al., 2019; Geva et al., 2021).

In this paper, we put this finding to the test, asking to what extent representations learned by pre-trained BERT (Devlin et al., 2019) capture systematic meaning distinctions, as opposed to more shallow and potentially idiosyncratic properties. The challenge of this question is that it is open-ended: in order not to bias the analysis, we do not want to rely on a set of categories that we *a priori* expect to be relevant, in contrast to most probing approaches, which correlate model representations with properties of inputs or performance metrics (see Section 2 for details).

Instead, we adapt the approach that Geva et al. (2021) proposed to analyze decoder-only Transformer models with causal masking. They regard feed-forward (FF) sublayers in such models as *neural memory units* and extract inputs that produce maximal activations in a random subset of their neurons. Manual analysis of these sets shows what categorization of inputs arises inside the model.[1]

We extend the approach by Geva et al. in two dimensions. First, we apply it to pre-trained bidirectional encoder-only transformer models like BERT (Devlin et al., 2019; Liu et al., 2019), which, unlike causal LMs, do not have a specific token guaranteed to represent all of the input. To do so, we analyze two types of "prominent" tokens: the CLS pseudo-token, often used for whole-sentence representation (Ma et al. 2019; even if its usefulness for downstream tasks is debatable, cf. Reimers and Gurevych 2019), and the first subword of the `root` element in sentences annotated with Universal Dependencies (Nivre et al., 2020). We regard these two tokens as good candidates for loci of high-level, abstract representations of inputs learned by BERT.

Second, we replace the analysis of random neurons by *guided exploration*. We find that embeddings of both CLS tokens and `root` tokens at upper layers are highly intercorrelated. Therefore we propose to analyze major principal components of activation matrices, in essence tracking influential groups of highly congruent neurons.

We exploit this approach to provide an analysis of the sentence patterns that BERT attends to. We find that while lower levels of BERT are predictably more attuned to lexical effects, activations in higher levels track a wide range of idiosyncratic phenomena from various linguistic levels, from individual wordforms and bigrams to lexical classes (rare

[1] An automated approach to finding features that trigger neuronal activations was proposed by Rethmeier et al. (2020). They assign probability distribution over features to different neurons, which makes qualitative analysis impractical.

Proceedings of the 15th International Conference on Computational Semantics pages 99–105
June 21–23, 2023. ©2023 Association for Computational Linguistics

words), syntactic patterns (e.g., clauses with imperatives), and miscellaneous sentence types (recommendations, incomplete sentences, short exclamations). Overall, our results indicate that the typology of sentences according to BERT is dominated by what may be called *natural classes* (Mielke et al., 2011) – clusters of objects that are characterized by a combination of values of several features – with the embeddings showing little evidence of principled semantic properties.[2]

2 Related Work

Probing transformers Two prominent avenues of the study of Transformers in NLP are (i) probing analysis of internal representations of linguistic inputs computed by the models (e.g., Vulić et al., 2020; Pimentel et al., 2020; Belinkov, 2022) and (ii) the analysis of the attention patterns that Transformers converge on to compute these representations (Voita et al. 2019; Bian et al. 2021 and many others). Both strategies rely on predefined arrays of NLP tasks and features, which are either used as benchmarks or are selected to highlight peculiarities of models.

In contrast, Geva et al. analyze the representation of the last unmasked token of the input sequence in the causal language model by Baevski and Auli (2019), which serves as the representation of the whole prefix. By sampling neurons from feedforward sublayers of the model and manually inspecting sentences that give rise to maximum values of these neurons, they show that the latter recognise different patterns in the input – with lower-layer activations tuned to more superficial lexical and syntatctic features and upper layers arguably more tuned to semantics. We extend this approach to bidirectional LMs.

3 Methods

3.1 Models and Data

All experiments are conducted based on the `bert-base-cased` model provided by Wolf et al. (2020). We use the train and development splits of the Georgetown University Multilayer (GUM) corpus (Zeldes, 2017), which is annotated with Universal Dependencies. Together, the two splits comprise 6,507 sentences.

3.2 Analysis Procedure

Token selection and representation We consider two tokens that are promising candidates as loci for high-level categories that we would expect BERT pre-training to extract from inputs: the CLS token and the dependency root of the sentence.

For the CLS token, used in pre-training for the next-sentence-prediction (NSP) task, we concentrate on the output of the pooler layer: an additional MLP is applied to the raw BERT encoding before it is fed into the classifier head.

The `root` token does not play a special role in pre-training, but we assume that, as it largely corresponds to the head predicate of the clause, it should attend to its various syntactic elements in order to be selected correctly and to guide selection of other tokens.[3] To analyze `root` tokens, we experiment with the outputs of feedforward sublayers in the 3rd, 6th, and 11th BERT layers, which should roughly correspond to different layers of generic linguistic abstraction attained by the model. The final layer has been suspected of being too task-specific (Kovaleva et al., 2019).

Analysis procedure Our analysis proceeds in two steps for both types of tokens: (1) we gauge the extent of redundancy in the representations, given that BERT neurons are known to be highly redundant in general (Dalvi et al., 2021). As we will show in Section 4.1, CLS-token embeddings are in particular highly redundant. Consequently, (2) we identify the first 5 principal components of embedding matrices and extract sentences with maximum and minimum scores for these PCs to manually to identify shared features, similarly to Geva et al. (2021).

Hypotheses Under the theory that BERT rediscovers the traditional hierarchy of NLP tasks during pre-training (Tenney et al., 2019), we expect it to be possible to interpret principal components as bundles of linguistic features, with higher layers moving from morphosyntax towards sentence semantics. Alternatively, we can hypothesize that BERT optimizes its representations primarily for its pre-training objectives (next-sentence prediction for CLS and masked-token prediction for `root` tokens), which would presumably not support a clean interpretation in terms of a feature hierarchy.

[2] The code used for the analyses in this paper is available at https://github.com/macleginn/universe-of-utterances

[3] Another vector that is often used as a stand-in for the whole sentence is the average of all tokens embeddings. As by construction it cannot be tied to any particular sentence component, it is less interpretable.

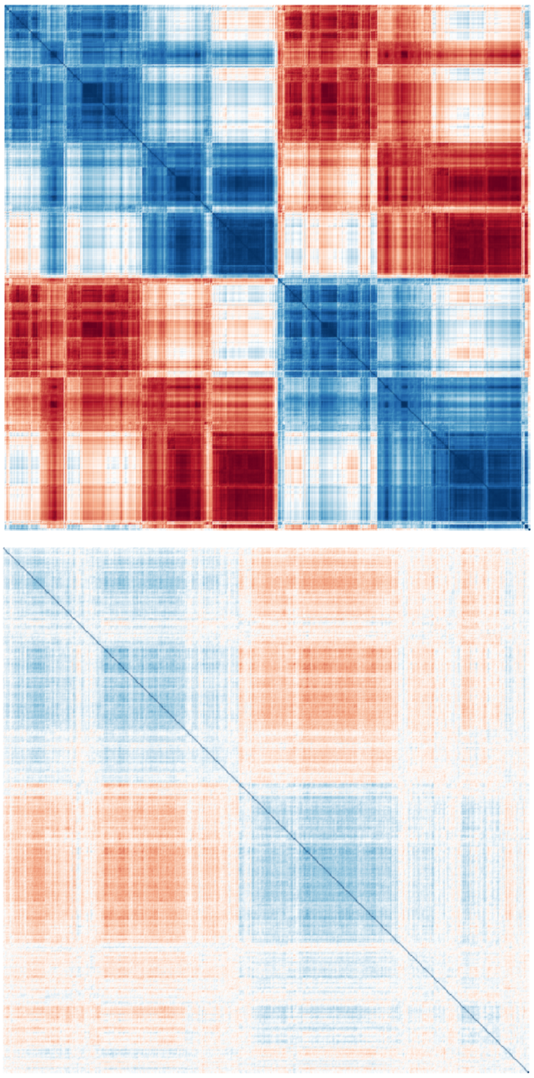


Figure 1: Correlations of neuronal activations in the output of the pooler layer for CLS tokens (top) and the output of the FF sublayer in layer 11 for `root` tokens (bottom). Rows and columns were reordered using hierarchical clustering. More intense blue/red hues correspond to stronger positive/negative correlations.

4 Results

4.1 Neural Redundancy and Major Principal Components

As motivated above, we first assess the redundancy of the pre-trained BERT embeddings for `root` tokens and CLS tokens (Dalvi et al., 2021). The results, shown in the correlation plots in Figure 1, reproduce the findings of earlier studies: there are evident clusters of mildly correlated and anticorrelated neurons in `root` token embeddings, while CLS neurons are extremely intercorrelated.

This redundancy motivates our use of principal-components analysis (PCA) to reduce the dimensionality of these representations. We find that more than 50% of the variance in the output of the pooler layer for CLS is explained by the first component alone; the first three components explain 76% of the variance. In the case of FF sublayer embeddings of `root` tokens, the first component explains 25% of the variance in the 11th layer (36% for the first 3 PCs together), but only 5% in the 6th layer (10% for the first 3 PCs), and only 3% in the 3rd layer (8% for the first 3 PCs). Taken together, this shows that as BERT progresses in its analysis of the inputs, it aggressively discards more and more information (Tishby and Zaslavsky, 2015). Upper layers are less redundant than middle layers (Dalvi et al., 2020) but might still be overparameterized (although this may also be interpreted as "spare capacity" for fine-tuning).

4.2 Principal Component-based Analysis

Sentences with maximum and minimum scores for 5 PCs for all studied settings can be found in the paper's code repository.[4]

4.2.1 CLS Tokens

Table 1 shows examples and statistics for the first 5 PCs of the CLS embedding space. The PCs may be interpreted as as largely corresponding to *sentence complexity*: they are noticeably correlated with sentence length and somewhat correlated with the number of rare words, operationalized as hapax legomena in the test set. The particular patterns, however, are highly varied.

Sentences with top scores on **PC 1** are all short, consist of bare NPs, and do not include rare words. Sentences with minimal scores are, by contrast, mostly long and include rare words, such as person and place names.

Examples with minimal values for **PC 2** are all short conversational utterances, while sentences with maximum values do not form a coherent group.

Values for **PC 3** demonstrates the highest correlation with sentences length (0.57). Examples with minimum values are all short quotes without verbs of (reported) speech. Examples with maximum values seem all to be narrative sentences with first-person-pronoun subjects.

Minimum values for **PC 4** are mostly triggered by sentences with an opening quote mark but without a closing one, i.e. those starting a direct-speech

[4] https://github.com/macleginn/universe-of-utterances

PC	SL	HL	Inputs w/ extremal values
1st	-0.26	-0.1	**[max]** Estimated electricity use in residential sector; Second baseman / Shortstop / Outfielder; High school career
2nd	0.37	0.37	**[min]** Yeah, I bet.; Yeah , that’s a good idea.; Probably.; Sure.; Nah, I’m kidding.; Yeah , "think again" or something like that.; I have no idea.
3rd	0.57	0.22	**[min]** “With what?”; “Have you thought of that?”; "Why?"; "Are you ashamed of her?"; “It’s not a joke.”; "No." **[max]** I would find myself entering those crypts...; ...I came up with an individual story called Thad’s World Destruction...; We just want to be able to bring, like she said, bring light into the entertainment...
4th	0.34	0.27	**[min]** We are a colony.; They’re going to implant a chip.; Go away!
5th	-0.06	0.1	**[min]** THE END; Chapter Two: Master Lunre; 1 Harvest and prune

Table 1: CLS token analysis: Spearman correlations with measures of complexity (SL: sentence length, HL: hapax-legomena counts per sentence) and examples inputs with extremal values (minimum / maximum).

segment. Sentences with maximum PC 4 values are not easily interpretable.

Minimum values for **PC 5** are shown by a varied set of sentences many of which are chapter/section names. Sentences with maximum values on this axis do not afford a simple interpretation.

Overall, it is evident that CLS representations are finely attuned to different kinds of sentences that are likely to appear in particular contexts and are thus informative for the next-sentence-prediction task. Their semantic properties, which CLS tokens are often assumed to be representations of, seem to be largely irrelevant.

4.2.2 `root` Tokens

Layer 3 As expected, the FF sublayer of layer 3 is focused on shallow features. Sentences with minimum values for **PC 1** are headed by the verb *have*. Minimal values for **PC 2** correspond to a combination of the verb form *said* and quote marks. Maximal values for **PC 3** track non-third-person subject and the verb *know* in the present tense, preferably in combination.[5] **PC 4**, despite being orthogonal to previous components, assigns minimal values to sentences headed with *have* and maximal values to sentences with *said* and quote marks. Similarly, **PC 5** assigns minimal values to sentences with *know* but maximal values to sentences headed by forms of *go* and *come*, including phrasal verbs with widely differing semantics (*go on*, *go through*, *come home*), which shows that this combination is more collocational than semantic.

Layer 6 We expect Layer 6 to encode more abstract features. However, **PC 1** of `root` tokens on layer 6 is highly negatively correlated with sentence length ($r = -0.62$). Examples with high scores include one-word utterances (*Alright.*), dates, and image captions of the form *Image: [AUTHOR].* Minimum values of **PC 2** correspond to sentences with forms of the verb *say* and a couple of other verbs of speech as the head predicate. Maximum values seem to be uninterpretable. Similarly, minimum values of **PC 3** correspond to sentences headed by the verb *have*, while sentences with maximum values are, with several exceptions, headed by *be*. Small values for **PC 4** are indicative of verbs of creation (*make*, *construct*, *build*). Small values of **PC 5** again correspond to sentences with the verb *have*. Sentences with high scores on this component, however, are predominantly headed by a verb in the imperative mood (*see*, *know*, *come*, *tell*, etc.).

Layer 11 We expect Layer 11 to represent high-level semantic features. But again, **PC 1** on layer 11 is also correlated with sentence length, this time positively ($r = 0.57$). This time, sentences with minimal scores have a rather specific form of technical instructions, including recipes.[6] Minimal values of **PC 2** seem to be connected to different kinds of short sentences (*You ass.*; *Absolutely great.*; *She sighs.*), incomplete phrases (*They’re really —*; *Melanie lies but —*), and nominative heading-like constructions (*Basalt columns*; *Country-specific advise*). Minimal values of **PC 4** correspond to sentences headed with *there’s*, *there is*, *it’s*, and,

[5] *Know your audience.*; *We know self-isolation works.*

[6] *Position a large mirror so you can check your positioning and see what you’re doing.*; *Add six Skittles to 25 ml of vodka.*

somewhat incongruously, *I'm*. **PCs 3** and **5** do not support an obvious interpretation.

Discussion Overall, the PCs of `root` token representations on layer 3 are oriented towards frequent verbs tokens, while layer 6 adds a morphosyntactic category of imperatives, and layer 11 singles out a wide variety of sentence patterns anchored by features at the level of surface properties (sentence length, presence of a particular verb), lexical groups (verbs of creation), syntactic categories (imperatives), or text types (technical instructions). Sentence length remains a recurring feature, as it is for the CLS token.

5 Conclusions

The good performance of Transformers on downstream tasks is often explained by their ability to extract meaningful linguistic, generalizing features from raw text (Tenney et al., 2019; Rogers et al., 2020; Geva et al., 2021). When approaching this problem from the point of view of a particular set of tasks, however, there is always the danger that good model performance is due to accidental covariates in the data that help models solve the task without creating useful generalizations (Levy et al., 2015; Gururangan et al., 2018).

Our analysis of loci in BERT that are highly likely to aggregate linguistic generalizations about the input sentence indicates that this problem might indeed be present in this model as well: we find a conspicuous absence of high-level generalizations and prominent shallow features even in the final layers, arguably because they prove useful in solving the cloze and next-sentence-prediction pre-training tasks. Many of these are complexity-related, similar to biases found in word embeddings (Wilson and Schakel, 2015).

These findings arguably go some way towards explaining the instability of the performance of different instances of BERT on the same downstream task (McCoy et al., 2020) and of the variance in the effects of BERT interventions (Sellam et al., 2021). The question of whether it is possible to create a pre-training task that would nudge the model towards extracting high-level features remains open.

Limitations

One limitation of this study is that it demands manual inspection of extracted sentences. While this makes it possible to identify patterns in a way not prejudiced by the downstream task or the available annotations of the inputs, it also makes it harder to provide quantitative arguments in favor of the proposed analysis.

Another limitation is that we only focus on maximum and minimum values of the principal components when extracting diagnostic sentences. This provides for a clear interpretation when PCs can be construed as well-defined axes; however, sometimes they appear to be "discontinuous", with different properties surfacing at the two extreme points. This suggests that there may be other interesting classes of inputs encoded by mid-range values.

References

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. 2021. On attention redundancy: A comprehensive study. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 930–945, Online. Association for Computational Linguistics.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2019. Universal text representation from BERT: An empirical study. *arXiv preprint arXiv:1910.07973*.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Jeff Mielke, Elizabeth C. Zsiga, and Paul Boersma. 2011. The nature of distinctive features and the issue of natural classes. In Abigail C Cohn, Cécile Fougeron, and Mary K M. Huffman, editors, *The Oxford Handbook of Laboratory Phonology*, pages 185–196.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Isabel Papadimitriou, Ethan A. Chi, Richard Futrell, and Kyle Mahowald. 2021. Deep subjecthood: Higher-order grammatical features in multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2522–2532, Online. Association for Computational Linguistics.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Rethmeier, Vageesh Kumar Saxena, and Isabelle Augenstein. 2020. TX-Ray: Quantifying and explaining model-knowledge transfer in (un-)supervised NLP. volume 124 of *Proceedings of Machine Learning Research*, pages 440–449, Virtual. PMLR.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. 2021. The MultiBERTs: BERT reproductions for robustness analysis. In *Proceedings of ICLR 2022*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Benjamin J. Wilson and Adriaan M. J. Schakel. 2015. Controlled experiments for word embeddings. *CoRR*, abs/1510.02675.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Sparser is better: one step closer to word embedding interpretability

Simon Guillot[1,2] **Thibault Prouteau**[1] **Nicolas Dugué**[1]
Le Mans Université, LIUM[1]
INaLCO, ERTIM[2]
firstname.lastname@univ-lemans.fr

Abstract

Sparse word embeddings models (SPINE, SINr) are designed to embed words in interpretable dimensions. An interpretable dimension is such that a human can interpret the semantic (or syntactic) relations between words active for a dimension. These models are useful for critical downstream tasks in natural language processing (*e.g.* medical or legal NLP), and digital humanities applications. This work extends interpretability at the vector level with a more manageable number of activated dimensions following recommendations from psycholinguistics. Subsequently, one of the key criteria to an interpretable model is sparsity: in order to be interpretable, not every word should be represented by all the features of the model, especially if humans have to interpret these features and their relations. This raises one question: to which extent is sparsity sustainable with regard to performance? We thus introduce a sparsification procedure to evaluate its impact on two interpretable methods (SPINE and SINr) to tend towards sustainable vector interpretability. We also introduce stability as a new criterion to interpretability. Our stability evaluations show little albeit non-zero variation for SPINE and SINr embeddings. We then show that increasing sparsity does not necessarily interfere with performance. These results are encouraging and pave the way towards intrinsically interpretable word vectors.

1 Introduction

Word embeddings models (Mikolov et al., 2013; Pennington et al., 2014; Devlin et al., 2018) allowed tremendous evolution in natural language processing. However, they embed the lexicon in dense representation spaces with opaque dimensions. It is possible to obtain an understanding of these models via probing (Rogers et al., 2021) and embedding matrix analysis (Shin et al., 2018). However such methods are subject to criticism with regard to the interpretation that can actually be drawn from them (Hewitt and Liang, 2019; Ravichander et al., 2021; Elazar et al., 2021). This *a posteriori* approach to understanding models' decisions corresponds to the explainability paradigm in machine learning.

On the other hand, interpretability (Rudin, 2019) is defined for word embedding models as the possibility to find semantic (or syntactic) consistency in the dimensions of the embedding space (Murphy et al., 2012; Faruqui et al., 2015; Subramanian et al., 2018; Prouteau et al., 2022). Models such as SPINE (Subramanian et al., 2018) and SINr (Prouteau et al., 2021) meet this requirement: Table 1 illustrates the interpretability of the dimensions resulting from such methods. These inherently interpretable approaches to represent the lexicon are deemed preferable for high-stakes downstream use such as medical or legal NLP (Rudin, 2019). Interpretability also eases connection between word embeddings and linguistic models of the lexicon, since consistent semantic dimensions can be grasped as semantic features, which are used in a variety of theoretical models (Jackendoff, 1983; Pottier, 1963; Rastier, 2009).

As far as we know, only the interpretability of dimensions is considered in the literature and human evaluations such as the *Word Intrusion Detection* (Murphy et al., 2012) are targeted specifically towards this aspect. In this paper, we introduce **vector-level interpretability** and define it as the capacity for a speaker to make sense of the set of activated dimensions in a word vector. It is possible only if the set of dimensions to describe the word is limited. The size of this set is bounded by two different kinds of psychological experiments: semantic features production (Garrard et al., 2001; McRae et al., 2005) and features retention (Miller, 1956; Peterson and Peterson, 1959). This body of literature comes to an agreement at roughly ten fea-

Proceedings of the 15th International Conference on Computational Semantics pages 106–115
June 21–23, 2023. ©2023 Association for Computational Linguistics

tures. We consider in this paper that this number of features is a desirable horizon for vector-level interpretability. Following this objective and to further reduce the amount of information provided to the speaker, we also consider binary word vectors as in Faruqui et al. (2015). Moreover, this binary approach is consistent with componential analysis (Goodenough, 1956; Katz and Fodor, 1963).

Considering these criteria, and to tend towards more interpretability, our work offers the following contributions :

- Refine interpretability by introducing additional criteria: stability and increased sparseness for vector-level interpretability.
- Evaluate the effects of increased word vector sparseness and binarity on performance.
- Illustrate the effects of increasing vector sparseness on the embedding space.

To this end, we introduce Section 2 the criteria for interpretability and their different settings in the literature. Section 3 introduces the models considered for our experiments. In Section 4, we detail the experimental setup adopted to evaluate the impact of sparsity as well as binarity on performance and vector-level interpretability. In Section 5, we demonstrate that the trade-off between sparsity and interpretability is not as strong as one would think. Finally, Section 6 illustrates the impact of sparsity on word vectors and discusses its benefits.

2 Related work

Interpretability : criteria and models. The seminal article of (Murphy et al., 2012) paves the way towards psycholinguistically plausible distributional representations. The authors fix the following set of constraints on the representation space: sparseness, positivity and performance. Sparseness is justified by the difficulty to cover a vast vocabulary comprised of many different topics with a small set of features. Thus, a large number of dimensions is needed, but only some of those are activated for the description of each word. Positivity is motivated by the fact that storing null or negative features for each item of the lexicon is not cognitively efficient (Palmer, 1977; Lee and Seung, 1999). The performance criterion is needed since it is possible to produce interpretable representations of the lexicon (*e.g* raw co-occurrence matrices) with subpar performances on intrinsic or extrinsic evaluations. This sparse interpretable word model research is carried on with SPOWV (Faruqui et al., 2015), SPINE (Subramanian et al., 2018) and SINr (Prouteau et al., 2021). The first two models transform previously trained dense representations into sparse word embeddings while the latter builds a sparse embedding space from a word co-occurrence matrix. The word intrusion tests (Murphy et al., 2012; Senel et al., 2018; Subramanian et al., 2018; Prouteau et al., 2022) are designed to assess the internal consistency of dimensions in the embedding space. As introduced Section 1, we wish to allow interpretability at the vector level which might benefit from a smaller set of activated components in word vectors.

Stability. Pierrejean (2020) demonstrate the non-determinism of neural models' training which lead to variations in evaluation scores and word neighborhoods. On the front of explicability, new deterministic methods are emerging (Zafar and Khan, 2021). However, Rudin (2019) encourages to prioritise interpretable approaches over explicable approaches, motivating this work.

From these observations and as stated Section 1, we refine the criteria necessary to enable vector-level interpretability by redefining sparsity and adding stability.

Binary embeddings. Prototypicality theory (Rosch, 1975; Rosch et al., 1976) introduced the paradigm of weighted features in psychology and linguistics. However, feature-based analysis preempted this theoretical framework with componential analysis. This approach based on binary features was used by anthropological linguists (Goodenough, 1956), in structuralist work (Pottier, 1963) and in cognitively informed generativist frameworks (Katz and Fodor, 1963). Faruqui et al. (2015) construct binary vectors using sparse coding to sparsify dense word embeddings in more dimensions than the original space—called overcomplete vectors (SPOWV). The model is then binarized simply by setting each non-zero value to one. In computer science, another use to binary models is to reduce the memory footprint of word embeddings by replacing floats with bits and also the compute needed to exploit these representations. It is especially critical in low-resource embedded systems—*e.g* mobile phones. Tissier et al. (2019) and Navali et al. (2020) introduce autoencoder approaches to binarize

	Word2Vec	SPINE	SINr
insulin	scalar, tablespoon, vesicular, dystrophy antiserum, falsifiable, experimenter, internat PBS, NC, arginine, IFN	glutathione, pancreas, gastroduodenal, vitamin immunologically, hyperplasia, transgene, nociceptive insulin, sulphasalazine, interferon, cholangitis	hypertriglyceridaemia, mellitus, porcine, insulin aldosterone, aminotransferase, creatinine, glycated ulcerative, sulphasalazine, colitis, sera
mint	scalar, tablespoon, vesicular, dystrophy cube, geranium, Berowne, curiosities polyunsaturated, misfire, margarine, methile	spoonfuls, parsnips, kebabs, preheat onion, basil, yogurt, coriander dial, screams, vibration, spadefoot	tbsp, oregano, diced, dijon Gibson, gigged, charvel, Ibanez minted, minting, hoards, coinages
oxygen	scalar, tablespoon, vesicular, dystrophy herbicides, menstrual, deprave, angiotensin pou, tenascin, cytoplasm, platelet	glutathione, pancreas, gastroduodenal, vitamin lipid, crypt, tris, calcium monoxide, oxides, sulphuric, nitrogen	monoxide, dioxide, nitrous, oxides supplemental, hypoxaemic, electrocardiographic, gastroscopy diastolic, systolic, transfusion, transfusions

Table 1: Words with the highest values on the top three dimensions of ”*insulin*”, ”*mint*” and ”*oxygen*” in Word2Vec, SPINE and SINr sparsified to 100 active dimensions per vector according to the protocol described Section 4.

dense representations. Both of these models optimize for non-redundancy among dimensions and conservation of semantic information. Once vectors are binary, classical evaluation tasks such as word similarity or analogy may be redefined with bitwise operations (Sokal and Michener, 1958; Tissier et al., 2019). These models achieve competitive results to the baseline considering their small footprint.

3 Interpretable word embeddings

SPINE and SPOWV achieve close results on intrinsic and downstream evaluations but SPINE scores better in terms of interpretability (Subramanian et al., 2018), we thus do not consider SPOWV in the experiments that follow. Furthermore, SINr performances and interpretability are on a par with SPINE, we thus consider both SPINE and SINr as our reference interpretable models.

SPINE. SPINE, first introduced in Subramanian et al. (2018) derives sparse word embeddings from a previously trained dense model such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Architecturally, it is an autoencoder whose hidden layer is of higher dimension than the dense input—*e.g* sparsifying from 300 dense dimensions to 1000 sparse dimensions. Three losses are implemented to enforce sparsity and interpretability. The *Reconstruction Loss* penalizes the poor reconstruction of the input representation from the output of the hidden layer, the *Average Sparsity Loss* and the *Partial Sparsity Loss* enforce sparse representations by limiting the number of active dimensions and skew vector values towards 0 or 1. SPINE has multiple hyperparameters: the minimum sparseness, the number of epochs and the vector output dimension.

SINr. Introduced in Prouteau et al. (2021), SINr is a graph-based approach to word embeddings. From a co-occurrence matrix extracted on a corpus, SINr builds a weighted word co-occurrence graph—words are represented by nodes and the number of co-occurrences by edges. A community detection algorithm, the *Louvain* method (Blondel et al., 2008), then uncovers dense groups of co-occurring words in the graph. SINr then leverages the distribution of each node over this partition to derive a sparse representation—not all words co-occur with words from each community. The representation is sparse by design, each component of the embedding space is related to a community. Community detection is an unsupervised process admitting a single parameter allowing to potentially control the number of communities detected.

4 Methodology

Models. Alongside the models presented Section 3, Word2Vec is used as a baseline. We use the *Skip-gram with negative sampling* (SGNS) architecture and the parameters described in Levy and Goldberg (2014). Word2Vec embeddings have 300 dimensions with a context window of 5 words. Since SPINE's number of dimensions is adjustable when SINr's is not—it is dependent on the number of communities detected—we base the number of dimensions of SPINE on SINr. Optimal performances for SINr are observed with the hyperparameter controlling the number of communities set to 50 resulting in 4460 dimensions for OANC (Nancy et al., 2011) and 8454 for BNC (Consortium, 2007) —the English corpora we use in our experiments is presented at the end of the next section. SPINE embeddings are trained from the Word2Vec model previously presented. The sparsity parameter of SPINE has little impact on the sparsity of the output. Subsequently, after several rounds of training, the model selected is that which achieves the best performances on the similarity task with a sparseness—95% after 1000 epochs—allowing further sparsification according to our experimental setup described hereinafter.

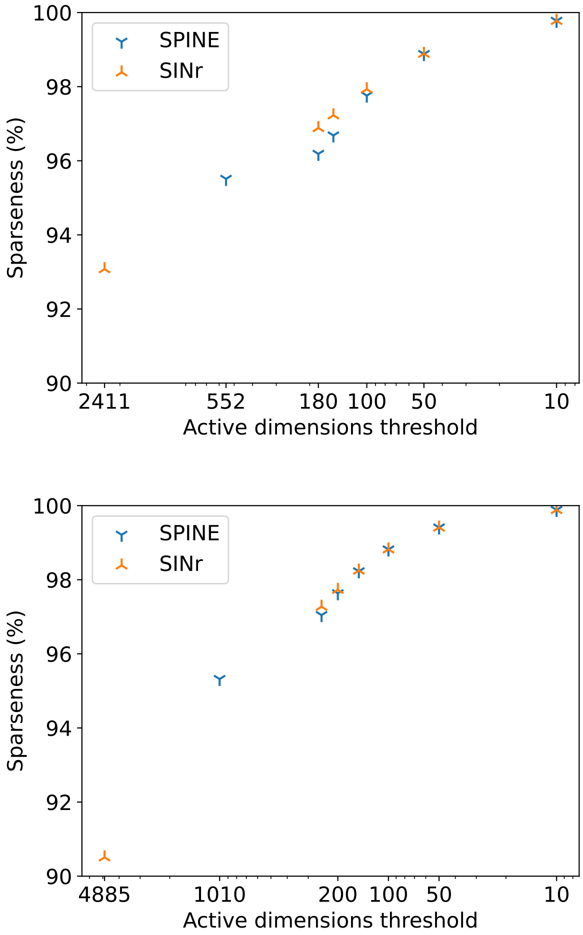


Figure 1: Sparseness of SPINE and SINr according to the maximum number of activated dimensions per vector on OANC (top) and BNC (bottom). First data point of each model is sparseness before sparsification.

Experimental framework. We introduce an experimental framework allowing to evaluate word embedding interpretability. We first consider a performance-sparsity compromise. Our hypothesis is that sparse vectors are both more interpretable and psycholinguistically plausible. To control sparseness, we introduce our sparsification method: from each embedding model, we keep only the k top strongest dimensions by value in each vector—k is in range $250 - 10$. Components not in the top k for the vector are set to zero. Figure 1 presents the sparseness of SPINE and SINr with regard to the active dimensions threshold. In the case of Word2Vec, we keep the top k dimensions out of the absolute values from the vectors.

In our second setup, we study the impact of switching to binary vectors. The binarization step is straightforward, we simply replace all non-zero values in each sparsified and unsparsified model by 1 as in Faruqui et al. (2015).

To evaluate the quality of the representations after sparsification and binarization, we use the word similarity evaluation—the correlation between the cosine similarity of words in our model and similarity rated by humans. Selected datasets model a variety of relations : MEN (Bruni et al., 2014), WS353 (Agirre et al., 2009), SCWS (Huang et al., 2012). To evaluate the stability of vectors produced by SPINE and SINr, our second criterion to interpretability, we learn 10 models and present the averaged results.

As similarity datasets are mostly available in English, we use the *British National Corpus* (BNC) (Consortium, 2007) and the text part of the *Open American National Corpus* (OANC) (Nancy et al.,

BNC	MEN		WS353		SCWS	
	$\overline{Pearson}$	σ	$\overline{Pearson}$	σ	$\overline{Pearson}$	σ
Word2Vec	0, 72	0, 002	0, 65	0, 005	0, 57	0, 002
SPINE	0, 65	0, 006	0, 57	0, 01	0, 60	0, 004
SINr	0, 66	0, 0006	0, 62	0, 002	0, 54	0, 001
OANC	MEN		WS353		SCWS	
	$\overline{Pearson}$	σ	$\overline{Pearson}$	σ	$\overline{Pearson}$	σ
Word2Vec	0, 43	0, 002	0, 50	0, 005	0, 46	0, 003
SPINE	0, 36	0, 009	0, 43	0, 01	0, 39	0, 01
SINr	0, 39	0, 0008	0, 44	0, 002	0, 39	0, 002

Table 2: Stability results for the word similarity evaluation on BNC (top), and OANC (bottom). Average Pearson correlation coefficient and standard deviation σ over 10 runs.

2011) to train our models. BNC contains 100 million tokens and OANC 11 million. Both corpus are composite in domain and genres. Those relatively small corpora, considering the standards in natural language processing, are chosen because documented corpora allow for finer interpretations of dimensions. Text preprocessing was performed using spaCy : tokenization with named-entity chunking, deletion of words shorter than three characters, of punctuation and of numerical characters. The minimum frequency for a type is set to 20. After preprocessing, OANC contains 20,814 types and roughly 4 million tokens, 58,687 types and 40 million tokens for BNC.

5 Results

Stability. The first property we consider with regards to interpretability is the stability of the models trained. This experiment is twofold, it allows to show whether methods are stable and also sets reference values for the similarity evaluation prior to sparsifying. Each model was run ten times on the same data with the same hyperparameters.

As reported in Table 2, the three models achieve scores in close ranges, with all models showing some degree if variability, their standard deviation being non-zero across ten runs. While Word2Vec and SINr seem more stable than SPINE, the overall observed variability on the small samples of the vocabulary present in the similarity datasets hinders reproducibility and is a flaw to the three model's interpretability.

Impact of sparsity on similarity. Results presented Figure 3 show the Pearson correlation scores on the similarity evaluation with regard to the number of components activated. The similarity scores are given with regard to the maximum number of top values kept in each vector according to our sparsification procedure. First, the three models achieve comparable results to those reported Table 2 up until 50 dimensions. More surprisingly, sparsifying SINr embeddings seems to improve performances. Sparsification may filter out noise from the base SINr model. Subsequently, there is not necessarily a trade-off between sparseness and efficiency. Furthermore, the fact that results remain satisfactory on our Word2Vec control model despite the sparsification is an unexpected behavior and is interesting with regard to how the semantic information is organized in its vectors.

In order to approach the sparsity objective of 10 dimensions presented Section 1, the experiment is also conducted at this level. Although we observe an overall drop in performance and especially for Word2Vec, a significant part of the semantic information is retained within these ten dimensions. Indeed, they allow to solve at least partially the similarity task. Even though the usefulness of this representation for downstream tasks can be discussed, it still allows to build interpretable word vectors despite the drop in performance. The low number of active dimensions render these models compatible with theoretical models leveraging semantic features, thus paving the way for new empirical opportunities.

Impact of binarization on similarity. Results presented Figure 3 follow the same display than sparsity results except that all models are binarized. Overall, we observe drops in performance across all models but to drastically varying extents. While SPINE and SINr lose some semantic information compared to the sparsified weighted models, they tend to retain performances of the same magnitude. This is especially true for models trained on BNC, considering that the models trained on OANC show bigger drops in word similarity performance. On the other hand, overall Word2Vec performances crumble with binarized vectors. This result is to be expected since Word2Vec is a dense model.

We can observe a common pattern across all models, where performance of binarized embeddings increases with sparsification until 100 or 50 activated dimensions. Binarizing while maintaining a lot of active dimensions flattens the hierarchy between components with strong values and others with low activations, thus otherwise very weak activations may gain weight in the vector as a result of binarization. In this case, the sparsification may remove noise from representations, by restoring a hierarchy between the few strong dimensions, activated with a 1 value, and the others set to 0. This denoising behavior resulting from sparsification seems common to binarized models, and weighted SINr.

6 Discussion

Our results show that there is not necessarily a trade-off between interpretability and performance. On the contrary, stability and increased sparseness of interpretable models can even improve results. At thresholds close to what is described in psycholinguistics, performances may remain accept-

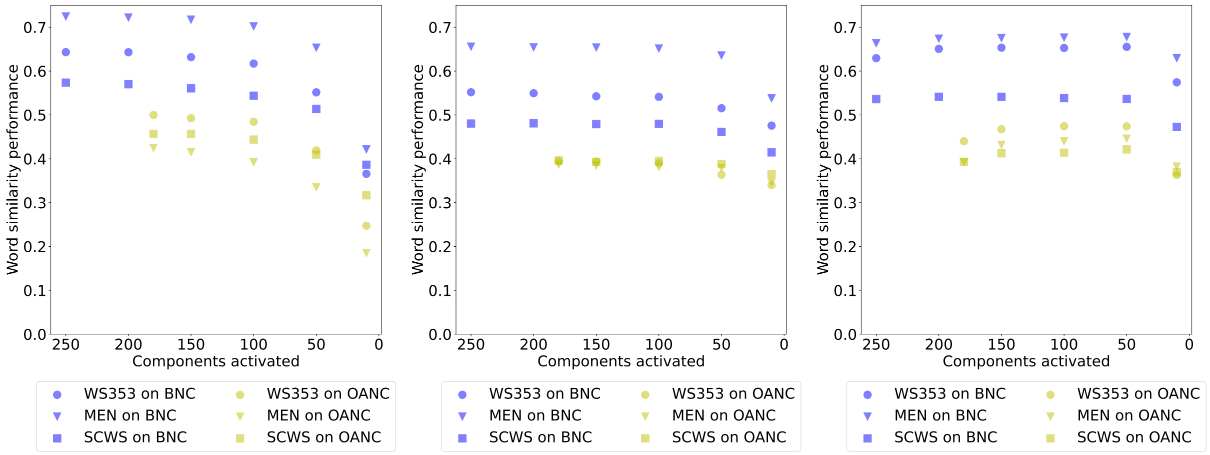


Figure 2: Word similarity performance (Pearson correlation) against maximum number of activated dimensions per vector for Word2Vec (left), SPINE (middle) and SINr (right). Performances on OANC are reported in yellow, and performances on BNC in blue.

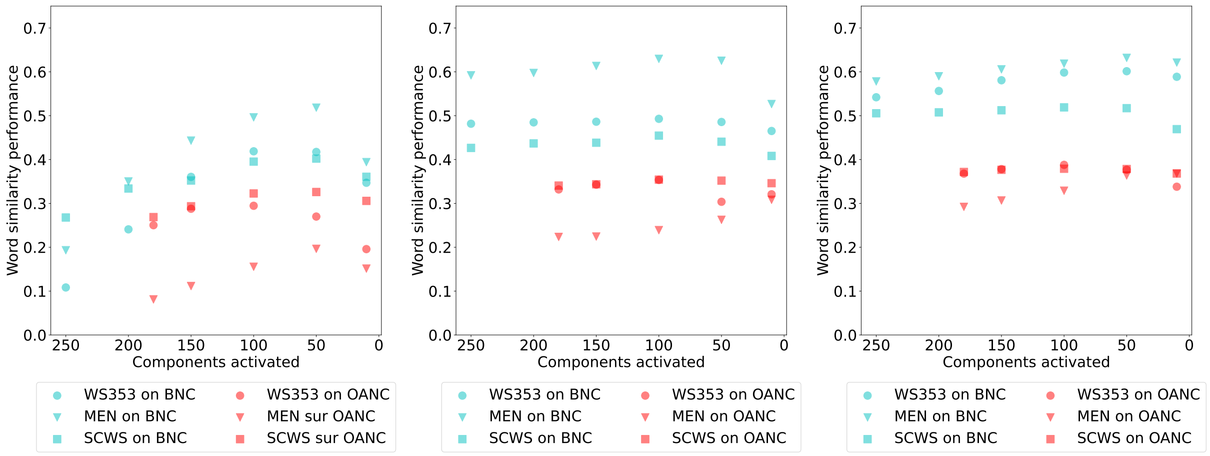
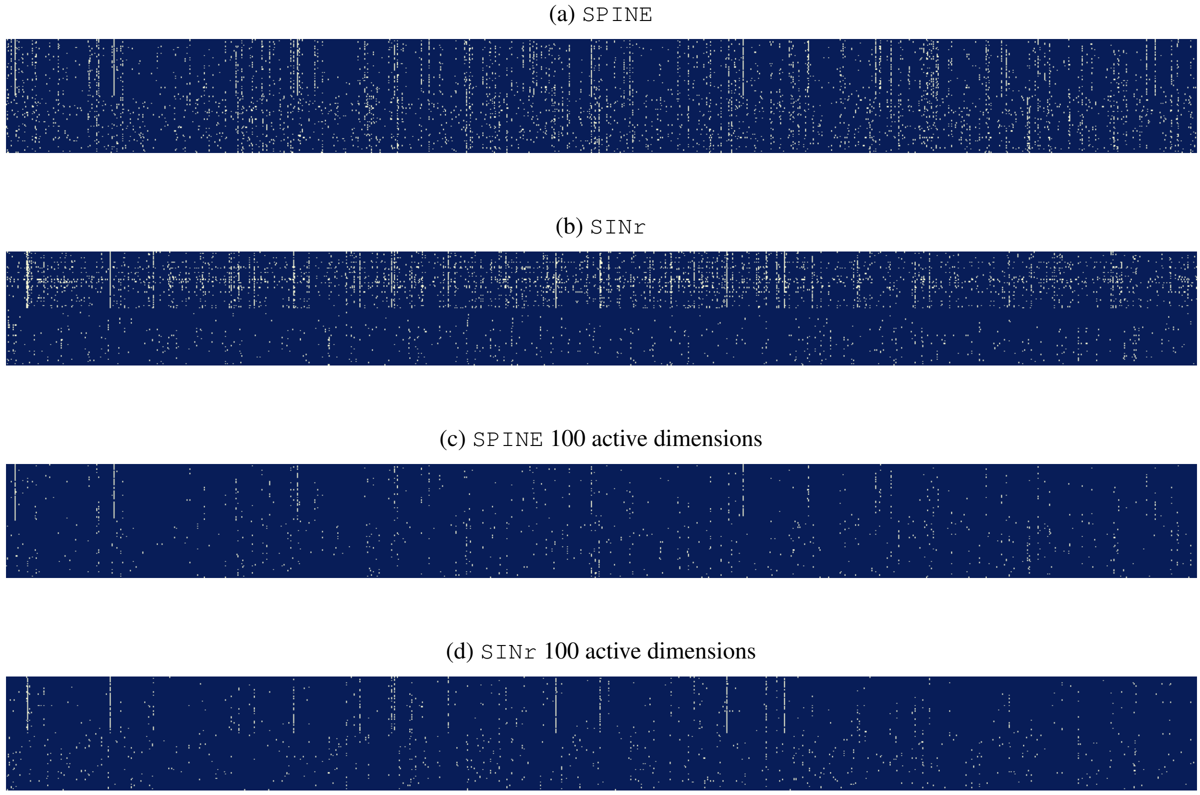


Figure 3: Word similarity performance (Pearson correlation) on binary models against maximum number of activated dimensions per vector for Word2Vec (left), SPINE (middle) and SINr (right). Performances on OANC are reported in magenta, and performances on BNC in cyan.

able considering the number of dimensions activated. Interpretability is hard to visualize without a set objective. In the discussion ensuing, we illustrate the interpretability of models through visualizations on selected items.

Interpretability of the dimension. Interpretability of the dimensions can be assessed after conducting a *word intrusion* evaluation with humans, both SPINE and SINr's dimension interpretability have been previously evaluated without prior sparsification (Subramanian et al., 2018; Prouteau et al., 2022). The goal is to evaluate whether dimensions are interpretable—words with highest values on a dimension should be related. We present Table 1 top dimensions for three words as a glimpse into how interpretable dimensions of SPINE and SINr are in comparison with Word2Vec. As in previous evaluations, Word2Vec does not exhibit dimensions with related terms. If we consider the term ”*insulin*”, words on the first three strongest dimensions in the vectors are all related to medical conditions or biological functions. The word ”*mint*” presents interesting dimensions, for SPINE, the first two dimensions are related to food and ingredients, the third one is less interpretable as one has trouble linking ”*spadefoot*”, a frog specie to ”*dial*”. SINr captures the polysemous nature of the word ”*mint*” with top dimensions unrelated with one another. The first one is most probably related to mint as an aromatic, meanwhile, the sec-



(a) SPINE

(b) SINr

(c) SPINE 100 active dimensions

(d) SINr 100 active dimensions

Figure 4: Shared dimensions across 50 most and least similar words to "*mint*" in SPINE and SINr. The models are trained on BNC both without sparsification, and with a threshold set to 100 dimensions on BNC. The top half of each figure represents the most similar words and the bottom half the least similar words.

ond one as the adjective describing guitars in mint condition, and the third one as a verb, to mint, in the sense of producing and managing currency. The same analysis can be drawn for the word "*oxygen*" where the use in the medical field is represented alongside chemical characteristics.

Interpretability of the vector. We evaluated increasingly sparsified word embeddings with the hypothesis that fewer features makes interpreting words vectors themselves easier. Our evaluations show that this gain in interpretability is not necessarily at the cost of model performances, the sparsification of representation can even increase performances up to a certain sparseness level. The following paragraphs aim to illustrate interpretability at the word vector level.

We present Figure 4 the distribution of values in the 50 most (top of each figure) and least similar (bottom of each figure) words to "*mint*" for SPINE (a; c) and SINr (b; d) on BNC. Lines appearing vertically across figures show shared dimensions between vectors in the embedding space. The first two figures (a; b) represent the shared features in the model prior to sparsification. SPINE presents vertical lines spanning most similar and least similar vectors, the embeddings seemingly share a large number of dimensions. SINr, on the other hand, exhibits a clear distinction between most and least similar words. One can clearly see shared dimensions among close neighbors of SINr for the word "*mint*". These first two distributions need to be compared with the distributions observed after sparsifying the vectors (c; d). At the 100 active dimensions sparsity setup, SINr seems to display more shared dimensions than SPINE for the word "*mint*". We assume that the performance gain in the similarity task observed for SINr Figure 2 is due to a process of noise reduction induced by the sparsification of the model.

The interesting results on similarity evaluation showed by sparsified interpretable models seems to indicate that the most important part of the semantic information is stored in the few strongest components of each vector. This observation allows us to analyze these models through the lens of our constrained version of interpretability di-

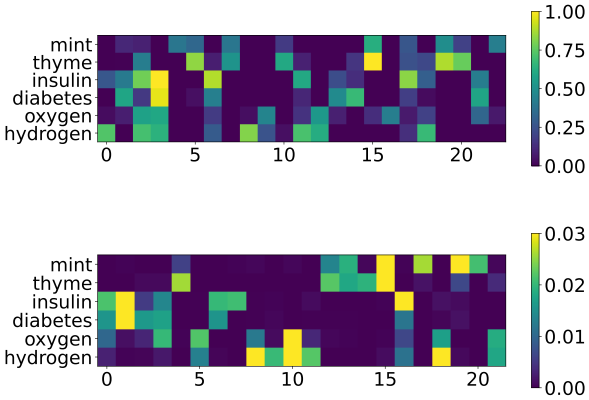


Figure 5: Word vectors on the set of top 5 shared dimensions for ”*mint*”, ”*insulin*” and ”*oxygen*” and their respective closest neighbors for SPINE (top) and SINr (bottom) on BNC.

rected towards the interpretation of word vectors. A speaker might want to interpret word embeddings by composing the meaning of a word with a limited subset of the features that describe it. In this case, the stability of the models becomes an increasingly important issue. Indeed, interpreting dimensions amounts to finding a consistency to a set of words that strongly interpret a dimension. However, interpreting a word vector relies both on this consistency and the strength of the activation of each dimension for a given vector. Thereby, even subtle variations in the representation across runs may induce different interpretations.

Binary representations. Our last experiment aims to quantify the benefit of weighted features over binary features. Considering results Figure 4, it appears that a significant part of the semantic information for sparse interpretable models is encoded in the mere activation of a dimension by a vector. Binarity is a means of reducing time and memory complexity of semantic models and is undoubtedly beneficial in embedded applications with low latency requirements or low resource hardware. We observe with Figure 5 that a SINr weighted model tends to have fewer and more strongly activated dimensions than a SPINE weighted model, which makes the former more alike binarized representations. This property facilitates the interpretation at the vector level: for example, dimensions 12 to 15 are strongly activated for ”*mint*” and ”*thyme*”, and not at all for the other words, in the SINr representation. Recognizing the similarity of ”*mint*” and ”*thyme*”, and their opposition to the other words, is easier when there is a clear gap between a strong activation and no activation of the dimension considered, like in a binarized vector.

Taking a step back, the comparison between weighted and binarized vectors performances allow us to pinpoint where the information is encoded. A significant part of the semantic information is stored in the activation of a few dimensions for each word vector, but the dimensions weights are needed to reach the most competitive performances. This assessment is coherent with the theoretical paradigm shift mentioned Section 2. Furthermore, it appears that, while binarizing embeddings represents a cost in performance, sparsifying them is not necessarily a trade-off. In some cases, it might even be beneficial.

7 Conclusion

Previously, the interpretability of embedding spaces focused mainly on dimension, this work redefined interpretability from the vector standpoint. We state that stability of the models and sparsity are necessary conditions to intepretability. Constraining on sparsity echoes psycholinguistic plausibility, it is essential to find semantic coherence within dimension of the embedding space but also to describe a word with a limited set of these dimensions. We hypothesize that vectors constrained following this protocol are interpretable by a speaker, since it becomes possible to manipulate this small number of dimensions in working memory.

Interpretable word embedding models achieve good results on the intrinsic word similarity evaluation task even with higher sparseness levels. SINr even benefits from being sparsified. Furthermore, we show through examples that dimensions remain interpretable even on sparsified vectors and that, indeed words that are close in the embedding space are represented by a common set of dimensions. Lastly, we show that real-valued vectors are a slight improvement upon binary representation.

These results allow to reconsider the interpretability performance for distributed representations. A following step would be to conceive an evaluation framework to measure vector-level interpretability, allowing us to investigate if and how speakers would make sense of interpretable word vectors. Such models also open up new perspectives in which theoretical models describing the lexicon benefit from semantic features of word embeddings. In the field of semantic drift detection, it would also allow to easily characterize the drift by keeping track of the few dimensions at stake.

Acknowledgments

The work has been funded by the ANR project DIGING (ANR-21-CE23-0010).

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Vincent D. Blondel, Jean Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47.

BNC Consortium. 2007. British national corpus, XML edition. Oxford Text Archive.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Association for Computational Linguistics*, 9:160–175.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. 2015. Sparse overcomplete word vector representations. *arXiv preprint arXiv:1506.02004*.

P. Garrard, M. A. Lambon Ralph, J. R. Hodges, and K. Patterson. 2001. Prototypicality, distinctiveness, and intercorrelation: Analyses of the semantic attributes of living and nonliving concepts. *Cognitive Neuropsychology*, 18(2):125–174.

Ward H. Goodenough. 1956. Componential analysis and the study of meaning. *Language*, 32(1):195.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Empirical Methods in Natural Language Processing*, page 2733–2743.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Ray Jackendoff. 1983. *Semantic and Cognition*. MIT Press.

Jerrold J. Katz and Jerry A. Fodor. 1963. The structure of a semantic theory. *Language*, 39(2):170–210.

Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 2, page 2177–2185.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris Mcnorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Conference on Computational Linguistics*, pages 1933–1950.

Ide Nancy, Reppen Randi, and Suderman Keith. 2011. The open anc (oanc). ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Samarth Navali, Praneet Sherki, Ramesh Inturi, and Vanraj Vala. 2020. Word Embedding Binarization with Semantic Information Preservation. In *International Conference on Computational Linguistics*, pages 1256–1265.

Stephen E. Palmer. 1977. Hierarchical structure in perceptual representation. *Cognitive Psychology*, 9(4):441–474.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543.

Lloyd Peterson and Margaret Jean Peterson. 1959. Short-term retention of individual verbal items. *Journal of Experimental Psychology*, 58(3):193.

Bénédicte Pierrejean. 2020. *Qualitative Evaluation of Word Embeddings: Investigating the Instability in Neural-Based Models*. Ph.D. thesis, Université Toulouse 2 - Jean Jaurès.

Bernard Pottier. 1963. *Recherches sur l'analyse sémantique en linguistique et en traduction mécanique*. Publications linguistiques de la Faculté des lettres et sciences humaines de Nancy.

Thibault Prouteau, Victor Connes, Nicolas Dugué, Anthony Perez, Jean-Charles Lamirel, Nathalie Camelin, and Sylvain Meignier. 2021. SINr: Fast Computing of Sparse Interpretable Node Representations is not a Sin! In *Intelligent Data Analysis*, 12695, pages 325–337.

Thibault Prouteau, Nicolas Dugué, Nathalie Camelin, and Sylvain Meignier. 2022. Are embedding spaces interpretable? results of an intrusion detection evaluation on a large french corpus. In *Language Ressources and Evaluation Conference*.

François Rastier. 2009. Principes et conditions de la sémantique componentielle. In *Sémantique interprétative*, Formes sémiotiques, pages 17–37. Presses Universitaires de France.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *European Chapter of the Association for Computational Linguistics*, pages 3363–3377.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Eleanor Rosch. 1975. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104:192–233.

Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Braem. 1976. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yucesoy, Aykut Ko.c, and Tolga Cukur. 2018. Semantic structure and interpretability of word embeddings. *Transactions on Audio, Speech, and Language Processing*, 26(10):1769–1779.

Jamin Shin, Andrea Madotto, and Pascale Fung. 2018. Interpreting word embeddings with eigenvector analysis. *Advances in Neural Information Processing Systems*, 32.

Robert R. Sokal and Charles Duncan Michener. 1958. A statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2018. Spine: Sparse interpretable neural embeddings. In *AAAI conference on artificial intelligence*, volume 32.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2019. Near-lossless Binarization of Word Embeddings. *AAAI Conference on Artificial Intelligence*, 33(01):7104–7111.

Muhammad Rehman Zafar and Naimul Khan. 2021. Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 3(3):525–541.

Semantically Informed Data Augmentation for Unscoped Episodic Logical Forms

Mandar Juvekar[*]
Boston University
Boston, MA, USA 02215
mandarj@bu.edu

Gene Louis Kim
University of South Florida
Tampa, FL, USA 33620
genekim@usf.edu

Lenhart Schubert
University of Rochester
Rochester, NY, USA 14627
schubert@cs.rochester.edu

Abstract

Unscoped Logical Form (ULF) of Episodic Logic is a meaning representation format that captures the overall semantic type structure of natural language while leaving certain finer details, such as word sense and quantifier scope, underspecified for ease of parsing and annotation. While a learned parser exists to convert English to ULF, its performance is severely limited by the lack of a large dataset to train the system. We present a ULF dataset augmentation method that samples type-coherent ULF expressions using the ULF semantic type system and filters out samples corresponding to implausible English sentences using a pretrained language model. Our data augmentation method is configurable with parameters that trade off between plausibility of samples with sample novelty and augmentation size. We find that the best configuration of this augmentation method substantially improves parser performance beyond using the existing unaugmented dataset.[1]

1 Introduction

Kim and Schubert (2019) introduced Unscoped Episodic Logical Form (ULF) as a semantic representation that captures syntactic type structure within the Episodic Logic formalism, while staying close to the surface form for ease of annotation and parsing. Kim et al. (2021a) presented a learned approach to parsing English sentences to ULF which showed promising results. Their parsing efforts, however, were limited by the size of the training data available. They released a dataset of 1,738 sentences with corresponding manual ULF annotations alongside their parser which—to the best of our knowledge—remains the only dataset of ULF annotations to date. Our work aims to alleviate this limitation of data sparsity.

```
(|Mary| ((past place.v)
         |Glenn|
         (under.p (k anesthesia.n))))
```

Figure 1: An example ULF for the sentence "*Mary placed Glenn under anesthesia.*"

In this paper, we present a method of augmenting ULF datasets. Our method leverages ULF's underlying type structure and works by replacing subtrees of seed ULFs with other subtrees of the same semantic type. This, combined with the use of pretrained language models to prune out the most incoherent sentences, allows us to expand relatively small datasets of ULF, such as that of Kim et al. (2021a), into datasets several times larger in size. We evaluate the efficacy of our system by looking at the performance of the existing ULF parser when trained on augmented versions of the original training set.

The importance of our work, and more generally of ULF parsing, comes from the role of ULF in the broader Episodic Logic (EL) framework. Episodic Logic (EL) is an extended first-order logic designed to closely match the form and expressivity of natural language (Schubert, 2000). EL is a powerful representation with rich model-theoretic semantics which enable a variety of inferences including deductive inference, uncertain inference, and natural logic-like inference (Morbini and Schubert, 2009; Schubert and Hwang, 2000; Schubert, 2014). However, parsing ordinary English sentences into fully resolved EL forms is a difficult task.

ULF is an underspecified form of EL designed to balance encoding adequate semantic information with ease of parsing. It fully specifies the semantic type structure of EL by marking the types of the atoms and of all the predicate-argument relationships while leaving issues such as quantifier scope, word sense, and anaphora unresolved. ULF is the critical first step in parsing full-fledged EL formu-

[*] Work done in part while at the University of Rochester.

[1] The code is available at https://github.com/genelkim/subtree-sampled-ulf-data-augmentation.

Proceedings of the 15th International Conference on Computational Semantics pages 116–133
June 21–23, 2023. ©2023 Association for Computational Linguistics

las. A detailed description of how ULF fits into the EL interpretation process is given by Kim and Schubert (2019). ULF is also a useful interpretation in its own right. It is capable of generating inferences based on clause-taking verbs, counterfactuals, questions, requests, and polarity (Kim et al., 2019, 2021b,c), and has been an effective representation in schema-based story understanding (Lawley et al., 2019) and spatial reasoning-related dialogue (Platonov et al., 2020).

2 Background

ULFs are trees written in parenthesized list form. The leaves of these trees, which we will refer to as *atoms*, can be:

- Surface words marked with suffix tags of their semantic types (e.g. `.v`, `.n`, `.pro`, `.d` for verbs, nouns, pronouns, and determiners, respectively);
- Case-sensitive symbols such as names and titles marked with pipes (e.g. `|Glenn|`). Pipe-marked symbols may be left without a semantic tag, in which case they are interpreted as having an entity type;
- One of a closed set of logical and macro symbols (e.g. `k` and `mod-n` for denoting kind-forming and noun modifier-forming operators, respectively). These symbols have unique types and are left without suffix tags.

Figure 1 contains an example ULF for the sentence "*Mary placed Glenn under anesthesia.*" The different types of atoms described above are all present here. The names "Mary" and "Glenn" are enclosed in pipes and the other surface words have POS-related semantic tags (e.g. `place.v`). The type-shifter `k` is used to turn the nominal predicate `anesthesia.n` into a *kind*, which is an abstract individual whose instances are entities. The special operator `past` is used to specify the tense of the verb `place.v`.

As mentioned before, there is a machine learning-based parser to convert English sentences (Kim et al., 2021a) to ULFs. A brief description of how the parser works is given in Appendix A. In the other direction, Kim et al. (2019) introduced a simple ULF-to-English translator, `ulf2english`, which they reported as achieving 78% grammaticality. Broadly speaking, `ulf2english` works by analyzing the ULF type of each clause, adding morphological details based on that analysis, removing purely logical operators, and mapping logical symbols to their corresponding surface forms. A more up-to-date version (whose performance exceeds the evaluation in that paper to an unknown degree)[2] is used in our sampling system.

2.1 The ULF Type System

The EL/ULF type system is the backbone upon which our data augmentation system is built. The semantics of EL are defined over a domain of individuals denoted by $\mathcal{D}$ and a set of truth values denoted by $\mathbf{2}$. A set of situations $\mathcal{S} \subset \mathcal{D}$ consisting of first-class individuals provides the basis for intensionality.[3] Since EL is a first-order logic, the domain $\mathcal{D}$ contains all the individuals that can be spoken about directly. $\mathcal{D}$ not only contains ordinary individuals and situations, but also collections, kinds of entities, propositions, and more. Special type-shifting operators are used to access these other individuals. For example, the so-called *kind* operator `k` can be applied to the nominal predicate `dog.n` (i.e. `(k dog.n)`) to talk about "dogs" as a whole (as opposed to any particular dog or collection of dogs). Predicates can be thought of as true/false-valued functions that take a certain number of objects from the domain and a situation as input. Viewing that in a curried form gives us the type of an arbitrary predicate: $(\mathcal{D} \rightarrow (\mathcal{D} \rightarrow (\cdots \rightarrow (\mathcal{D} \rightarrow (\mathcal{S} \rightarrow \mathbf{2}))\cdots)))$. For convenience, we shorten this to $(\mathcal{D}^n \rightarrow (\mathcal{S} \rightarrow \mathbf{2}))$ where n is the number of $\mathcal{D}$s in the previous type.[4] For our purposes (where we are mostly concerned with ULFs) intensionality is not very relevant, and so henceforth we will abbreviate $(\mathcal{S} \rightarrow \mathbf{2})$ by $\widehat{\mathbf{2}}$. Since monadic predicates (type $(\mathcal{D} \rightarrow \widehat{\mathbf{2}})$) commonly occur in the type system, we will use $\mathcal{N}$ as a shorthand for $(\mathcal{D} \rightarrow \widehat{\mathbf{2}})$.

A couple of key differences exist between the ULF and EL type systems. ULF types may have syntactic restrictions, denoted by subscripts, e.g., a verbal monadic predicate is denoted by $\mathcal{N}_V$. Determiners are denoted with the type $(\mathcal{N} \rightarrow \mathcal{D})$, which anticipates their replacement in EL by a variable of type $\mathcal{D}$ bound by a restricted quantifier.

Each ULF atom can be one of a few related

[2] `https://github.com/genelkim/ulf2english`

[3] The description of EL semantics we give is informal and limited to our purposes. For a more detailed, formal discussion we recommend reading Schubert and Hwang (2000, pp. 9–14).

[4] For technical reasons, EL supplies the situations last.

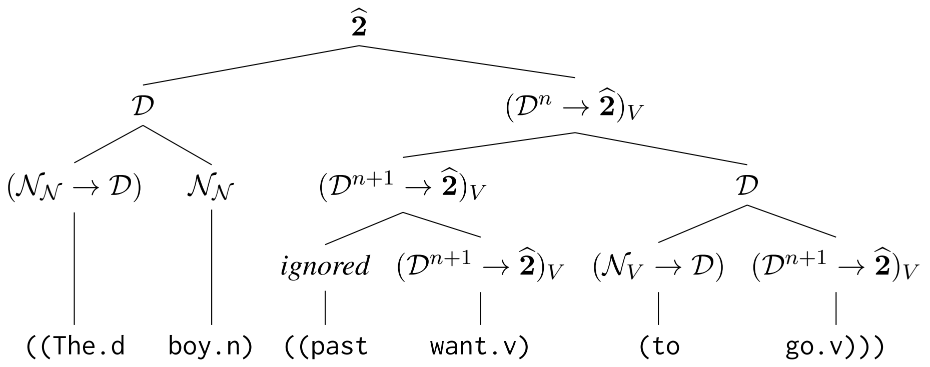

$\widehat{\mathbf{2}}$
$\mathcal{D}$
$(\mathcal{D}^n \to \widehat{\mathbf{2}})_V$
$(\mathcal{N}_{\mathcal{N}} \to \mathcal{D})$
$\mathcal{N}_{\mathcal{N}}$
$(\mathcal{D}^{n+1} \to \widehat{\mathbf{2}})_V$
$\mathcal{D}$
ignored
$(\mathcal{D}^{n+1} \to \widehat{\mathbf{2}})_V$
$(\mathcal{N}_V \to \mathcal{D})$
$(\mathcal{D}^{n+1} \to \widehat{\mathbf{2}})_V$
((The.d
boy.n)
((past
want.v)
(to
go.v)))

Figure 2: An example of how atomic ULF types combine to give the type of the ULF.

semantic types. Logical operators have a unique semantic type, whereas suffix-tagged atoms are restricted by the semantic types that correspond to their tags. A detailed correspondence between ULF atoms and their semantic types is given by Kim (2022, pp. 34–40). The types of atoms can *combine* (or *compose*) via function application to give the type of the ULF composed of those atoms. For example, `a.d` which has type $(\mathcal{N} \to \mathcal{D})$, and `dog.n` which has type $\mathcal{N}$ can compose to give `(a.d dog.n)`, with type $\mathcal{D}$. Such ULFs can further compose to give types for more complex ULFs. Figure 2 gives an example of such a type composition. Here, the entire ULF has type $\widehat{\mathbf{2}}$, the type for a complete sentence. Notice that `want.v` has type $(\mathcal{D}^{n+1} \to \widehat{\mathbf{2}})_V$. The variable n (taken to be a non-negative integer) is used to account for the fact that we do not have prior knowledge of how many arguments the verb takes. It is treated as an integer variable until the last step, where we instantiate it to 1 so that $(\mathcal{D}^n \to \widehat{\mathbf{2}})_V$ can combine with $\mathcal{D}$ to give $\widehat{\mathbf{2}}$. Such treatment is typical for verbs and other types that can take a variable number of arguments. We will call trees similar to the one in Figure 2 without the actual ULF atoms *type derivation trees*. A type derivation tree shows one way the types at the leaves can combine to give the type at the root.

All properly annotated ULFs, including ULFs that do not correspond to complete sentences, should have a valid type that can be found by composing the types of its atoms. This fact is what we use to build our ULF sampler. Our method of sampling ULFs produces new ULFs from a *seed* ULF by picking a random subtree of the seed, finding the semantic type of that subtree, and then replacing the subtree with another ULF of the same type. In our experiments, these seed ULFs are ULFs in the training set of the manually annotated ULF dataset released by Kim et al. (2021a). The type structure helps ensure that the result is a valid ULF where at least the composition of semantic types is coherent, and limiting our sampler to small subtrees makes sampling meaningful sentences significantly more likely than generating entire sentences from scratch.

3 System Description

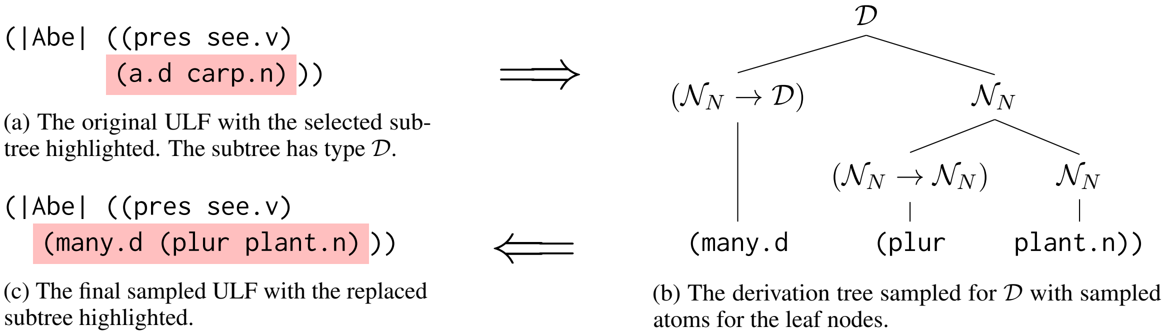
Our system can be broadly broken into two parts: a *sampler* that takes a single seed ULF as input and generates one new ULF-English pair, and a *handler* which uses the sampler repeatedly to augment a given dataset. Pseudocode for the salient parts of this process is given in Appendix E.

3.1 The Sampler

The sampler goes through four phases: (1) picking a random subtree, (2) finding its type, (3) sampling a ULF of that type, and (4) replacing the original subtree in-place. In this subsection we describe that process, illustrating it by walking through the process with the seed ULF `(|Abe| ((pres see.v) (a.d carp.n))` (see Figure 3 for an overview).

3.1.1 Picking a random subtree

This phase involves two parameters that can be tweaked: a maximum size M for the subtree picked, and a "recursion probability" p. Given these parameters and an input ULF, our algorithm first descends the ULF (viewed as a tree) top-down by picking uniformly random children at each level until it reaches a subtree with size (number of leaves) less than or equal to M. Then at each step where it is not at a leaf node it descends another level (by picking a random child) with probability p, and returns the subtree with the current node at its root with probability $1 - p$. If the algorithm ever



```
(|Abe| ((pres see.v)
        (a.d carp.n) ))
```

(a) The original ULF with the selected subtree highlighted. The subtree has type $\mathcal{D}$.

```
(|Abe| ((pres see.v)
   (many.d (plur plant.n) ))
```

(c) The final sampled ULF with the replaced subtree highlighted.

$\mathcal{D}$
$(\mathcal{N}_N \rightarrow \mathcal{D})$
$\mathcal{N}_N$
$(\mathcal{N}_N \rightarrow \mathcal{N}_N)$
$\mathcal{N}_N$
(many.d
(plur
plant.n))

(b) The derivation tree sampled for $\mathcal{D}$ with sampled atoms for the leaf nodes.

Figure 3: The sampling process illustrated.

reaches a leaf node it simply returns it. Pseudocode for this procedure is given in Algorithm 1 in Appendix E. In our running example (in Figure 3a) the recursion goes down the right side of the tree and stops with the subtree `(a.d carp.n)`.

3.1.2 Computing the subtree's type

The selected subtree's type is computed using ULF's type composition rules. We use a pre-existing ULF type system implementation[5] which finds the semantic type of a given ULF fragment by recursively composing types from the atoms in a bottom-up fashion. Due to the presence of variables in some types of leaf nodes (for example for verbs which can have multiple arities), the type system can return a list of possible types corresponding to different values of the variables. In such a case, we pick a random type from this list. Since variables in ULF type compositions rarely take high values (for example, verbs do not frequently take more than three arguments), we pick types corresponding to smaller values of the variable with higher probability. Specifically, if the number of options is less than 4, we pick uniformly. If the number of options is 4 or more, we pick uniformly from the first three options with probability $3/4$, and uniformly from all the options with probability $1/4$. Picking from multiple possible types in a more principled manner (for example by looking at the type composition tree of the seed ULF) could be an avenue for future work in improving our sampler.

Using this process, we find that the chosen subtree in the running example has type $\mathcal{D}$.

3.1.3 Sampling a ULF with a given type

This phase involves one parameter: the maximum size M' for the sampled ULF fragment; and takes one argument: τ_{root}, the desired ULF type (in our running example this is $\mathcal{D}$). To sample a ULF with the given type, we first sample a type derivation tree with τ_{root} at the root. Then, for each leaf type in the derivation tree, we sample a ULF atom with that type. Combining those atoms with the tree structure of the derivation tree gives us a ULF with the desired type.

Sampling a type derivation tree. Sampling a derivation tree is done via three functions: SAMPLETYPEDERIVATION, SAMPLETYPESOURCE, and SAMPLEARGDERIVATIONS. The top-level function is SAMPLETYPEDERIVATION which, as the name suggests, generates a type derivation tree with type τ_{root}. To do so it first uses SAMPLETYPESOURCE to sample a *source type*, τ_{src}, which is a type which can give τ_{root} when supplied 0 or more arguments *and* which is known to be the type of an atomic ULF. It then calls SAMPLEARGDERIVATIONS which takes τ_{root} and τ_{src} and returns a list of derivation trees for the argument types that need to be supplied to τ_{src} to obtain τ_{root}. Finally, SAMPLETYPEDERIVATION combines the source and argument into a derivation tree for τ_{root} which it returns.

SAMPLETYPESOURCE takes one argument, τ_0, and returns a type that can be combined with 0 or more arguments to obtain τ_0 and which can be the type of an atomic ULF. Let T be the set of all types that can be taken by atomic ULFs, and let μ_T be a distribution over T. We take μ_T to be the uniform distribution in our implementation. We leave the selection of a more informed distribution for future work.[6] SAMPLETYPESOURCE iteratively finds all the types in T that can combine with 0 or more arguments to give τ_0 and adds them to a set T'.

[5] https://github.com/genelkim/ulf-lib

[6] For example, while a four-argument verb is possible (e.g. in "I sold my car to John for $400."), it is far less likely than a one- or two-argument verb. A good choice for μ_T might account for that.

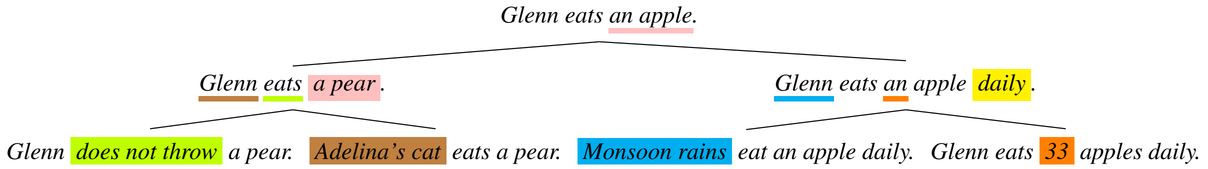

Glenn eats an apple.
Glenn eats a pear.
Glenn eats an apple daily.
Glenn does not throw a pear.
Adelina's cat eats a pear.
Monsoon rains eat an apple daily.
Glenn eats 33 apples daily.

Figure 4: An example of what a tree of sentences (ULFs omitted for brevity) generated from the seed "*Glenn eats an apple*" with depth of 2 and branching factor of 2 might look like. Newly sampled text segments are highlighted. The corresponding replaced text segment in the parent (if any) is underlined with the same color.

It then returns a sample from T' with distribution weights from μ_T.Details on how exactly T' is computed are provided in Algorithm 2 in Appendix E.

SampleArgDerivations takes parameters τ_{cur} and τ_{src}, and computes a list of derivation trees for types that can be composed with τ_{src} to get τ_{cur}. This starts with τ_{cur} and "grows" outward to get τ_{src}. It begins by finding the first type τ_{next} that needs to be prepended to τ_{cur} to get τ_{src}. For instance, if $\tau_{src} = (A \to (B \to (\mathcal{S} \to \mathbf{2})))$ and $\tau_{cur} = (\mathcal{S} \to \mathbf{2})$, then $\tau_{next} = B$. On finding τ_{next}, the algorithm makes a mutually recursive call to SampleTypeDerivation to compute a derivation tree D_{next} for τ_{next}. It then recurses with $\tau_{cur} = (\tau_{next} \to \tau_{cur})$ and the same τ_{src} to obtain a list of derivations, ℓ_D. The algorithm returns $[D_{next}] + \ell_D$. Algorithm 2 in Appendix E contains pseudocode for the entire derivation tree sampling process.

Example. In our running example (Figure 3), the top-level function call is SampleTypeDerivation($\mathcal{D}$). That function calls SampleTypeSource($\mathcal{D}$), which returns the source type $(\mathcal{N}_N \to \mathcal{D})$. This is a valid source type since it can combine with one or more type to give $\mathcal{D}$, and since there are atoms (e.g. `the.d`) which have type $(\mathcal{N}_N \to \mathcal{D})$. The top level function then calls SampleArgDerivations($\tau_{cur} = \mathcal{D}$, $\tau_{src} = (\mathcal{N}_N \to \mathcal{D})$). That function identifies that $\mathcal{N}_N$ can be combined with $(\mathcal{N}_N \to \mathcal{D})$ to get $\mathcal{D}$, and thus calls SampleTypeDerivation($\mathcal{N}_N$).

In turn, that call does the same process as above, but with $\mathcal{N}_N$ as the root. It samples a source which, let us say, turns out to be $(\mathcal{N}_N \to \mathcal{N}_N)$. It then calls SampleArgDerivations($\tau_{cur} = \mathcal{N}_N$, $\tau_{src} = (\mathcal{N}_N \to \mathcal{N}_N)$), which deduces that the required argument type is $\mathcal{N}_N$ and calls SampleTypeDerivation($\mathcal{N}_N$) to find a derivation tree for the argument. In our example, the mutual recursion will end here: the call just mentioned will sample $\mathcal{N}_N$ as the source, which needs no further arguments to get to $\mathcal{N}_N$.

Putting everything together, this process leads to the derivation tree in Figure 3b.

Sampling ULF atoms. After generating a type derivation tree, we sample ULF atoms that have the types at the leaves of the derivation tree. Those atoms are then put in the structure induced by the derivation tree to obtain the ULF sampled. Sampling atoms with given types is done using the ULF lexicon used by Kim et al. (2021a). The sampling is weighted by probabilities computed by normalizing unigram counts from the Google n-gram dataset (Franz and Brants, 2006). In our example there are three leaf nodes with types $(\mathcal{N}_N \to \mathcal{D})$, $(\mathcal{N}_N \to \mathcal{N}_N)$, and $\mathcal{N}_N$. Suppose they are instantiated to the atoms `many.d`, `plur`, and `plant.n`.

3.1.4 Replacing in place

The final sampled ULF is obtained from the input ULF by replacing the random subtree picked in the first phase with the ULF fragment sampled in the previous phase. This is done using simple tree operations. In our example, this leads to the final sampled ULF, `(|Abe| ((pres see.v) (many.d (plur plant.n))))`.

3.2 The Handler

The sampling handler takes three inputs: the dataset that is to be augmented, a depth d, and a branching factor b. For each ULF U in the dataset, the handler performs the following steps:

1. Use the sampler b times on input ULF U to get b different samples from the seed U.

2. On each new ULF U' sampled in the previous step, use the sampler b times.

3. Repeat step 2 d times, thus obtaining a tree of ULFs with depth d and branching factor b. In this tree, each node is obtained from its parent via an application of the sampler. Figure 4 shows an example of such a tree.

4. Collect all the ULFs in this tree along with their English translations (which are found using the ULF-to-English library) into a set.

Combining all the sets obtained from the above process gives us a raw augmented dataset.

After generating a raw augmented dataset we assign a quality score using language model perplexity. The final dataset consists of the top $F * N$ ULF-English pairs according to the quality score, where N is total number of samples and F is a preset constant proportion ($0 \leq F \leq 1$). We use the GPT-2 language model (Radford et al., 2019) in our implementation. This last pruning step is done in order to remove highly incoherent results. Algorithm 3 contains pseudocode for the handler.

The reason we branch out from the seed instead of repeatedly modifying in a linear fashion is that in a linear design, if the sampler ever returns an incoherent result, every sentence generated from then onwards is likely to be incoherent too. This leads to a lot of "wasted" seeds leading to a smaller yield of good ULF-English pairs. In our branching-based design, even if one sample ends up being incoherent, the other branches of the algorithm still remain viable.

3.3 ULF Macros

One notable limitation of our sampler is that it does not support most ULF macros. ULF macros perform unique transformations over their arguments to handle complex but regular mappings from syntax to semantic structure (e.g., topicalization, post-nominal modification, etc.) and do not fit directly into the type-compositional system.

4 Experiments

We evaluate our sampler on the hand-annotated ULF 1.0 dataset by Kim et al. (2021a), the only dataset of gold ULF annotations that we are aware of. This dataset has 1,378, 180, and 180 sentences of ULF-English pairs in the training, development, and test sets, respectively.

Metrics. Following prior work on this dataset, we use SEMBLEU as the primary evaluation metric and use EL-SMATCH secondarily for analysis, since it is broken down into F1, precision, and recall components. The SEMBLEU score better reflects the the parser's ability to generate coherent ULFs because it takes into account chains of multiple nodes and edges that EL-SMATCH does not. Thus, SEMBLEU is used as the primary evaluation metric for ULF parsing. Kim and Schubert (2016) describes EL-SMATCH in detail, which includes a method for representing ULFs as a set of triples similar to AMRs. When SEMBLEU is run on ULF, the same set-of-triples representation is used for ULFs so that the metric designed for AMR can be run on ULF.

d	b	M	M'	p	N
1	3	5	5	0.5	5,035
2	3	5	5	0.5	14,503
3	1	5	5	0.5	4,777
3	2	5	5	0.5	16,050
3	3	5	5	0.5	40,708
3	4	5	5	0.5	83,383
4	3	5	5	0.5	116,112

Table 1: Sampling parameters and the resulting dataset sizes. The table uses the same variable conventions as Section 3 for sampling parameters and dataset size.

4.1 Settings

Model. In order to isolate the benefits of the data augmentation method, we use the current state-of-the-art ULF parsing model (Kim et al., 2021a). This parser is described in detail in Appendix A. While Kim et al. (2021a) released the code for their model, it runs on PyTorch 1.2 with Python 3.6 which are incompatible with the drivers in some of our more up-to-date computing machines. We updated the code to use PyTorch 1.11 and Python 3.10. This initially led to a reduction in parser performance, but we found that we could replicate the original parser performance by reducing the step size by a factor 0.25 and doubling the number of training epochs. We detail the replication experiment in Appendix D, including the model hyperparameters. We use the model that successfully replicated the original results in our experiments.

Sampled Datasets. The sampling parameter combinations we test are listed in Table 1 along with the number of unique examples that result from this sampling process. We vary the handler parameters: depth and branching factors, which largely determine the number of samples. We fix the subtree sampling parameters: maximum sample size to 5, maximum replacement size to 5, and recursion probability to 0.5. During the development process, we found this to lead to the best balance of quality and speed. We remove duplicated samples

d	b	SemBLEU	EL-Smatch		
			F1	Precision	Recall
Reported		47.4	**59.8**	60.7	**59.0**
Replicated		47.1	58.7	60.6	56.9
1	3	48.2	59.5	61.6	57.6
2	3	46.0	58.2	59.5	57.0
3	1	47.9	59.0	**61.8**	56.5
3	2	46.1	57.9	59.8	56.1
3	3	48.3	**59.8**	61.5	57.8
3	4	47.8	58.1	60.1	56.3
4	3	**49.0**	59.3	60.9	57.8

Table 2: Test set parser performance for augmented training on various sampling parameters and no filtering—the average of 5 runs with different random seeds.

to reduce unintended bias towards these sentences.

LM Filtering. To evaluate the trade-off between sample quality and quantity, we vary the number of LM-filtered samples in our final augmented datasets. For each sampled dataset, we retain the following proportions of samples: 0.1, 0.25, 0.5, and 1.0. We limit our filtering experiments to the three largest sampled sets. This ensures that sufficient samples remain even after aggressive filtering.

4.2 Training & Hyperparameters

We modify the training process of the baseline model to include some number of epochs where the model trains on both the manually annotated ULF examples and the type-sampled dataset. After that, the remaining epochs are trained using only the manually annotated ULF examples. Other than this new hyperparameter, the only hyperparameter that is changed from the original model is the total number of epochs. We reduce the number of total epochs trained since the model begins to overfit earlier when a larger sampled dataset is added.

We estimated the number of epochs at which the model begins to overfit with sample augmentation using $d = 3$ and $b = 3$ at 1.0, 0.5, 0.25, and 0.1 filtering proportions. For these parameters, we set the augmented training epochs to 1 greater than where we consistently saw overfitting.[7] We then generalize this to other experiments under the assumption that similarly sized datasets will begin to overfit at similar numbers of epochs. The training epoch specifics are provided in Appendix B.

4.3 Results

In this section, we report only the average test set metrics. Appendix B reports the full results including the development set metrics and standard deviations for both test and development sets.

Handler Parameters. We first compare the performance of the baseline model when augmented with the unfiltered samples from the sampler with sampling parameters specified in Table 1. These results are reported in Table 2. The model augmented with $d = 4$ and $b = 3$ has the best performance, with a SemBLEU score that is 1.6 points over the reported baseline and 1.9 points over the replicated baseline. Augmenting the training with sampled pairs improves SemBLEU scores for most sampler parameters. Under closer inspection, we find a curious pattern in these results. When we fix b to 3 and vary d from 1 to 4, we see a U-shaped SemBLEU performance curve. Similarly, when we fix d to 3 and vary b from 1 to 4, we see a similar pattern, though the performance drops a bit again when $b = 4$.

The rise in SemBLEU scores with data augmentation is not reflected as strongly in the EL-Smatch scores. The EL-Smatch F1 scores are typically slightly higher than the replicated baseline, but still under the score reported by Kim et al. (2021a). This suggests that the augmented samples push the model towards overall parse coherence without much changing the expected performance on a particular node or edge.[8]

LM Filtering. Table 3 shows the parser performance when the augmented dataset is filtered at different levels based on LM perplexity. Moderate filtering (0.5) tends to result in a small improvement, leading to the best SemBLEU results in this paper of 49.1 on the $d = 3$, $b = 3$ dataset. Curiously, moderate filtering seems to push the model toward higher EL-Smatch recall over precision.

Aggressive filtering (0.1) consistently degrades performance, even relative to the baseline model. This does not seem to be due to dataset size, since similarly sized augmented datasets in Table 2 ($d = 1, b = 3$ and $d = 3, b = 1$) still improve over the baseline. This suggests that aggressive LM filtering

[7] We consider an increase in development set perplexity to be an overfit model.

[8] EL-Smatch scores are based on overlaps of individual nodes and edges whereas SemBLEU scores are based on chains of node-edge-node links.

d	*b*	Filter	SEMBLEU	EL-SMATCH		
				F1	Precision	Recall
Reported			47.4	**59.8**	60.7	**59.0**
Replicated			47.1	58.7	60.6	56.9
3	3	1.00	48.3	**59.8**	**61.5**	57.8
		0.50	**49.1**	**59.8**	61.0	58.6
		0.25	46.6	58.3	59.3	57.4
		0.10	46.9	59.7	60.8	58.7
3	4	1.00	47.8	58.1	60.1	56.3
		0.50	47.9	59.0	60.4	57.5
		0.25	47.5	59.0	60.1	57.9
		0.10	46.6	58.4	59.8	56.4
4	3	1.00	49.0	59.3	60.9	57.8
		0.50	48.1	59.5	60.2	58.9
		0.25	48.1	59.0	60.0	58.1
		0.10	45.3	57.9	58.9	56.9

Table 3: Test set parser performance for LM-filtered augmented data for larger type sampling parameters—the average of 5 runs with different random seeds.

removes useful variance in the samples and leads to overfitting to low-perplexity sentences.

4.4 Qualitative Evaluation

We performed a qualitative analysis of the sampled sentences in an early version of the sampler[9] to evaluate the syntactic and semantic coherence of the generated samples. This experiment used $d = 3, b = 2$ sampling parameters and LM filtering to a dataset size of 5,000 samples. 400 randomly selected examples from this set were scored by human evaluators for both syntactic and semantic coherence, each on a 5-point scale. This resulted in a mean syntax score of 3.87 and a mean semantics score of 3.96, showing that the sampler typically succeeds in generating ULFs corresponding to well-formed and understandable text. Appendix C provides exact prompts given to human evaluators and more details of the results.

5 Related Work

Gibson and Lawley (2022) fine-tune GPT2-large on the ULF 1.0 dataset to learn both an English to ULF parser and a ULF to English generator. Their ULF parser underperforms Kim et al.'s (2021a) on the primary SEMBLEU metric but achieves the state-of-the-art on the EL-SMATCH metric. Their ULF to English generator matches or outperforms the `ulf2english` system on automatic machine translation metrics, BLEU (Papineni et al., 2002), chrF++ (Popović, 2017), and METEOR (Banerjee and Lavie, 2005) but uses more compute resources.

Data augmentation is far from a new idea for training neural networks. Data augmentation in computer vision is common via translation, rotation, cropping, flipping, noise injection, and color space transformations (Shorten and Khoshgoftaar, 2019). NLP has its own suite of data augmentation techniques that have been explored with token-level perturbations (Wei and Zou, 2019), graph-level perturbations (Şahin and Steedman, 2018), example interpolation (Zhang et al., 2018; Verma et al., 2019; Faramarzi et al., 2022), and distributional model-based synthetic sampling (Sennrich et al., 2016; Yang et al., 2020; Kobayashi, 2018) covering the major common approaches. Feng et al. (2021) provide a comprehensive survey of the NLP data augmentation approaches.

Focusing in on semantic parsing, Jia and Liang (2016) and Yu et al. (2021) learn synchronous context-free grammars using available data from which new examples are sampled. Andreas (2020) infers shared lexical environments and performs substitutions of words between them to encourage compositionality in semantic parsers. van Noord and Bos (2017) cross-reference two independent AMR parsers to automatically generate likely-high-quality examples which led to major parsing performance gains. None of these methods are able to exploit the knowledge that we have about ULF types and the rules that mediate their composition. Some of the approaches described in this section, such as van Noord and Bos' (2017), could be used in conjunction with our approach.

6 Conclusions

We presented a data augmentation method for ULFs that leverages ULF's underlying semantic type structure. This method helps alleviate the data sparsity problem that currently exists for ULF parsing, leading to a new state-of-the-art in this task without any change in the parsing model. Though we tested our data augmentation method on ULFs, this technique is applicable to any semantic parsing task with an underlying tree-structured compositional type system. For example, parsing in combinatory categorical grammar (Steedman, 2000) is another appropriate candidate for this sampling technique. Some details of the sampling procedure can also be improved in obvious, but not trivial ways. For example, our ULF atom sampling procedure uses word frequencies without ULF type

[9] This version failed to properly propagate certain syntactic restrictions leading to sampling failures, in which case we repeated the sampling process.

information.This leads to an over-representation of type-ambiguous words in our generated samples.

We think that type system-driven data augmentation for ULF is a promising way to further improve ULF parser performance. We expect further parsing improvements through refinement of the sampling parameters and expansion of the sampler to include macros. The additional data provided by such augmentation would support more general neural network-based semantic parsers as have been successful in other semantic representations (van Noord et al., 2018; Liu et al., 2018; Buys and Blunsom, 2017; Konstas et al., 2017). We are hopeful to see an improved semantic parser find utility in ULF-related tasks such as those mentioned at the end of section 1.

Acknowledgements

This work was supported in part by NSF grant IIS-1940981. We thank Omar Abdelrahman for assisting in the replication of the ULF parser performance for the newer Python version. We are grateful to the anonymous reviewers for their helpful feedback.

References

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Jan Buys and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1215–1226, Vancouver, Canada. Association for Computational Linguistics.

Mojtaba Faramarzi, Mohammad Amini, Akilesh Badrinaaraayanan, Vikas Verma, and Sarath Chandar. 2022. Patchup: A feature-space block-level regularization technique for convolutional neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):589–597.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Alex Franz and Thorsten Brants. 2006. All our n-gram are belong to you. `https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html`. Google AI Blog.

Erin Gibson and Lane Lawley. 2022. Language-model-based parsing and english generation for unscoped episodic logical forms. *The International FLAIRS Conference Proceedings*, 35.

Daniel Gildea, Giorgio Satta, and Xiaochang Peng. 2018. Cache transition systems for graph parsing. *Computational Linguistics*, 44(1):85–118.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Gene Kim, Viet Duong, Xin Lu, and Lenhart Schubert. 2021a. A transition-based parser for unscoped episodic logical forms. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 184–201, Groningen, The Netherlands (online). Association for Computational Linguistics.

Gene Kim, Mandar Juvekar, Junis Ekmekciu, Viet Duong, and Lenhart Schubert. 2021b. A (mostly) symbolic system for monotonic inference with unscoped episodic logical forms. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA)*, pages 71–80, Groningen, the Netherlands (online). Association for Computational Linguistics.

Gene Kim, Mandar Juvekar, and Lenhart Schubert. 2021c. Monotonic inference for underspecified episodic logic. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA)*, pages 26–40, Groningen, the Netherlands (online). Association for Computational Linguistics.

Gene Kim, Benjamin Kane, Viet Duong, Muskaan Mendiratta, Graeme McGuire, Sophie Sackstein, Georgiy Platonov, and Lenhart Schubert. 2019. Generating discourse inferences from unscoped episodic logical formulas. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 56–65, Florence, Italy. Association for Computational Linguistics.

Gene Kim and Lenhart Schubert. 2016. High-fidelity lexical axiom construction from verb glosses. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 34–44, Berlin, Germany. Association for Computational Linguistics.

Gene Louis Kim. 2022. *Corpus annotation, parsing, and inference for Episodic Logic type structure*. Ph.D. thesis, University of Rochester.

Gene Louis Kim and Lenhart Schubert. 2019. A type-coherent, expressive representation as an initial step to language understanding. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 13–30, Gothenburg, Sweden. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Lane Lawley, Gene Louis Kim, and Lenhart Schubert. 2019. Towards natural language story understanding with rich logical schemas. In *Proceedings of the Sixth Workshop on Natural Language and Computer Science*, pages 11–22, Gothenburg, Sweden. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2018. Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 429–439, Melbourne, Australia. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Fabrizio Morbini and Lenhart Schubert. 2009. Evaluation of EPILOG: a reasoner for Episodic Logic. In *Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning*, Toronto, Canada.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands*, 7.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Xiaochang Peng, Linfeng Song, Daniel Gildea, and Giorgio Satta. 2018. Sequence-to-sequence models for cache transition systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1842–1852, Melbourne, Australia. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Georgiy Platonov, Lenhart Schubert, Benjamin Kane, and Aaron Gindi. 2020. A spoken dialogue system for spatial question answering in a physical blocks world. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 128–131, 1st virtual meeting. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Lenhart Schubert. 2014. From treebank parses to episodic logic and commonsense inference. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 55–60, Baltimore, MD. Association for Computational Linguistics.

Lenhart K. Schubert. 2000. The situations we talk about. In Jack Minker, editor, *Logic-based Artificial Intelligence*, pages 407–439. Kluwer Academic Publishers, Norwell, MA, USA.

Lenhart K. Schubert and Chung Hee Hwang. 2000. Episodic Logic meets Little Red Riding Hood: A comprehensive natural representation for language understanding. In Lucja M. Iwańska and Stuart C. Shapiro, editors, *Natural Language Processing and*

Knowledge Representation, pages 111–174. MIT Press, Cambridge, MA, USA.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(60).

Mark Steedman. 2000. *The Syntactic Process*, volume 24. MIT press, Cambridge, MA.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir R. Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

A Baseline ULF Parser Description

Kim et al.'s (2021a) ULF parser is a transition system-based parser where the transition actions are selected using an LSTM. This parser modifies the cache transition parser (Gildea et al., 2018) to better model ULFs. At a high level, the modification introduces methods of generating ULF symbols on the fly from input words, rather than assuming a sequence of symbols as input. These symbol generation methods are further designed to reflect the relationship between ULF symbols and their corresponding words rather than assuming that an arbitrary mapping can exist between them. The cache transition system oracle, which is needed for training, is similarly modified to support these changes in the possible actions.

The LSTM is then trained to take a concatenation of the relevant input word, the relevant ULF symbol, and the current transition system state features as input and predicts the next action for the transition system. The transition system is inspected to determine which word is relevant, this is called hard attention (Peng et al., 2018). The relevant ULF symbol is similarly inferred from the transition system state and action history. Either we find which symbol we generated based on the current word, or if it has not been generated yet, the most recently generated symbol. The word features include the RoBERTa (Liu et al., 2019) embedding, GloVe embedding (Pennington et al., 2014), and learned embeddings of the lemma, POS tag, and NER tag. The symbol tokens are learned. The transition state features further includes information about the dependency tree distances of relevant words and the transition system phase. We refer you to Kim et al.'s (2021a) paper for further details of the parser.

B Additional Experiment Details

B.1 Augmented Epoch Determination

Filtering	N	overfit epoch
1.00	40,708	2
0.50	20,354	4
0.25	10,177	9

Table 4: Epochs values at which the model begins to overfit when trained with an augmented dataset using $d = 3$ and $b = 3$ parameters at various GPT-2 filtering levels.

d	b	N	F	Aug.	Total
1	3	5,035	1.00	25	45
2	3	14,503	1.00	10	30
3	1	4,777	1.00	25	45
3	2	16,050	1.00	10	30
3	3	40,708	1.00	2	20
3	3	20,354	0.50	5	25
3	3	10,177	0.25	10	30
3	3	5,083	0.10	25	45
3	4	83,383	1.00	2	20
3	4	41,691	0.50	2	20
3	4	20,845	0.25	5	25
3	4	10,422	0.10	10	30
4	3	116,112	1.00	2	20
4	3	58,056	0.50	2	20
4	3	29,028	0.25	5	25
4	3	14,514	0.10	10	30

Table 5: Number of epochs trained on the augmented dataset and in total for each sampling and filtering configuration. "Aug." is the number of augmented epochs. "Total" is the total number of epochs trained. Includes the total size of each sampling configuration results to help interpret the motivation behind the epoch values.

Table 4 shows when the model would begin to overfit at various augmented dataset levels. Specifically, we use the augmented dataset with $d = 3$ and $b = 3$, filtered with GPT-2 at various proportions. We use this to determine the number of epochs that we should train the model with sampling augmented data before only training on the manually annotated dataset. The procedure we use here is to add 1 to the results from Table 4. We do not add 1 to the full $d = 3$ and $b = 3$ dataset. Due to the size of the dataset, 1 additional epoch would likely severely overfit the model. For filtering at a 0.1 level, we extrapolate from the 0.5 and 0.25 levels, assuming a linear relationship between the number of augmenting examples and epochs.

We then generalize these results to other sampling settings under the assumption that similarly sized datasets will begin to overfit at similar numbers of epochs. We select the filtering level for the $d = 3$, $b = 3$ dataset whose N value is the closest lower value to the augmenting dataset in question. Table 5 lists the number of epochs that we trained each model on the augmented set and in total.

As with the rest of the parser details, we follow Kim et al.'s (2021a) approach to selecting the test model. After all training epochs, we select

Model		SemBLEU		EL-Smatch					
d	b			F1		Precision		Recall	
		Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)
Reported		46.4 ± 1.4	47.4 ± 1.3	58.4 ± 0.7	**59.8** ± 1.0	59.1 ± 1.1	60.7 ± 1.5	57.8 ± 0.5	**59.0** ± 0.7
Replicated		46.4 ± 2.7	47.1 ± 1.4	56.9 ± 1.9†	58.7 ± 1.4	59.5 ± 1.8†	60.6 ± 1.7	54.6 ± 2.2†	56.9 ± 1.3
1	3	48.7 ± 1.6	48.2 ± 1.2	58.4 ± 0.7	59.5 ± 0.7	61.3 ± 1.5	61.6 ± 1.3	55.7 ± 0.9	57.6 ± 0.6
2	3	46.9 ± 1.9	46.0 ± 1.6	56.9 ± 1.7	58.2 ± 1.0	59.6 ± 1.6	59.5 ± 1.1	54.4 ± 1.9	57.0 ± 1.7
3	1	49.2 ± 1.1	47.9 ± 1.5	58.4 ± 1.0	59.0 ± 0.8	61.9 ± 0.6	**61.8** ± 0.9	55.3 ± 1.5	56.5 ± 1.2
3	2	48.0 ± 3.3	46.1 ± 3.3	57.5 ± 2.0	57.9 ± 1.7	60.7 ± 2.2	59.8 ± 1.8	54.7 ± 2.3	56.1 ± 2.4
3	3	49.6 ± 1.6	48.3 ± 1.7	59.2 ± 1.5	**59.8** ± 1.4	62.1 ± 1.7	61.5 ± 1.4	56.7 ± 1.6	57.8 ± 2.2
3	4	49.3 ± 3.8	47.8 ± 4.0	58.3 ± 2.2	58.1 ± 2.4	61.1 ± 1.8	60.1 ± 2.0	55.7 ± 2.6	56.3 ± 3.2
4	3	50.4 ± 0.8	**49.0** ± 1.0	58.7 ± 1.1	59.3 ± 1.7	61.2 ± 1.0	60.9 ± 1.2	56.5 ± 1.3	57.8 ± 2.5

Table 6: Detailed parser performance for augmented training on various sampling parameters and no filtering—the average & standard deviation of 5 runs. See the caption for Table 10 regarding the meaning of the † superscript.

Model			SemBLEU		EL-Smatch					
d	b	F			F1		Precision		Recall	
			Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)
Reported			46.4 ± 1.4	47.4 ± 1.3	58.4 ± 0.7	**59.8** ± 1.0	59.1 ± 1.1	60.7 ± 1.5	57.8 ± 0.5	**59.0** ± 0.7
Replicated			46.4 ± 2.7	47.1 ± 1.4	56.9 ± 1.9†	58.7 ± 1.4	59.5 ± 1.8†	60.6 ± 1.7	54.6 ± 2.2†	56.9 ± 1.3
3	3	1.00	49.6 ± 1.6	48.3 ± 1.7	59.2 ± 1.5	**59.8** ± 1.4	62.1 ± 1.7	**61.5** ± 1.4	56.7 ± 1.6	57.8 ± 2.2
		0.50	49.5 ± 0.8	**49.1** ± 1.7	58.7 ± 0.5	**59.8** ± 1.0	61.2 ± 0.9	61.0 ± 1.5	56.4 ± 0.5	58.6 ± 0.8
		0.25	47.6 ± 1.8	46.6 ± 1.8	57.4 ± 1.3	58.3 ± 1.4	59.8 ± 1.3	59.3 ± 1.9	55.3 ± 1.4	57.4 ± 1.3
		0.10	47.2 ± 1.5	46.9 ± 1.5	57.9 ± 0.6	59.7 ± 0.8	59.7 ± 1.8	60.8 ± 1.5	56.2 ± 0.7	58.7 ± 0.9
3	4	1.00	49.3 ± 3.8	47.8 ± 4.0	58.3 ± 2.2	58.1 ± 2.4	61.1 ± 1.8	60.1 ± 2.0	55.7 ± 2.6	56.3 ± 3.2
		0.50	48.6 ± 1.1	47.9 ± 1.4	58.4 ± 1.0	59.0 ± 0.9	61.1 ± 0.8	60.4 ± 1.2	55.8 ± 1.4	57.5 ± 0.9
		0.25	47.3 ± 2.7	47.5 ± 2.4	57.4 ± 1.6	59.0 ± 2.2	59.8 ± 2.5	60.1 ± 2.7	55.1 ± 0.8	57.9 ± 1.9
		0.10	46.7 ± 2.4	46.6 ± 2.2	57.2 ± 2.0	58.4 ± 2.4	60.1 ± 1.8	59.8 ± 1.7	54.7 ± 2.4	56.4 ± 3.3
4	3	1.00	50.4 ± 0.8	49.0 ± 1.0	58.7 ± 1.1	59.3 ± 1.7	61.2 ± 1.0	60.9 ± 1.2	56.5 ± 1.3	57.8 ± 2.5
		0.50	49.0 ± 3.1	48.1 ± 3.6	59.2 ± 1.8	59.5 ± 2.1	60.9 ± 2.2	60.2 ± 2.4	57.7 ± 1.5	58.9 ± 1.9
		0.25	48.3 ± 1.3	48.1 ± 1.7	57.8 ± 0.8	59.0 ± 0.8	60.0 ± 0.9	60.0 ± 1.4	55.7 ± 0.8	58.1 ± 0.6
		0.10	45.5 ± 2.7	45.4 ± 2.7	56.9 ± 2.2	57.9 ± 2.1	58.9 ± 2.3	58.9 ± 1.8	55.1 ± 2.3	56.9 ± 1.8

Table 7: Detailed parser performance for LM-filtered augmented data for larger type sampling parameters—the average & standard deviation of 5 runs.

the epoch at which the model has the best development set SemBLEU performance and restore that checkpoint for testing.

B.2 Detailed Parser Results

Table 6 shows the full detailed parsing results with full augmented datasets, no filtering. These results include the development set results and standard deviations. These details should help in checking replication. It also shows that adding the augmented data tends to lead to more overfitting of the model. That is, the development set performance is relatively higher compared to the test set performance when using data augmentation. Still, the average test set performance is only a point or two below the average development set performance so the overfitting does not tend to be very severe. The standard deviations also show that certain sampling configurations lead to very unstable training. $d = 3, b = 4$ for example has a 4-point standard deviation in SemBLEU scores. Table 7 shows similarly detailed results for the filtering experiments.

B.3 Hyperparameters

Model hyperparameters are listed in Table 8. All of them except for the learning rate are grandfathered in from Kim et al.'s (2021a) parser.

C Qualitative Evaluation Details

For the qualitative analysis, we sampled an augmented dataset using the following parameters $d = 5, b = 2, M = 5, M' = 5, p = 0.5$. This earlier version of the parser performed filtering based on a maximum augmented size, including the seed examples, rather than filtering proportional to only the sampled set. We set the maximum size to 5,000. Excluding the 1,378 seed sentences (the training set of ULF 1.0), this results in 3,622 new samples. Of these, we uniformly randomly select 400 and had human evaluators rank the English translations (using `ulf2english`) for both syntactic and semantic coherence. Each example was

GloVe.840B.300d	
dim	300
RoBERTa embeddings	
source	RoBERTa-Base
dim	768
POS tag embeddings	
dim	100
Lemma embeddings	
dim	100
CharCNN	
num_filters	100
ngram_filter_sizes	[3]
Action embeddings	
dim	100
Transition system feature embeddings	
dim	25
Word encoder	
hidden_size	256
num_layers	3
Symbol encoder	
hidden_size	128
num_layers	2
Action decoder	
hidden_size	256
num_layers	2
MLP decoder	
hidden_size	256
activation_function	ReLU
num_layers	1
Optimizer	
type	ADAM
learning_rate	**0.0025**
max_grad_norm	5.0
dropout	0.33
num_epochs	25
Beam size	3
Vocabulary	
word vocab size	9200
symbol vocab size	7300
Batch size	32

Table 8: Model hyperparameters. The learning rate, which differs from Kim et al.'s (2021a) parser, is bolded.

ranked. The samples were distributed among three in-person human volunteers for ranking. Volunteers were given descriptions for the meaning of each score value. These are provided in Table 9.

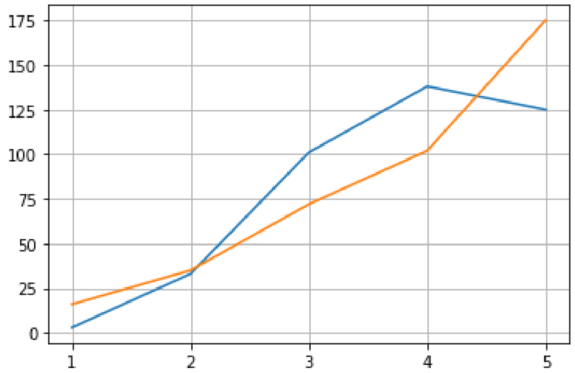


Figure 5: Frequencies for each score value reported by scorers. Score frequencies for syntax are in blue, those for meaning are in orange.

They were also asked to treat syntax and meaning as orthogonal properties as far as possible.

Figure 5 plots the frequencies for the qualitative scores. The mean syntax score was 3.87 with a standard deviation of 0.97. The mean meaning score was 3.96 with a standard deviation of 1.15. The medians for both scores were 4. About 65.7% of examples scored 4s and 5s on syntax, and about 69.2% scored 4s and 5s on meaning. Very few (less than 40 each) scored 1s and 2s on either categories. According to the descriptions given to the volunteers, this means that the average sentence was somewhere between "some syntactic inaccuracies, but overall not bad" (a score of 3) and "minor syntax errors" (a score of 4) leaning heavily towards the latter, and was just a little below "meaning is clear but a little strange for the average ear" (a score of 4) for meaning.

D Baseline Replication

The results for the baseline replication experiments are presented in Table 10. These results are based on 5 random runs, however, due to technical challenges, a few results are based on 4 random runs. This was the first experiment run for this paper so the infrastructure was still brittle. We did not redo these failed runs since a single additional run would not affect our conclusions in this circumstance.

When we run the unmodified parameters with our code updated to newer Python and PyTorch releases, we see that our SEMBLEU performance drops by 4.5 points. However, when we reduce the learning rate from 0.001 to 0.00025 and increase the total epochs from 25 to 60, the performance difference is only 0.3. Considering that the standard deviations of the SEMBLEU scores are 1.3 and

Score	Description
1	Completely garbled
2	Garbled up but there are sizable chunks that are coherent
3	Some inaccuracies in grammar, but overall not bad
4	Minor syntax errors
5	Grammatical

Score	Description
1	This doesn't mean anything
2	You could speculate what it means, but it isn't very coherent
3	Either somewhat clear but still unclear, or quite implausible
4	Meaning is clear but a little strange for the average ear
5	Makes sense, is plausible

Table 9: Descriptions of scores given to volunteers. The first table corresponds to syntax scores and the second corresponds to scores for meaning.

1.4 for the reported and our modified runs, respectively, 0.3 is within the range of sample variance. EL-SMATCH results are similar, though our replicated runs are relatively stronger on precision over recall.

E Pseudocode for Algorithms

Algorithms 1 to 3 are the pseudocode algorithms for the PICKRANDOMSUBTREE, SAMPLETYPEDERIVATION, and AUGMENTDATASET, respectively, which are described in Section 3.

Algorithm 2, however, elides some implementational caveats. First, in practice, we add a global parameter M' that imposes a maximum on the number of leaves in the sampled tree. This is implemented by limiting the amount of mutual recursion that happens between SAMPLETYPEDERIVATION and SAMPLEARGDERIVATIONS. Second, while the pseudocode uses simple equality to compare types, in practice we use a TYPEMATCH function which takes two types τ and τ' and returns true if and only if τ' is the same as τ, except possibly with *additional* syntactic restrictions. Third, in practice SAMPLETYPESOURCE can return some non-atomic ULF types which are known to be types of atomic ULFs when operated on with specific operators. This is to account for operators (such as sentential operators) which are ignored during type composition. An example of this is that SAMPLETYPESOURCE can returned a "tensed verb" type which can be instantiated in the next step to a tense operator operating on a verb (e.g. `(pres eat.v)`).

Model	SEMBLEU		EL-SMATCH					
			F1		Precision		Recall	
	Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)	Dev ($\pm\sigma$)	Test ($\pm\sigma$)
Reported	46.4 ± 1.4	47.4 ± 1.3	58.4 ± 0.7	59.8 ± 1.0	59.1 ± 1.1	60.7 ± 1.5	57.8 ± 0.5	59.0 ± 0.7
Unmod.†	44.2 ± 1.7	42.9 ± 2.5	56.2 ± 0.4	56.7 ± 0.6	58.7 ± 1.9	58.3 ± 1.7	53.9 ± 1.2	55.3 ± 1.2
Modified	46.4 ± 2.7	47.1 ± 1.4	$56.9 \pm 1.9^{\dagger}$	58.7 ± 1.4	$59.5 \pm 1.8^{\dagger}$	60.6 ± 1.7	$54.6 \pm 2.2^{\dagger}$	56.9 ± 1.3

Table 10: Results for the baseline replication experiments. Results are based on 5 random runs. A "†" superscript indicates results based on 4 runs due to a system failure on one of the runs. The "Unmod." row contains the results of running our code updated to PyTorch 1.11 and Python 3.10 using the exact same parameters as the original. The "Modified" row contains the results where the learning rate is lowered four-fold and total epochs are increased from 25 to 60.

Algorithm 1 Picking a random subtree of a ULF.

```
global parameters: $M \in \mathbb{N}$, the maximum size of the subtree picked; $p \in (0, 1)$, a probability.
function PickRandomSubtree($U$)
    input: $U$, a ULF.
    if Size($U$) $> M$ then
        $U' \leftarrow$ (uniformly) random child of $U$.
        return PickRandomSubtree($U'$).
    else if $U$ is atomic then
        return $U$.
    else
        $U' \leftarrow$ (uniformly) random child of $U$.
        return $\begin{cases} \text{PickRandomSubtree}(U') \\ \quad \text{with probability } p; \\ U \text{ otherwise.} \end{cases}$
    end if
end function
```

Algorithm 2 Sampling a type derivation tree for a given type. This pseudocode ignores some implementation details. Those details are explained in the text description of this algorithm.

```
function SAMPLETYPEDERIVATION($\tau_{root}$)
    $\tau_{src} \leftarrow$ SAMPLETYPESOURCE($\tau_{root}$)
    $\overrightarrow{\tau_{args}} \leftarrow$ SAMPLEARGDERIVATIONS($\tau_{root}, \tau_{src}$)
    return $(\tau_{src}, \overrightarrow{\tau_{args}})$
end function
function SAMPLETYPESOURCE($\tau_0$)
    global parameters: $T$, the set of possible types of ULF atoms; $\mu_T$, a distribution over $T$.
    $T' \leftarrow \emptyset$
    for $\tau_a \in T$ do
        $\tau_{tmp} \leftarrow \tau_0$
        while $\tau_{tmp} \neq$ NIL do
            if $\tau_a = \tau_{tmp}$ then
                $T'$.append($\tau_a$)
            end if
            if $\tau_{tmp} \in \{\mathcal{D}, \mathcal{S}, \mathbf{2}\}$ then
                $\tau_{tmp} \leftarrow$ NIL
            else
                $\tau_{tmp} \leftarrow$ CODOMAIN($\tau_{tmp}$)
            end if
        end while
    end for
    return $\tau_{src} \sim \mu_T(T')$
end function
function SAMPLEARGDERIVATIONS($\tau_{cur}, \tau_{src}$)
    if $\tau_{src} = \tau_{cur}$ then
        return []
    end if
    $\tau_{arg} \leftarrow$ NEXTARGTYPE($\tau_{cur}, \tau_{src}$)
    $D_{arg} \leftarrow$ SAMPLETYPEDERIVATION($\tau_{arg}$)
    $\tau_{next} \leftarrow (\tau_{arg} \rightarrow \tau_{cur})$
    return $[D_{arg}] +$ SAMPLEARGDERIVATIONS($\tau_{next}, \tau_{src}$)
end function
```

Algorithm 3 The handler. We assume that the function SAMPLEFROMSEED is the top-level function for the sampler. It takes a ULF as input and runs the sampler to produce a single new (ULF, English) pair.

```
function AUGMENTDATASET($\mathscr{D}$, $d$, $b$, $F$)
    input: $\mathscr{D}$, a set of (ULF, English) pairs; $d$, the branching depth; $b$, the branching factor;
    $F$, top fraction of augmented set to keep.
    $\mathscr{D}' \leftarrow \mathscr{D}$
    for $(U, E) \in \mathscr{D}$ do
        $S \leftarrow [U]$
        for $i \in \{1, 2, \ldots, d\}$ do
            $S' \leftarrow \emptyset$
            for $U \in S$ do
                $U' \leftarrow$ POPFIRST$(S)$
                for $j \in \{1, 2, \ldots, b\}$ do
                    $(U'', E'') \leftarrow$ SAMPLEFROMSEED$(U')$
                    $\mathscr{D}'$.append($(U'', E'')$)
                    $S'$.append($U'$)
                end for
            end for
            $S \leftarrow S'$
        end for
    end for
    ORDERBYLANGUAGEMODELSCORE($\mathscr{D}'$)
    return first $F * |\mathscr{D}|$ elements of $\mathscr{D}'$
end function
```

Meaning-Text Theory within Abstract Categorial Grammars: Towards Paraphrase and Lexical Function Modeling for Text Generation

Marie Cousin
Université de Lorraine, CNRS, Inria, LORIA / F-54000 Nancy, France
marie.cousin@loria.fr

Abstract

The meaning-text theory is a linguistic theory aiming to describe the correspondence between the meaning and the surface form of an utterance with a formal device simulating the linguistic activity of a native speaker. We implement a version of a model of this theory with abstract categorial grammars, a grammatical formalism based on λ-calculus. This implementation covers the syntax-semantic interface of the meaning-text theory, i.e., not only the three semantic, deep-syntactic and surface-syntactic representation levels of the theory, but also their interface (i.e., the transformation from one level to another). This implementation hinges upon abstract categorial grammars composition in order to encode level interfaces as transduction operate.

1 Introduction

We present in this article our implementation of a model of the meaning-text theory (MTT, Mel'čuk et al., 2012; Milićević, 2006) with abstract categorial grammars (ACG, de Groote, 2001), focusing on the meaning to text direction, i.e., generation. MTT is a linguistic theory that has already been used in a generation context, while ACGs are a grammatical framework known to encode a range of various grammatical formalisms.

MTT aims to describe the link between the meaning and the textual representation of an utterance. This description is made possible thanks to a formal device, a meaning-text model (MTM), that simulates the linguistic activity of a native speaker. It uses, among others, a dependency syntax and the key concepts of paraphrase and lexical functions (LF) (see Section 2.2). The latter enables a text to be more natural: the syntagmatic LFs for instance encode collocations or support verbs. They play an important role, especially for surface representations, and when producing text. Some formalisations and implementations focusing on text generation already exist, such as GUST (Kahane and Lareau, 2005) or MARQUIS (Wanner et al., 2010).

ACGs are a grammatical framework based on λ-calculus. They enable the implementation of other grammatical formalisms, and have many advantages. ACGs are reversible. We can therefore use them in generation or analysis (Kanazawa, 2007). Their capacities to encode other formalisms, such as tree adjoining grammars (TAG, Pogodalla (2017a)), and to be used in generation were employed to generalize the G-TAG model (Danlos et al., 2014). They are also currently used in an industrial context by Yseop. We aim in this article to use these properties with another linguistic theory that was already proven usefull for generation systems: MTT.

We may now wonder if ACGs are a relevant tool to implement a linguistic theory based on a dependency syntax by assessing it on a text generation task. This article presents the feasibility of such an implementation, based on a restricted number of examples which illustrate several specificities of MTT. As a grammatical framework, ACGs can provide the grammatical formalisms they encode with their generic algorithms, making it unnecessary to develop and implement specific information. They also offer a reversible encoding so that, for instance, we get here both synthesis and analysis for MTT.

Because we wish to have a fined-grained control over the generated text, we choose to focus on text generation with formal methods. The same motivations can be found in Grammatical Framework (Ranta, 2004), that use a type-theoretic system too. The encoding and links to other formalisms in ACGs have been well studied. Indeed, Table 1 below highlights the expressive power of ACGs. A hierarchy of ACGs is used in this table, based

Proceedings of the 15th International Conference on Computational Semantics pages 134–143
June 21–23, 2023. ©2023 Association for Computational Linguistics

on two notions (order and complexity of an ACG) which are defined in Section 3.

ACG	generated language
$\text{ACG}_{(1,n)}$	finite languages
$\text{ACG}_{(2,1)}$	regular languages
$\text{ACG}_{(2,2)}$	context-free grammars (CFG)
$\text{ACG}_{(2,3)}$	well-nested multiple CFG
$\text{ACG}_{(2,4)}$	mildly context-sensitive grammars
$\text{ACG}_{(2,4+n)}$	$\text{ACG}_{(2,4)}$
$\text{ACG}_{(3,n)}$	MELL decidability

Table 1: Expressive power of ACGs (Pogodalla, 2017b).

We aim at encoding a MTM within ACGs, and especially the linguistic structures used by MTT, even thought other formalisms close to the ones used by MTT exist. MTT uses graphs for its semantic representation for instance, to represent predicate-arguments structures, and is therefore close to AMR (Banarescu et al., 2013). Regarding the deep-syntactic representation, MTT uses dependency trees with labels that can differ from other dependency formalisms. We are here interested by how the linguistic structures of MTT relate to each other and the unity of the whole.

2 Meaning-text theory, lexical functions and paraphrase

2.1 MTT

MTT (Mel'čuk et al., 2012; Milićević, 2006) describes the meaning-text correspondance of an utterance. The meaning is "a linguistic content to be communicated" (Milićević, 2006), and is not directly observable, while the text is "any fragment of speech" (Milićević, 2006), immediately perceptible.

MTM consists of 7 representation levels (see Figure 1): the semantic (SemR), deep-syntactic (DSyntR), surface-syntactic (SSyntR), deep-morphologic (DMorphR), surface-morphologic (SMorphR), deep-phonetic (DPhonR) and surface-phonetic (SPhonR) representation levels.

It also has 6 transition modules between these levels. Between each pair of (neighbor) representation levels lay one module, which performs the transition from one of these adjacent representation levels to the other one. They take as input one representation of the former level, perform the transition, and output the obtained representations of the next level. For example, if we give the semantic module (between the semantic level and the deep-syntactic level) the SemR represented Figure 6a as an input, it will output the DSyntR represented Figure 6b.

On top of this transition, they may also perform paraphrase steps, that are transformations inside the same level. It is the case of the deep-syntactic module (between the deep-syntactic and surface-syntactic levels) that performs deep-syntactic paraphrasing and LFs realization on top of DSyntR to SSyntR structure transition.

Thus, depending on the chosen direction, the MTM enables the generation (from SemR to SPhonR) or the analysis (from SPhonR to SemR) of an utterance (see Figure 1).

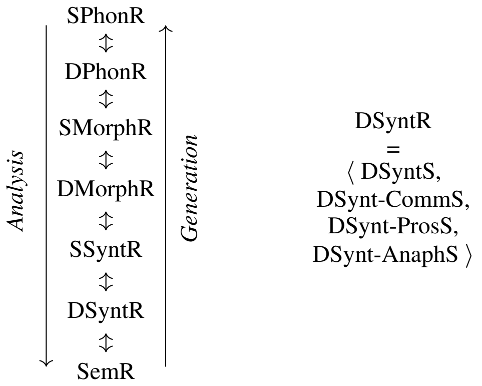


Figure 1: Schema of the MTM and detail of the substructures of DSyntR (Mel'čuk et al., 2012).

Each one of these representations has substructures: one main structure and other structures that add information to the main one.

In this article and our implementation we use mainly DSyntS (the main structure of DSyntR), and not the other three substructures of DSyntR. Therefore, we will not give further detail about them, and only work with DSyntS. The same applies to the substructures of the other representation levels.

- SemS is the main structure of SemR. It is a directed graph whose nodes are semantemes and arcs are labeled with numbers (starting with 1). For each semantic predicate, the numbers on the arcs leading to its arguments indicate their order (see Figure 6a).

- DSyntS is the main structure of DSyntR (see Figure 6b). It is a dependency tree, whose nodes are deep-syntactic lexemes and branches are labelled with deep-syntactic relations. Deep-syntactic relations include actantial relations (labelled from I to VII), attributive relations (labelled ATTR and ATTR_{descr}), other subordinate relations (labelled APPEND) and coordinative relations (labelled COORD and QUASI-COORD) relations.

- SSyntS is the main structure of SSyntR. It is also a dependency tree, whose nodes are surface-syntactic lexemes, and branches are labelled with surface-syntactic relations (see Figure 6c for an example).
- The main structures of all other levels are represented by strings.

Regarding DSyntR (see Figure 1), its substructures are the deep-syntactic structure (DSyntS), deep-syntactic communicative structure (DSynt-CommS), deep-syntactic prosodic structure (DSynt-ProsS) and deep-syntactic anaphoric structure (DSynt-AnaphS). Mel'čuk et al. (2013) give further detail about the construction rules of such a structure. DSynt-CommS consists of markers of communicative opposition, such as the theme of the DSyntR (see Milićević (2006), page 15 where an example is detailed). DSynt-ProsS consists of a set of markers of prosodies, such as "declarative" or "ironic" for instance. DSynt-AnaphS contains the links of co-referentiality between the node of a DSyntS.

We focus in this article on the deep-syntactic module, more precisely on the LFs realization.

2.2 LF and paraphrase

In the transition modules as well as at some representation levels MTT uses the key concepts of paraphrase and LFs.

LFs (Mel'Čuk and Polguère, 2021) aim at describing linguistic phenomena that exist in all languages. Indeed, they are functions describing relations between lexical units (LU). They associate with that LU the set of all other alternative choices of LUs consistent with the relation they describe. They hinge on semantics, syntax and morphology. That means that they are part of the lexicon of the language, as well as part of its grammar. They are used in MTT in the explanatory combinatorial dictionary (we will not describe that part, see Mel'čuk et al. (2012, 2013) for further detail), and to perform linguistic paraphrase.

LFs are classified in two main categories: paradigmatic and syntagmatic LFs. The former ones express relations of semantic derivation between LUs, while the latter express the combinatorial properties of LUs. For instance, `anti` is a paradigmatic LF associating a LU with its contrary: `anti`(CALM) = {UPSET, RESTLESS} (based on Mel'čuk et al. (2013)) and `causFunc` is a syntagmatic LF which associates a LU with a support verb meaning *make it exist*: `causFunc`(ATTENTION) = DRAW (in the expression "*to draw attention*") (Milićević, 2006). LFs are useful to encode lexical phenomena such as collocations or support verbs. Further detail is given in Mel'Čuk and Polguère (2021).

As for the paraphrase, there are different kinds of paraphrases that can occur at different levels:

(a) at the semantic level with the definition of semantemes by simpler semantemes,

(b) at the deep-syntactic level with the transformation of the dependency tree into another one thanks to lexical paraphrasing rules and restructuring paraphrasing rules (that supports the lexical ones) (Mel'čuk et al., 2013) making some LFs appear (cf. Figures 2 and 3),

(c) between the deep-syntactic and surface-syntactic levels, when choosing how to realize a LF when more than one value of LU is possible (cf. Figures 2 and 3),

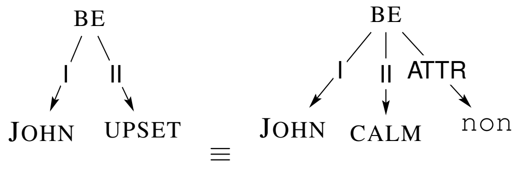
(d) at the surface-syntactic level.

Both types (b) and (c) of paraphrase are often considered to be part of the deep-syntactic paraphrase. We want here to make a distinction between what we call deep-syntactic paraphrase (i.e., type (b)) and what we call LF realization (i.e., type (c)), even if both of them occur in the deep-syntactic module.

Iordanskaja et al. (1991) gives further detail on the different paraphrase types. We will here evoke mechanisms to encode the deep-syntactic paraphrase (type (b)) and highlight the one concerning LFs (type (c)). Figure 2 shows an example of paraphrase that uses both types (b) and (c), i.e., deep-syntactic paraphrase and LF realization (it will be further explained in Section 5).

3 Abstract categorial grammars

ACGs (de Groote (2001), whose definitions we use here) are a grammatical formalism based on λ-calculus. An ACG is composed of two languages, linked together by a lexicon. The first langage is called the abstract language and is the set of abstract grammatical structures, such as analysis



(a) "*John is upset.*" (b) "*John is not calm.*"

Figure 2: Representation of two deep-syntactic structures representing the paraphrase of "*John is upset*". That paraphrase uses lexical and restructuration deep-syntactic rules as well as the relation `anti`(UPSET) = CALM (based on Mel'čuk et al. (2013)). The dependency tree obtained between deep-syntactic paraphrase and LF realization is given in figure 3a.

trees. The second one, called the object language, is the set of the surface representations generated by the abstract language, such as strings or logical representations in the form of a graph. Each one of these languages is a set of λ-terms obtained by induction over a signature.

Definition 1 *Let A be a set of atomic types. $\mathcal{T}(A)$ is the set of* ***linear implicative types****, obtained inductively over A:*

- *if $a \in A$ then $a \in \mathcal{T}(A)$*
- *if $\alpha, \beta \in \mathcal{T}(A)$ then $(\alpha \to \beta) \in \mathcal{T}(A)$*

Definition 2 *Let Σ be a* ***higher order signature****. Σ is of the form $\Sigma = \langle\, A,\ C,\ \tau\,\rangle$, where:*

- *A is a set of atomic types,*
- *C a set of constants,*
- *$\tau : C \longrightarrow \mathcal{T}(A)$ a function.*

We express with $\vdash_{\Sigma_1} t : s$ that the type of a λ-term t is s in the signature Σ (or $t : s$ if there is no ambiguity).

We express $\Lambda(\Sigma)$ the set of λ-terms obtained using the constants of C, the variables, the abstractions and the applications.

Definition 3 *Let Σ_1 and Σ_2 be two signatures. A* ***lexicon*** *$\mathcal{L}_{12}$ from Σ_1 to Σ_2 is a pair of morphisms $\langle F, G\rangle$ such that $F : \tau(A_1) \longrightarrow \tau(A_2)$ and $G : \Lambda(\Sigma_1) \longrightarrow \Lambda(\Sigma_2)$.*

We write $\mathcal{L}_{12}(t) = \gamma$ to express that γ is the interpretation of t by $\mathcal{L}_{12}$ (or $t := \gamma$ if there is no ambiguity on the used lexicon).

The signatures (like Σ_{dsynt_tree}, see Figure 2) we describe here use almost linear λ-terms. We will not explain this notion, for it is not of great interest for what we say (the used variables being neither discarded nor duplicated in the lexicons). Nevertheless we use the notation λ^o and λ for the linear and non-linear abstractions respectively.

Moreover, an interesting property of ACGs is that when they are second order almost linear, the morphisms inversions (see below) are decidable in a polynomial time (Salvati, 2005), and when they are not almost linear, they remain decidable, even though the complexity is not polynomial anymore (Salvati, 2010). We therefore were careful to use as much as possible second order almost linear ACGs in this implementation.

We may thereby define Σ_{dsynt_tree} in Figure 2 that corresponds to the DSyntR level. Indeed, the constants of this signature enable to build the deep-syntactic trees of DSyntS, like the ones in Figure 3.

- $A_{dsynt\_tree} = \{T, rel, l\}$,
- $C_{dsynt\_tree} = \{c^{dt}_{John},\ c^{dt}_{be},\ c^{dt}_{restless},\ c^{dt}_{upset},\ c^{dt}_{calm},\ A_1,\ A_2,\ ATTR,\ lex_0,\ lex_2,\ lex_3,\ \texttt{Anti},\ \texttt{Non}\}$,
- τ_{dsynt_tree} is given by Table 2 below.

Constant		Type
c^{dt}_{John}	:	l
c^{dt}_{be}	:	l
$c^{dt}_{restless}$	:	l
c^{dt}_{upset}	:	l
c^{dt}_{calm}	:	l
A_1	:	rel
A_2	:	rel
$ATTR$	:	rel
lex_0	:	$l \to T$
lex_2	:	$l \to rel \to T \to rel \to T \to T$
lex_3	:	$l \to rel \to T \to rel \to T \to rel \to T \to T$
`Anti`	:	$l \to l$
`Non`	:	T

Table 2: τ_{dsynt_tree}

In Table 2, all the constants of the form c^X_L encode LUs, while all constants of the form A_i and lex_i encode the dependency tree structure. The constants `anti` and `non` encode the eponyms LFs.

Due to space considerations, we will not define A and C in the following paragraphs and sections anymore, they can be deduced from the table representing τ. We may now define the notions of ACG, abstract and object languages:

Definition 4 *An* ***abstract categorial grammar*** *is a tuple $\mathcal{G} = \langle\Sigma_1, \Sigma_2, \mathcal{L}_{12}, s\rangle$ where:*

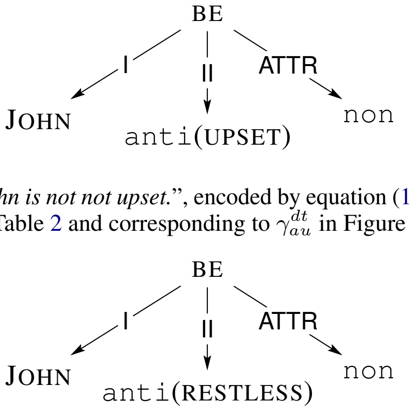
- $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ *and* $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ *are two higher order signatures,*
- $\mathcal{L}_{12} = \Sigma_1 \longrightarrow \Sigma_2$ *is the lexicon,*
- $s \in \mathcal{T}(A_1)$ *is the distinguished type of the grammar.*

Definition 5 *The **abstract language** $\mathcal{A}$ and the **object language** $\mathcal{O}$ of an ACG $\mathcal{G} = \langle \Sigma_1, \Sigma_2, \mathcal{L}_{12}, s \rangle$ are:*

- $\mathcal{A} = \{t \in \Lambda(\Sigma_1) | \vdash_{\Sigma_1} t : s$ *is derivable*$\}$
- $\mathcal{O} = \{t \in \Lambda(\Sigma_2) | \exists u \in \mathcal{A}(\mathcal{G})$ *such that* $t = \mathcal{L}_{12}(u)\}$

In this article we use $\beta\eta$-equivalence as equality between λ-terms.

Σ_{dsynt_tree} illustrated above gives the needed constants to build the dependency trees of Figure 3.



(a) "*John is not not upset.*", encoded by equation (1) below using Table 2 and corresponding to γ^{dt}_{au} in Figure 4.

(b) "*John is not not restless.*", encoded by equation (2) below using Table 2 and corresponding to γ^{dt}_{ar} in Figure 4.

Figure 3: Representation of two deep-syntactic structures representing the deep-syntactic paraphrases of "*John is not calm*" (based on Mel'čuk et al. (2013)).

But, in order to do so with an ACG, we need another signature as well as a lexicon: we now introduce the abstract signature $\Sigma_{deep\_syntactic}$ (see Table 3), to define $\mathcal{L}_{dsyntRel}$ (see Table 4). $\langle \Sigma_{deep\_syntactic}, \Sigma_{dsynt\_tree}, \mathcal{L}_{dsyntRel}, \mathrm{G} \rangle$ is the ACG that builds the dependency trees of DSyntS in Σ_{dsynt_tree} (see the articulation of these signatures in Figure 5).

We define the notion of order and complexity of an ACG as well. They are used in Table 1 which describes the expressive power of ACGs.

Constant		Type
c^{ds}_{John}	:	G
c^{ds}_{be}	:	$G \rightarrow G \rightarrow G$
c^{ds}_{upset}	:	G
$c^{ds}_{restless}$	:	G
c^{ds}_{calm}	:	G

Table 3: $\tau_{deep\_syntactic}$

$\Sigma_{deep\_syntactic}$		Σ_{dsynt_tree}
G	:=	T
c^{ds}_{John}	:=	$lex_0\ c^{dt}_{John}$
c^{ds}_{be}	:=	λ^o X Y. $lex_2\ c^{dt}_{be}\ A_1$ X lex_2 Y
c^{ds}_{upset}	:=	$lex_0\ c^{dt}_{upset}$
$c^{ds}_{restless}$	:=	$lex_0\ c^{dt}_{restless}$
c^{ds}_{calm}	:=	$lex_0\ c^{dt}_{calm}$

Table 4: $\mathcal{L}_{dsyntRel}$

Definition 6 (Pogodalla, 2017b) *The **order** of an ACG is the maximum of the order of its abstract constants. The order of an abstract constant is the order of its type τ. The order of a type $\tau \in \mathcal{T}(A)$ is inductively defined:*

- $order(\tau) = 1$ *if* $\tau \in A$,
- $order(\alpha \rightarrow \beta)$
  $= max(1 + order(\alpha), order(\beta))$ *else.*

*The **complexity** of an ACG is the maximum of the orders of its atomic types realizations. An ACG of order γ and of complexity η is written ACG$_{(\gamma,\eta)}$.*

In the following paragraphs and sections, we use the notation c^X_L for the constant of Σ_X encoding the LU L. If there are two different constants representing the same LU L in Σ_X, we write c^X_{L1} and c^X_{L2} to distinguish them. We also use γ^X_i for the complex λ-term of Σ_X indexed by i. These complex λ-terms being the encoding of possible representations for an expression, the index i indicate this expression. Therefore, we will use au for "*John is not not upset*" (cf. Figure 3a), ar for "*John is not not restless*" (cf. Figure 3b), and $c1$, $c2$ for "*John is not calm*".

We use dt and $d0f$ instead of $dsynt\_tree$ and $dsynt\_0\_fl$.

That being said, we define the complex λ-terms:

$$\gamma^{dt}_{au} = lex_3\ c^{dt}_{be}\ A_1\ (lex_0\ c^{dt}_{John})\ A_2\ (lex_0(\texttt{anti}\ c^{dt}_{upset}))\ ATTR\ c^{dt}_{non} \quad (1)$$

$$\gamma^{dt}_{ar} = lex_3\ c^{dt}_{be}\ A_1\ (lex_0\ c^{dt}_{John})\ A_2\ (lex_0(\texttt{anti}\ c^{dt}_{restless}))\ ATTR\ c^{dt}_{non} \quad (2)$$

$$\gamma^{fl}_{c1} = lex_3\ c^{fl}_{be}\ A_1\ (lex_0\ c^{fl}_{John})\ A_2\ (lex_0\ c^{fl}_{calm1})\ ATTR\ c^{fl}_{non} \quad (3)$$

$$\gamma^{fl}_{c2} = lex_3\ c^{fl}_{be}\ A_1\ (lex_0\ c^{fl}_{John})\ A_2\ (lex_0\ c^{fl}_{calm2})\ ATTR\ c^{fl}_{non} \tag{4}$$

$$\gamma^{d0f}_{c} = lex_3\ c^{d0f}_{be}\ A_1\ (lex_0\ c^{d0f}_{John})\ A_2\ (lex_0\ c^{d0f}_{calm})\ ATTR\ c^{d0f}_{non} \tag{5}$$

γ^{dt}_{au} encodes the upper tree of Figure 3 while γ^{dt}_{ar} encodes the lower tree of Figure 3.

Another advantageous property of ACGs is their ability to be composed. A specific case of composition is transduction: given two ACGs sharing the same abstract signature, transduction (see Figure 4) is the composition of the analysis (or the inversion) of a morphism (like $\mathcal{L}_{lexfl}{}^{-1}$) and the application of a morphism (as $\mathcal{L}_{reducefl}$) using both ACGs. Consequently, transduction is very useful since it enables two terms of two object signatures to be in relation with each other.

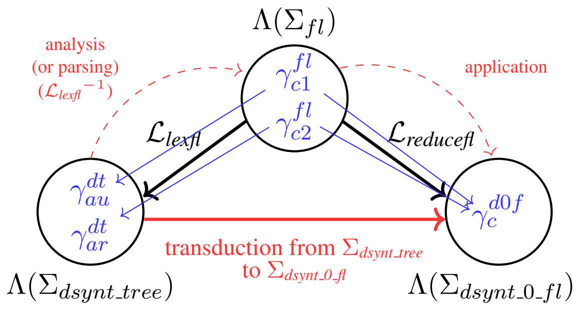


Figure 4: Transduction from DSyntR to a deep-syntactic representation where the LFs would be realized. $\mathcal{L}_{reducefl}$ is such that $\mathcal{L}_{reducefl}(\gamma^{fl}_{c1}) = \mathcal{L}_{reducefl}(\gamma^{fl}_{c2})$.

It is indeed a method to make possible the transition between the representation levels (represented here by signatures) and perform the transformation of the structures: given an initial λ-term, transduction gives a second λ-term, without modifying the first one. MTT being like a suite of structure transformations (see Figure 1), transduction seems well adapted to implement a MTM. This suite of structure transformations also appears in the overview of the ACG architecture of our implementation in Figure 5 presented in Section 4, especially between the areas 1, 2, 4, and 5 of Figure 5.

Moreover, we can produce a lot of structures inside a signature. But, they do not all have an antecedent in the abstract signature. Indeed, when parsing a structure of an object signature of an ACG, if it has an antecedent, it will be found (for parsing is decidable, see above). If it does not have one, then nothing happens. That means that, when applying transduction between two object signatures (or two representation levels, in the case of our implementation), if one structure should not have a correspondance in the next object signature, it will not have one: no new structure will be produced.

4 Overview of our implementation

In order to represent the MTM, at least from SemR to SSyntR, we implemented signatures and lexicons (see Figure 5) and experimented our encoding with ACGtk (Pogodalla, 2016), a piece of software allowing for defining grammars and using the associated parsing and interpretation operation. Nevertheless, for simplification purpose we did not implement the other substructures than the main one for each representation level, like the communicatives structures (see Section 2).

This implementation uses transduction (see Figure 4) which is the heart of our implementation, for it is used to perform all transformations (see Figure 5, where we can guess the use of transduction).

We can see that the third area (see Figure 5) looks like a detour. That is due to the fact that, on one hand, MTT does not modify a structure when going from a representation level to another (which transduction also does), and, on the other hand, during some steps of the generation process, we want to keep the former structures as well as the newly obtained ones. That is the case of the deep-syntactic paraphrasing, for we want to keep all possible dependency trees that would lead to a sentence, so all the possible paraphrases. In other words, MTT makes other structures appear inside of one representation level, and we want them all to reach the next representation level, not only the last one. For this is not the goal of transduction, deep-syntactic paraphrasing is hard to perform in the current state of our implementation, and is performed by this third area, although it is problematic, as we will explain later.

Our implementation encodes the different levels of representation of a MTM (see Figure 5): Σ_{sem} corresponds to SemR, Σ_{dsynt_tree} to DSyntR, and Σ_{ssynt_tree} to SSyntR. As we can see, we use the transduction between:

- $\Sigma_{semantic}$ and Σ_{dsynt_tree}: to perform the semantic paraphrasing and to make the transition from SemR to DSyntR (areas 1 and 2),
- Σ_{dsynt_tree} and Σ_{dsynt_rule}: to perform the deep-syntactic paraphrasing (area 3),

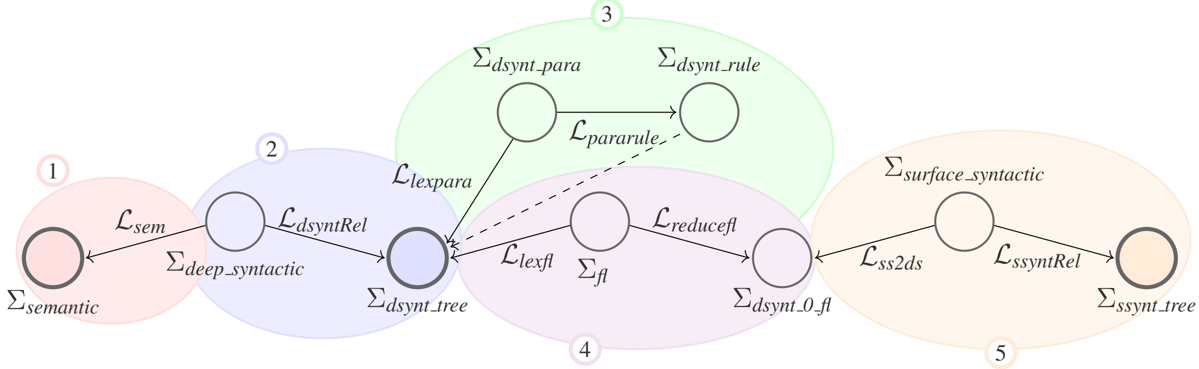


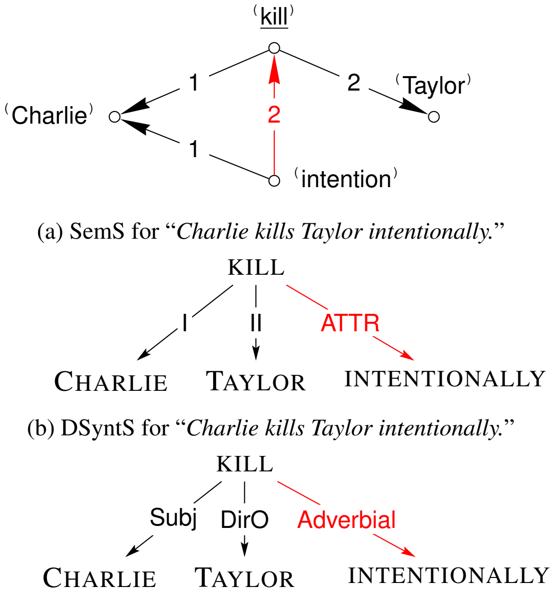
Figure 5: Overview of the ACG architecture. Area 1 corresponds to the semantic paraphrasing, area 2 to the transition between SemR and DSyntR, area 3 to the deep-syntactic paraphrasing, area 4 to the LFs realization step, and area 5 to the transition to SSyntR.

- Σ_{dsynt_tree} and $\Sigma_{dsynt\_0\_fl}$: to realize LFs (area 4, this part will be detailed in Section 5),
- $\Sigma_{dsynt\_0\_fl}$ and Σ_{ssynt_tree}: to make the transition from a deep-syntactic representation without LFs anymore to SSyntR.

It was tested on a sample of example sentences. This sample is short, but covers many lexical phenomena, like collocations, the use and realization of LFs, semantic or syntactic equivalences, as well as obligatory arguments optionally expressible. As stated at the end of Section 3, inside a representation level (or a signature), the structures that should not have a correspondance in the next representation level (because they are incorrect for example) do not have one: they will not find an antecedent by reversing the lexicon during transduction (Table 5 highlights this). If a structure should not lead to another one regarding MTT formalism, then our implementation of ACGs is such that, by transduction, no antecedent will be found.

The code and the examples are available at https://inria.hal.science/hal-04104453.

On top of that, this implementation also deals with adverbial groups, that have a specific treatment inspired by the work on TAG of Pogodalla (2017a). Their treatment is indeed not the same depending on the signature we look at. In SemR, i.e., in $\Lambda(\Sigma_{semantic})$, the arc between the adverbial group and the verb it modifies is directed toward the verb, while in DSyntR, so $\Lambda(\Sigma_{dsynt\_tree})$, it is directed toward the adverbial group. This is because adverbial groups are modifiers (Mel'čuk et al., 2013, 2015). We used the same approach as Candito and Kahane (1998) for the dependency inversions between derived trees and derivation trees in TAG (see Figure 6). The manipulation of obligatory arguments of a SemR that are optionally expressible is also possible, and was inspired by Blom et al. (2011). It is done thanks to some constants in $\Sigma_{deep\_syntactic}$. These last two points will not be detailed here, but you can find further detail in (Cousin, 2022). Nevertheless, the Section 5 will explain the last step of the paraphrase illustrated on Figure 2, that is the realization of LFs. It is a good example to show the different kinds of possible equivalence inside an ACG, as well as how LFs are used in this modeling.



(a) SemS for "*Charlie kills Taylor intentionally.*"

(b) DSyntS for "*Charlie kills Taylor intentionally.*"

(c) SSyntS for "*Charlie kills Taylor intentionally.*"

Figure 6: Dependency inversion for the adverb "*intentionally*" in "*Charlie kills Taylor intentionally*".

Expression linked to the semantic graph	"*Charlie kills Taylor intentionally*"	"*John is calm*"
SemR ($\Sigma_{semantic}$)	1	1
DSyntR before deep-syntactic paraphrasing ($\Sigma_{dsynt\text{-}tree}$)	2	1
of which will be accepted at the next stage	2	1
DSyntR after deep-syntactic paraphrasing ($\Sigma_{dsynt\text{-}tree}$)	2	6[1]
of which will be accepted at the next stage	2	3
DSyntR after FLs realization ($\Sigma_{dsynt\text{-}0\text{-}fl}$)	2	5
of which will be accepted at the next stage	2	3
SSyntR ($\Sigma_{ssynt\text{-}tree}$)	2	3

Table 5: Number of obtained structures by generation step for two initial semantic graphs, corresponding to the expressions "*Charlie kills Taylor intentionally*" and "*John is calm*".

5 Example

This section gives a detailed example on how the transduction works and how we realize LFs in our implementation. We consider here the signatures Σ_{dsynt_tree} (see Table 2), Σ_{fl} and $\Sigma_{dsynt\_0\_fl}$ only, as well as the lexicons $\mathcal{L}_{lexfl}$ and $\mathcal{L}_{reducefl}$. We saw in Figure 2 a paraphrase example using deep-syntactic paraphrase as well as LF realization. We will explain the LFs realization in this section.

We consider the following sentences:

(6) a. "*John is upset*"

b. "*John is not calm*"

c. "*John is restless*"

We consider the example of the paraphrase between expressions (6a) and (6b). We may remember that `anti`(CALM)={UPSET, RESTLESS}, so we also have `anti`(UPSET) = CALM = `anti`(RESTLESS) (among other values, but we are interested in CALM here). After the deep-syntactic paraphrasing of sentence (6a), before realizing LFs, we have a dependency tree such as Figure 3a. But, (6a) is a paraphrase of (6b), and so is (6c) (illustrated in Figure3b). They both have the same paraphrase (6b), so these two sentences (6a) and (6c) are paraphrases of each other themselves (depending on the context, but that point will be explained in the conclusion).

Therefore, we want for our implementation to allow this link, this equivalence between the expressions (6c) and (6a). Thus, we want to obtain an equivalence such as (7):

$$\gamma_{au}^{dt} \equiv \gamma_{ar}^{dt} \qquad (7)$$

Indeed, we may remember that γ_{au}^{dt} represents the DSyntS of expression (6a), and γ_{ar}^{dt} the DSyntS of expression (6c). Because both expressions are paraphrases of (6b) and because two different images of a morphism cannot have the same antecedent, we will have to use transduction here. The equivalence between (6a) (or (6c)) and (6b) will take two steps, i.e., parsing and application. We want, if we write $\mathcal{T}$ for the transduction relation, our implementation to allow equations such as (8):

$$\mathcal{T}(\gamma_{au}^{dt}, \gamma_{c}^{d0f}) \text{ and } \mathcal{T}(\gamma_{ar}^{dt}, \gamma_{c}^{d0f}) \qquad (8)$$

Hence, we want to use two ACGs sharing the same abstract signature, to have the following equations (9), (10) and (11) (see Figure 4 that illustrates it) in order to have equations (8) and (7):

$$\mathcal{L}_{reducefl}(\gamma_{c1}^{fl}) = \gamma_{c}^{d0f} = \mathcal{L}_{reducefl}(\gamma_{c2}^{fl}) \qquad (9)$$

$$\mathcal{L}_{lexfl}(\gamma_{c1}^{fl}) = \gamma_{au}^{dt} \qquad (10)$$

$$\mathcal{L}_{lexfl}(\gamma_{c2}^{fl}) = \gamma_{ar}^{dt} \qquad (11)$$

Tables 2, 6a, 6b and 6c define the constants of the signatures and lexicon we use in this section in order to do so. They are simplified and show only relevant information for this example. The constants such as lex_i and A_i (see Section 3, Table 2) are not specified anymore, for they do not change from one signature to another.

We implement the realization of LFs with the transduction and the properties of λ-calculus, like β-reduction. Indeed, one expression may have different representations in Σ_{dsynt_tree} (see Table 2), but only one in $\Sigma_{dsynt\_0\_fl}$ (see Table 6a). To realize LFs, we use different levels of equivalencies, that this example highlights.

The different levels of equivalencies are the following (see Figure 4):

- inside of a signature, and by application or parsing of a lexicon, two representations may

[1] In fact, more structures are obtained, but they are incorrect by construction. They will therefore not be considered here (and do not have an antecedent in $\Sigma_{surface\text{-}syntactic}$).

Constant		Type	Constant		Type
c^{fl}_{John}	:	l	c^{d0f}_{John}	:	l
c^{fl}_{be}	:	l	c^{d0f}_{be}	:	l
c^{fl}_{calm0}	:	l	c^{d0f}_{calm}	:	l
c^{fl}_{calm1}	:	l	Non	:	t
c^{fl}_{calm2}	:	l			
Non	:	t			

(a) τ_{fl} (left) and $\tau_{dsynt\_0\_fl}$ (right)

Σ_{fl}		Σ_{dsynt_tree}
$t := T, r := rel, l := l$		
c^{fl}_{John}	:=	c^{dt}_{John}
c^{fl}_{be}	:=	c^{dt}_{be}
c^{fl}_{calm0}	:=	c^{dt}_{calm}
c^{fl}_{calm1}	:=	$\mathtt{anti}(c^{dt}_{upset})$
c^{fl}_{calm2}	:=	$\mathtt{anti}(c^{dt}_{restless})$
Non	:=	Non

(b) $\mathcal{L}_{lexfl}$

Σ_{fl}		$\Sigma_{dsynt\_0\_fl}$
$t := t, r := r, l := l$		
c^{fl}_{John}	:=	c^{d0f}_{John}
c^{fl}_{be}	:=	c^{d0f}_{be}
c^{fl}_{calm0}	:=	c^{d0f}_{calm}
c^{fl}_{calm1}	:=	c^{d0f}_{calm}
c^{fl}_{calm2}	:=	c^{d0f}_{calm}
Non	:=	Non

(c) $\mathcal{L}_{reducefl}$

Table 6: τ_{fl}, $\tau_{dsynt\_0\_fl}$, $\mathcal{L}_{lexfl}$ and $\mathcal{L}_{reducefl}$.

be equal thanks to β-reduction. Nevertheless, this example does not show it. This β-equivalence inside of a signature is used in $\Sigma_{semantic}$ but not illustrated in this article due to space restrictions.

- by parsing a lexicon, for instance $\mathcal{L}_{reducefl}$: the sentence "*John is not calm*" has one representation in $\Sigma_{dsynt\_0\_fl}$, while it has two different ones in Σ_{fl}. Thus, we obtain thanks to the parsing the equality (9).

- by transduction: the two dependency trees γ^{dt}_{au} and γ^{dt}_{ar} in Σ_{dsynt_tree} are equivalent. Indeed, when parsing and applying the lexicons $\mathcal{L}_{reducefl}$ and $\mathcal{L}_{lexfl}$, we obtain (as wanted) the equations (9), (10) and (11), then (8) and finally (7) by transduction. Equations (10) and (11) show that anti(UPSET) and anti(RESTLESS) will be realized as CALM. Because one antecedent cannot have two different images, we need the second object signature $\Sigma_{dsynt\_0\_fl}$ in order to have one unique constant per lexeme (here CALM). Hence we have in $\Sigma_{dsynt\_0\_fl}$ deep-syntactic dependency trees where LFs are realized.

Thus transduction between signatures Σ_{dsynt_tree} and $\Sigma_{dsynt\_0\_fl}$ allows to realize LFs, and to perform the third type of paraphrase (see Section 2).

6 Conclusion and future work

We have shown a possible implementation of a MTM with ACGs. This implementation models the SemR to SSyntR levels of MTM. Even though this implementation uses only the main structures of the representation levels of MTT and not the other substructures (like the communicatives ones), when tested over a sample of example sentences, their SSyntS are correctly obtained (see Table 5). Indeed, for a given representation level, in the direction of the generation, the incorrect structures are not produced, for they do not have an antecedent by parsing the lexicon to the next abstract signature. Moreover, if we take the direction of analysis, we obtain the wanted semantic graphs.

The implemented model enables the semantic paraphrase to take place, as well as the transitions between the representation levels thanks to transduction, the realization of LFs thanks to transduction too, the handling of obligatory semantic arguments optionally expressible, and the handling of adverbial groups.

However, this implementation has some limitations. Indeed, the deep-syntactic paraphrasing is not optimal. It is actually made possible by the detour of area 3 in Figure 5, but we need to manually iterate the process. We need to save the previous structures, perform the paraphrasing loop, and apply again the process for the newly generated structures until no new correct structure is obtained. Transduction is showing some limitations here: this mechanism is not well suited for the deep-syntactic paraphrasing because rewritten structures still need to be processed as well as resulting structures.

Furthermore, we have not exploited all the possible types of paraphrasing (Iordanskaja et al., 1991) in our implementation yet: we also want to continue in this direction to implement all of them. Moreover, we want to include, for each level, other substructures, such as the communicative structures, to have more information about the theme, the rheme, and the speaker intentions, in order to have an implementation nearer to MTT than what it currently is.

Acknowledgments

I would like to thank Sylvain Pogodalla and the reviewers for their comments and constructive remarks, which helped to improve this article. I would also like to thank my two thesis supervisors, Philippe de Groote and Sylvain Pogodalla, for their supervision and our exchanges which enabled me to write this article.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Chris Blom, Philippe de Groote, yoad Winter, and Joost Zwarts. 2011. Implicit Arguments: Event Modification or Option Type Categories? In *18th Amsterdam Colloquium on Logic, Language and Meaning*, volume 7218 of *Lecture Notes in Computer Science*, pages 240–250, Amsterdam, Netherlands. Springer.

Marie-Hélène Candito and Sylvain Kahane. 1998. Can the TAG derivation tree represent a semantic graph? an answer in the light of meaning-text theory. In *Proceedings of the Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*, pages 21–24, University of Pennsylvania. Institute for Research in Cognitive Science.

Marie Cousin. 2022. Génération de texte avec les grammaires catégorielles abstraites et la théorie sens-texte. Master's thesis, Grenoble INP Ensimag, September.

Laurence Danlos, Aleksandre Maskharashvili, and Sylvain Pogodalla. 2014. Text generation: Reexamining G-TAG with abstract categorial grammars (génération de textes : G-TAG revisité avec les grammaires catégorielles abstraites) [in French]. In *Proceedings of TALN 2014 (Volume 1: Long Papers)*, pages 161–172, Marseille, France. Association pour le Traitement Automatique des Langues.

Philippe de Groote. 2001. Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 252–259, Toulouse, France. Association for Computational Linguistics.

Lidija Iordanskaja, Richard Kittredge, and Alain Polguère. 1991. *Lexical Selection and Paraphrase in a Meaning-Text Generation Model*, pages 293–312. Springer US, Boston, MA.

Sylvain Kahane and François Lareau. 2005. Grammaire d'Unification Sens-Texte : modularité et polarisation. pages 23–32. Grammaire d'Unification Sens-Texte : modularité et polarisation. Dourdan.

Makoto Kanazawa. 2007. Parsing and generation as datalog queries. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 176–183, Prague, Czech Republic. Association for Computational Linguistics.

Igor Mel'Čuk and Alain Polguère. 2021. Les fonctions lexicales dernier cri. In Sébastien Marengo, editor, *La Théorie Sens-Texte. Concepts-clés et applications*, Dixit Grammatica, pages 75–155. L'Harmattan.

I.A. Mel'čuk, I.A. Mel'čuk, D. Beck, and A. Polguère. 2012. *Semantics: From Meaning to Text*, volume 1 of *Semantics: From Meaning to Text*. John Benjamins Publishing Company.

I.A. Mel'čuk, I.A. Mel'čuk, D. Beck, and A. Polguère. 2013. *Semantics: From Meaning to Text*, volume 2 of *Semantics: From Meaning to Text*. John Benjamins Publishing Company.

I.A. Mel'čuk, I.A. Mel'čuk, D. Beck, and A. Polguère. 2015. *Semantics: From Meaning to Text*, volume 3 of *Semantics: From Meaning to Text*. John Benjamins Publishing Company.

Jasmina Milićević. 2006. A short guide to the meaning-text linguistic theory. *Journal of Koralex*, 8:187–233.

Sylvain Pogodalla. 2016. ACGtk : un outil de développement et de test pour les grammaires catégorielles abstraites (ACG TK : a toolkit to develop and test abstract categorial grammars). In *Actes de la conférence conjointe JEP-TALN-RECITAL 2016. volume 5 : Démonstrations*, pages 1–2, Paris, France. AFCP - ATALA.

Sylvain Pogodalla. 2017a. A syntax-semantics interface for Tree-Adjoining Grammars through Abstract Categorial Grammars. *Journal of Language Modelling*, 5(3):527–605.

Sylvain Pogodalla. 2017b. Abstract Categorial Grammars as a Model of the Syntax-Semantics Interface for TAG. In *FSMNLP 2017 and TAG+13 conference*, Umeå, Sweden.

Aarne Ranta. 2004. Grammatical framework. *J. Funct. Program.*, 14:145–189.

Sylvain Salvati. 2005. *Problèmes de filtrage et problème d'analyse pour les grammaires catégorielles abstraites*. Ph.D. thesis, Institut National Polytechnique de Lorraine. Thèse de doctorat dirigée par Philippe de Groote.

Sylvain Salvati. 2010. On the membership problem for non-linear abstract categorial grammars. *Journal of Logic, Language and Information*, 19(2):163–183.

Leo Wanner, Bernd Bohnet, Nadjet Bouayad-Agha, François Lareau, and Daniel Nicklaß. 2010. Marquis: Generation of user-tailored multilingual air quality bulletins. *Applied Artificial Intelligence*, 24(10):914–952.

Measuring Fine-Grained Semantic Equivalence with Abstract Meaning Representation

Shira Wein
Georgetown University
sw1158@georgetown.edu

Zhuxin Wang
Georgetown University
zw85@georgetown.edu

Nathan Schneider
Georgetown University
nathan.schneider@georgetown.edu

Abstract

Identifying semantically equivalent sentences is important for many NLP tasks. Current approaches to semantic equivalence take a loose, sentence-level approach to "equivalence," despite evidence that fine-grained differences and implicit content have an effect on human understanding and system performance. In this work, we introduce a novel, more sensitive method of characterizing cross-lingual semantic equivalence that leverages Abstract Meaning Representation graph structures. We find that parsing sentences into AMRs and comparing the AMR graphs enables finer-grained equivalence measurement than comparing the sentences themselves. We demonstrate that when using gold or even automatically parsed AMR annotations, our solution is finer-grained than existing corpus filtering methods and more accurate at predicting strictly equivalent sentences than existing semantic similarity metrics.

1 Introduction

Translation between two languages is not always completely meaning-preserving, and information can be captured by one sentence which is not captured by the other. Semantic divergence (or conversely, semantic equivalence) detection aims to pick out parallel texts which have less than equivalent meaning. Though semantic divergence across sentences in parallel corpora has been well-studied, current detection methods fail to capture the full scope of semantic divergence. State-of-the-art semantic divergence systems rely on perceived *sentence-level divergences*, which do not entirely encapsulate all semantic divergences.

For example, consider the parallel French and English sentences from the REFreSD dataset (Briakou and Carpuat, 2020) shown in Figure 1. The French sentence says "tous les autres édifices" (*all other buildings*) while the English specifies "all other *religious* buildings." Because the sentence goes on to list religious buildings, it could be inferred from context that the French is describing other *religious* buildings despite being omitted; the sentences thus convey the same overall meaning but are not *exactly* parallel. Under a strict or close analysis of the translation, these sentences could be considered divergent—because the meanings are not identical—but at the sentence-level they are essentially equivalent.

All other *religious* buildings are mosques or Koranic schools founded after the abandonment of Old Ksar in 1957.

Tous les autres édifices sont des mosquées ou des écoles coraniques fondées à l'époque postérieure à l'abondance du vieux ksar en 1957.

Figure 1: Two parallel sentences from the REFreSD dataset marked as having no meaning divergence, for which the AMRs diverge.

Fine-grained semantic equivalence detection is not widely studied—in spite of evidence that: (1) implicit information can be critical to the understanding of the sentence (Roth and Anthonio, 2021), (2) fine-grained divergences in parallel training data have a negative effect on neural machine translation system performance (Briakou and Carpuat, 2021), and finally, that (3) fine-grained semantic equivalence detection holds promise for a number of applications. Most notably, translation studies, semantic analyses, and language learning contexts could all benefit from the distinction between semantically equivalent sentence pairs and sentence pairs which have subtle or implicit differences (Bassnett, 2013). A fine-grained divergence detection system would enable the probing of machine translation models for semantic equivalence (Mallinson et al., 2017) and could point to areas where the source language itself affects semantics in parallel sentences (Taguchi, 2005). Other potential uses include: reducing the workload of human

Proceedings of the 15th International Conference on Computational Semantics pages 144–154
June 21–23, 2023. ©2023 Association for Computational Linguistics

translators in post-editing of machine translation output by filtering out exactly semantically equivalent sentence pairs (Green et al., 2013) and cross-lingual text reuse detection (plagiarism detection) (Potthast et al., 2011).

Given the wide-ranging motivation for the development of a fine-grained equivalence detection system, coupled with the notable gap in research on this task, we argue that a finer-grained measure of semantic equivalence is needed: a way to detect *strictly* semantically equivalent sentence pairs. We leverage explicit semantic information in the form of Abstract Meaning Representation (AMR; Banarescu et al., 2013) to fill this gap. In this work, we demonstrate that parsing sentences into AMR graphs and comparing those graphs enables a finer-grained semantic comparison than simply comparing the sentences. We suspect that AMR may be useful in this case because it makes explicit every concept and relationship between those concepts present in the sentence, taxonomically categorizing each concept's role and argument.

With analysis of data in two language pairs (English-French and English-Spanish), we demonstrate that sentence-level divergence annotations can be coarse-grained, neglecting slight differences in meaning (§3). We find that comparing two AMR graphs is an effective way to characterize meaning in order to uncover finer-grained divergences (§4), and this can be achieved even with automatic AMR parsers (§5). Finally, in §6 we evaluate our AMR-based metric on a cross-linguistic semantic textual similarity dataset, and show that for detecting semantic equivalence, it is more precise than a popular existing model, multilingual BERTScore (Zhang et al., 2020).

Our primary contributions include:

- Our novel approach to the identification of semantic divergence which uses AMR to move beyond perceived sentence-level divergences
- A simple pipeline algorithm (which modifies Smatch (Cai and Knight, 2013)) to automate the detection of AMR-level divergence in cross-lingual pairs
- Studies demonstrating that our AMR-based approach accurately captures a finer-grained degree of semantic equivalence than both the state-of-the-art corpus filtering method and a semantic textual metric

We will release the code and dataset for this work upon publication to enable the use of AMR for semantic divergence detection.

2 Background on Semantic Divergence

Semantic divergences can arise when translating from one language to another. These divergences can arise due to different language structure, syntactic differences in the language, or translation choices (Dorr, 1994, 1990). Additional divergences can be introduced when automatically extracting and aligning parallel resources (Smith et al., 2010; Zhai et al., 2018; Fung and Cheung, 2004).

To address these divergences, a number of systems have been developed to automatically identify divergences in parallel texts (Carpuat et al., 2017; Vyas et al., 2018; Briakou and Carpuat, 2020, 2021; Zhai et al., 2020). The approach taken by Briakou and Carpuat (2020) to detecting sentence-level semantic divergences involves training multilingual BERT (Devlin et al., 2018) to rank sentences diverging to various degrees. They introduced a novel dataset called Rational English-French Semantic Divergences (REFreSD). REFreSD is a subset of the French-English WikiMatrix (Schwenk et al., 2021) with crowdsourced annotations classifying the sentences as having no meaning divergence, some meaning divergence, or being unrelated.

Recent work has investigated the differences in cross-lingual (English-Spanish) AMR pairs within the framework of translation divergences (Wein and Schneider, 2021). Specifically, this work developed an annotation schema to classify the types and causes of differences between cross-lingual AMR pairs. We use this dataset to test the performance of our system on English-Spanish gold AMR pairs. (For English-French, we produce our own gold judgments of AMR divergence to test our algorithm.) Additional prior work has explored the role of structural divergences in cross-lingual AMR parsing (Blloshmi et al., 2020; Damonte, 2019).

The relationship between Abstract Meaning Representation metrics and measures of semantic similarity has been explored in (Leung et al., 2022). Recent work has also integrated sentence-level embeddings and comparison of AMR graphs (Opitz et al., 2021; Wein and Schneider, 2022; Zeidler et al., 2022).

3 AMR for Identification of Semantic Equivalence

Semantic representations are designed to capture and formalize the meaning of a sentence. In partic-

He later scouted in Europe for the Montreal Canadiens.

```
(s / scout-02
      :ARG0 (h / he)
      :ARG1 (c / continent
            :wiki "Europe"
            :name "Europe")
      :ARG2 (c2 / canadiens
            :mod "Montreal")
      :time (a / after))
```

Il a plus tard été dépisteur du Canadiens de Montréal en Europe. (*He later scouted for the Montreal Canadiens in Europe.*)

```
(d / dépister-02
      :ARG0 (i / il)
      :ARG1 (c / continent
            :wiki "Europe"
            :name "Europe")
      :ARG2 (c2 / canadiens
            :mod "Montreal")
      :time (p / plus-tard))
```

Figure 2: A pair of sentences and their human annotated AMRs, for which the sentences receive a "no meaning divergence" judgment in the REFreSD dataset, and are also equivalent per AMR divergence.

ular, the Abstract Meaning Representation (AMR) framework aims to formalize sentence meaning as a graph in a way that is conducive to broad-coverage manual annotation (Banarescu et al., 2013, 2019). These semantic graphs are rooted and labeled, such that each node of the graph corresponds to a semantic unit. AMR does not capture nominal or verbal morphology or many function words, abstracting away from the syntactic features of the sentence.

We leverage the semantic information captured by AMR to recognize semantic equivalence or divergence across parallel sentences. Figure 2, for example, illustrates a strictly meaning-equivalent sentence pair along with the AMRs. Though the sentences differ with respect to syntax and lexicalization, the AMR graphs are structurally isomorphic. If the AMR structures were to differ, that would signal a difference in meaning.

Two particularly beneficial features of the AMR framework are the rooted structure of each graph, which elucidates the semantic focus of the sentence, as well as the concrete set of specific non-core roles, which are useful in classifying the specific relation between concepts/semantic units in the sentence. For example, in Figure 3, the emphasis on the English sentence is on possession—*your* planet—but the emphasis on the Spanish sentence is on place of origin, asking, which planet are you *from?* This difference in meaning is reflected in the diverging roots of the AMRs.

Which is your planet?

```
(p / planet
    :poss (y / you)
    :domain (a / amr-unknown))
```

¿ De qué planeta eres ? (*Which planet are you from?*)

```
(s / ser-de-91
   :ARG1 (t / tú)
   :ARG2 (p / planeta
       :domain (a / amr-desconocido)))
```

Figure 3: Two parallel sentences and AMRs from the Migueles-Abraira et al. English-Spanish AMR dataset, which diverge in meaning. The Spanish role labels are translated into English here for ease of comparison.

Finally, we identify the fact that non-core roles (such as `:manner`, `:degree`, and `:time`) are particularly helpful in identifying parallelism or lack of parallelism between the sentences. This is because AMR abstracts away from the syntax (so that word order and part of speech choices do not affect equivalence), but instead explicitly codes relationships between concepts via semantic roles. Furthermore, AMRs use special frames for certain relations, such as `have-rel-role-91` and `include-91`, which can be useful in enforcing parallelism when the meaning is the same but the specific token is not the same. For example, if the English and French both have a concession which the English marks via "although" and the French marks with "*mais*" (*but*), the AMR special frame role will still preserve parallelism by indicating them both as a concession.

Granularity of the REFreSD dataset. Sentence-level divergences (as annotated in REFreSD) do not capture all meaning differences. Another example of this surface-level divergence adjudication, using sentences from the REFreSD dataset, is shown in Figure 4. These sentences are marked as having no meaning divergence in the REFreSD dataset but do have diverging AMR pairs. The difference highlighted by the AMR pairs is the `:time` role of reach / *atteindre*. The English sentence says that no. 1 is reached "within a few weeks" of the release, while the French sentence says that no. 1 is reached the first week of the release (*la première semaine*).

We explore the ability to discover semantic divergences in sentences either with gold parallel AMR annotations or with automatically parsed AMRs using a multilingual AMR parser, in order to enable the use of this approach on large corpora (considering that AMR annotation requires training).

We propose that an approach to detecting di-

Although the sales were slow (admittedly, according to the band), the second single from the album, "Sweetest Surprise" reached No. 1 in Thailand *within a few weeks* of release.

Même si les exemplaires ont du mal à partir (comme l'admet le groupe), le second single de l'album, Sweetest Surprise, atteint la première place en Thaïlande *la première semaine* de sa sortie.

Figure 4: Two parallel sentences from the REFreSD dataset (Briakou and Carpuat, 2020) marked as having no meaning divergence, but for which the AMRs diverge. Italicized spans indicate the cause of the AMR divergence.

vergences using AMR will be a stricter, finer-grained measurement of semantic divergence than perceived sentence-level judgments.

4 Examining and Automatically Detecting Differences in Gold AMRs

In this section, we **evaluate the ability of AMR to expose fine-grained differences in parallel sentences** and how to **automatically detect those differences**. In order to do so, we produce and examine English-French AMR pairs, which is the first annotated dataset of French AMRs; we also examine a number of English-Spanish AMR pairs.

This is a relatively small dataset (100 English-French items and 50 English-Spanish items) because it serves as a manually annotated precursor to validate our hypothesis, ahead of our extensive automatically-produced AMR experimentation (§5) which uses 1033 items.

4.1 Examination of Gold AMR Data

We focus on French for effective comparison with sentence-level semantic divergence models (because of the available resources), though it also makes for ideal candidates in a cross-lingual AMR comparison, as it is broadly syntactically similar to English. This suggests that the AMRs could be expected to look similar (though not exactly the same) as inflectional morphology and function words are not represented in AMR. Prior work has investigated the transferability of AMR to languages other than English, and has found that it is not exactly an interlingua, but in some cases cross-lingual AMRs align well. Additionally, some languages are more compatible (Chinese) with English AMR than other languages (Czech) (Xue et al., 2014).

English-French AMR Parallel Corpus In investigating the differences between the degree of divergence captured by AMR and sentence-level divergence, we aim to compare quantitative measures of AMR similarity with corresponding sentence-level judgments of similarity. In order to compare human judgments and AMR judgments, we develop the first French-English AMR parallel corpus, which represents the first application of AMR to French. We produce gold AMR annotations for 100 sentences, which were randomly sampled, from the REFreSD dataset (Briakou and Carpuat, 2020; Linh and Nguyen, 2019). We also test our system on the full REFreSD dataset, using an automatic AMR parser (described in §5).

For the French AMR annotation process, the role/argument labels were added in English as has been done in related non-English AMR corpora (Sobrevilla Cabezudo and Pardo, 2019), and the concept (node) labels were in French. The specific concept sense numbers were based on English PropBank frames (Kingsbury and Palmer, 2002; Palmer et al., 2005).

	AMR Div.	AMR Equi.
Sentence-Level Div.	57	0
Sentence-Level Equi.	26	17

Table 1: Comparison between AMR Divergence annotations and Sentence-Level Divergence REFreSD annotations for 100 French-English sentences.

Findings from Corpus Annotation In light of our research question considering whether AMR can serve as a proxy of fine-grained semantic divergence, we consider both qualitative and quantitative evidence. While producing this small corpus of French-English parallel AMRs, our suspicions that AMR would be able to more fully capture semantic divergence than perceived sentence-level divergence were confirmed. We uncovered a number of ways in which perceived sentence-level equivalence is challenged by the notion of AMR divergence. Take the example in Figure 1. The difference between "religious" being applied in the French sentence and appearing in the English sentence is not captured by perceived sentence-level divergence, but is captured by AMR divergence.

The results in Table 1 demonstrate that when using AMR as a lens to filter meaning, the result is always stricter than when simply comparing their corresponding sentences in the form of human judgment. There are no instances where the sentence-level annotation claims that the sentences are divergent but the AMR annotations are equivalent. Conversely, there are 26 instances with AMR divergence but no perceived sentence-level semantic divergence. From this annotation we find that AMR divergence is a finer-grained measure of divergence than perceived sentence-level divergence.

4.2 Quantifying Divergence in Cross-Lingual AMR Pairs

We have shown that not all pairs that humans considered equivalent at the sentence level receive isomorphic AMRs because they actually contain low-level semantic divergences. This suggests AMRs can be useful for more sensitive automatic detection of divergence. Now, we investigate whether we can automatically detect and quantify this divergence on gold AMRs via the graph comparison algorithm Smatch. In order to quantify this divergence in cross-lingual AMR pairs, we develop a simple pipeline algorithm which is a modified version of Smatch and incorporates token alignment. We test our modified Smatch algorithm on gold English-French AMR pairs and gold English-Spanish AMR pairs in comparison to the similarity scores output by Briakou and Carpuat (2020).

Modified cross-lingual version of Smatch. Our simple pipeline algorithm extends Smatch, a measurement of similarity between two (English) AMRs (Cai and Knight, 2013). Smatch quantifies the similarity of two AMRs by searching for an alignment of nodes between them that maximizes the F_1-score of matching (*node1*, *role*, *node2*) and (*node1*, `instance-of`, *concept*) triples common between the graphs. However, Smatch was designed to compare AMRs in the same language, with the same role and concept vocabularies.

To compare AMR nodes across languages, the nodes first need to be cross-lingually aligned. This involves translating the concept and role labels. We take a simple approach of first word-aligning the sentence pair to ascertain corresponding concepts (most of which are lemmas of content words in the sentence). Our approach is similar to that of *AMRICA* (Saphra and Lopez, 2015), but we use a different word aligner (fast_align rather than GIZA++[1]) and deterministic translation of role names if the labels are not in English. The deterministic translation is done using a mapping of the role names between Spanish and English provided in the Spanish annotation guidelines (Migueles Abraira, 2017). To align AMR graphs across languages, we word-align the sentence pairs, then map these alignments onto nodes in the graph (most concept labels on nodes correspond to lemmas of words in the sentence). Role names are mapped deterministically based on a list from Migueles Abraira (2017).

We normalize the strings and remove sense labels from the English and French/Spanish concept labels. An error that we noticed while developing the system was associated with the same concept label appearing more than once in either AMR, so we tag repeated words numerically before performing the alignment.

Finally, we run Smatch with the default number of 4 random restarts to produce an alignment. The Smatch score produced is an F1 score from 0 to 1 where 1 indicates that the AMRs are equivalent. This can be converted to a binary judgment, where all non-1 pairs are divergent, or used as a continuous value (as in §5).

Testing our Approach on Gold AMRs. One of the benefits of leveraging semantic representations in our approach to semantic divergence detection is that the identification of divergence boils down to determining whether the graphs are isomorphic or not (and accurate word alignment). This suggests that our pipeline algorithm (§4.2) should be highly effective at identifying whether AMR pairs are divergent or equivalent. In order to test our AMR-based approach to strict semantic equivalence identification, we first test on gold AMRs, which are created by humans and thus have no external noise from being automatically parsed.

We expect that our AMR divergence characterization would behave differently from a classifier of sentence-level divergence. This is because the sentence-level classification methods require specialized training data and as such learn to classify based on the perceived sentence-level judgments of semantic divergence. To test the strictness of our framing, we validate our quantification on gold English-French and gold English-Spanish cross-lingual AMR pairs.

Results on Gold English-French AMR Pairs We test our pipeline algorithm on the 100 English-French annotated AMR pairs described in §4.1. As expected, the simple pipeline algorithm is very accurate at correctly predicting whether the cross-lingual pairs do or do not diverge according to the

[1] fast_align has been shown to produce more accurate word alignments, such as in the case for Latvian-English translation (Girgzdis et al., 2014).

	Equivalent (17)			**Divergent** (83)			**All**
System	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Ours	1.00	0.82	0.90	0.97	1.00	0.98	0.97
BC'20	0.39	0.82	0.53	0.95	0.73	0.83	0.75

Table 2: FR-EN: Binary divergence classification on on 100 gold French-English AMR pairs, annotated for sentences from the REFreSD dataset. Precision (P), Recall (R), and F1 scores are reported for the equivalent, divergent, and all AMR pairs. We compare the performance of our model with the performance of the (Briakou and Carpuat, 2020) model, referenced as BC'20, on our finer-grained measure of divergence for the same English-French parallel sentences.

stricter criterion.

Table 2 showcases the ability of our pipeline system and the (Briakou and Carpuat, 2020) system (described in §2) to identify these finer-grained semantic divergences. On these English-French AMR pairs, the F1 score for our system is 0.97 overall and 1.00 for equivalent AMR pairs. This high level of accuracy indicates we can reliably predict cross-lingual AMR divergence.

The (Briakou and Carpuat, 2020) system performs worse when using our finer-grained delineation of semantic divergence, achieving an F1 score of 0.75.[2] Unsurprisingly, the precision, recall, and F1 for their system is lower than the performance of our system, because theirs is not trained to pick up on these more subtle divergences. Note that on their own measure of divergence (perceived sentence-level divergence), the system achieves an F1 score of 0.85 on these same 100 sentences.

Of the 3 errors made by our algorithm (in all cases, classifying equivalent AMR pairs as divergent), 2 of the 3 are caused by word alignment errors. Named entities seem to pose an issue with fast_align for our use case.

	Equivalent (13)			**Divergent** (37)			**All**
System	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
Ours	1.00	0.92	0.96	0.97	1.00	0.99	0.98
BC'20	0.24	0.38	0.29	0.72	0.57	0.64	0.52

Table 3: EN-ES: Binary divergence classification with gold parallel AMRs. Included are Precision (P), Recall (R), and F1 for the Equivalent, Divergent, and All AMR pairs for our pipeline algorithm compared to the system by Briakou and Carpuat (2020), referenced as BC'20, on the same English-Spanish parallel sentences.

Results on Gold English-Spanish AMR Pairs. In addition to testing our system on our English-French AMR annotations, we test our system on the 50 English-Spanish AMRs and sentences released by Migueles-Abraira et al. (2018), who collected sentences from *The Little Prince* and altered them to be more literal translations; recent work classified these AMRs according to a structural divergence schema (Wein and Schneider, 2021).

In Table 3, we measure the ability of our pipeline system and the (Briakou and Carpuat, 2020) system to detect semantic divergences at a stricter level, as picked up by the AMR divergence schema.

Our system performs similarly well on Spanish-English pairs as it did on the English-French pairs, described in Table 2. This demonstrates that our pipeline algorithm is not limited to success on only one language pair, and we further affirm that the simple pipeline algorithm is a reliable way to predict cross-lingual AMR divergence.

5 Strictness Results Using Automatic English-French AMR Parses

In §4, we confirmed our hypothesis by demonstrating that we are able to use gold (human annotated) AMRs to capture a finer-grained level of semantic divergence, quantifiable via Smatch. We extend this further by determining whether fine-grained semantic divergences can be detected well even when using noisy automatically parsed AMRs. To do so, we compare the Smatch scores of automatically parsed AMR pairs with the human judgments output on the corresponding sentences by Briakou and Carpuat (2020).

To take the expensive human annotation piece out of the process, we show that automatic AMR parses can be used instead of gold annotations by establishing a threshold, instead of via binary classification. Therefore, we use the F1 score output by our pipeline algorithm as a *continuous score* and establish thresholds (described later in this section) to divide the data between divergent and equivalent.

We automatically parse cross-lingual AMRs for the entirety of the English-French parallel REFreSD dataset (1033 pairs). The REFreSD dataset is parsed using the `mbart-st` version of SGL, a state-of-the-art multilingual AMR parser (Procopio et al., 2021). The (monolingual) Smatch score for the SGL parser, comparing our gold AMRs with the automatically parsed AMRs, is 0.41 for the 100 French sentences using Smatch (0.43 using our

[2] The Briakou and Carpuat (2020) system does not take AMRs as input, so we use the corresponding sentences as input for their system.

pipeline algorithm)[3] and 0.52 for the 100 parallel English sentences using Smatch.

In doing error analysis, we find that the data points which are classified as having no meaning divergence but have extremely low F1 scores are largely suffering from parser error. We do find that there are pairs classified in REFreSD as having no meaning divergence at the sentence-level that do correctly receive low F1 scores. For example, the sentence pair in Figure 4, which has a REFreSD annotation of sentence-level equivalence and a gold AMR-level annotation of divergence, was assigned an F1 score of 0.3469.

Despite Smatch scores of 0.5 between the gold and automatic parses, both are usable for the task of detecting finer-grained semantic equivalence. To demonstrate the usefulness of our continuous metric of semantic divergence using automatically parsed AMR pairs, we develop potential thresholds at which you could separate data as being equivalent vs. divergent.

Because our metric is more sensitive, a practitioner could choose their own threshold by determining appropriate precision (how semantically equivalent they wanted a subset of filtered data to be) and recall (how much data they are willing to filter out) needs. This tradeoff is depicted in Figure 5. For example, if all pairs are marked as equivalent, precision would be approximately 40% on the REFreSD dataset if considering solely the “no meaning divergence” pairs equivalent.

Comparing with model probabilities. Though it is reasonable to assume that if the gold AMR annotations provide a distinctly finer-grained measure of divergence than sentence-level divergence then this would also be the case when using automatically parsed AMRs, we want to ensure the continued strictness of our methodology. To do this, we compare the values of our continuous metric and the probabilities produced by the (Briakou and Carpuat, 2020) system.

Because the probabilities produced by the system described in (Briakou and Carpuat, 2020) are always very close to 1 (equivalent) or very close to 0 (divergent) and there are far more divergent instances than equivalent instances, median and mode serve as a more effective form of comparison than mean between our F1 score and their probability score. Above the 0.7 threshold, the median F1 for our system is 0.7869 and mode is 0.8; the median probability for the Briakou and Carpuat (2020) system is 0.9990 and the mode is 1.0. For the 0.6 threshold, our median is 0.6667 and our mode is 0.6667; their median is 0.9871 and mode is 1.0. Above the 0.5 threshold, our median is 0.5814 and our mode is 0.5; their median is 0.8907 and mode is 1.0. Because these numbers are lower for our system than their system, we confirm that our measure is a stricter measure of equivalence even when using the automatically parsed AMRs.

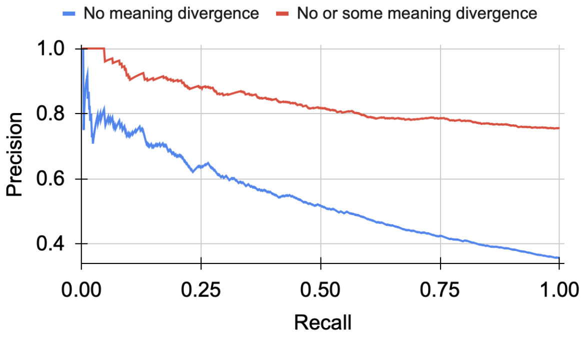


Figure 5: Precision / recall curve for equivalence detection in the 1033 sentence pairs in the full REFreSD dataset (English-French) using automatic AMR parses. Precision reflects the percent of sentences in which REFreSD human annotation was equivalent (as labeled as no meaning divergence in the blue/bottom curve, or as labeled as having either no or some meaning divergence in the red/top curve).

If the goal is to prioritize items for a human to look at on a fixed budget, the absolute scores may matter less than rankings, though the rankings additionally differ drastically. Of the top 50 sentences ranked by AMR divergence (which range in AMR similarity score from 0.96 to 0.67), only 19 of the 50 appear in the 166 sentences scored 1.0 by the Briakou and Carpuat (2020) system.

6 Sentence Similarity Evaluation with Automatically Parsed English-Spanish AMRs

As we have shown in previous sections, our AMR-focused approach in general is stricter than sentence-based measures of equivalence, in particular corpus filtering methods. Because our system is a stricter measure of semantic equivalence, it may be the case that our system can more precisely identify the most similar sentences than existing

[3] The SGL parser approaches cross-lingual parsing as the task of recovering the AMR graph for the English translation of the sentence, as defined in prior work (Damonte and Cohen, 2018). The result is that the parses of French sentences are largely in English, and default to French concepts only for out-of-vocabulary French words. The alignments in our pipeline account for this to better reward the native French concepts.

measures of sentence similarity. In this final results section, we look at the most semantically equivalent sentences in the dataset (as judged by our approach and as judged by multilingual BERTscore (mBERTscore; Zhang et al., 2020)) in comparison to their human judgments of equivalence. Specifically, we aim to investigate: (1) whether the average human similarity score for the most similar n sentences is higher when ranked by our AMR-based metric versus when ranked by mBERTscore, and (2) whether human judgments of sentence similarity for the most similar sentences are more correlated with our AMR-based metric than with mBERTscore (an embedding-based automatic evaluation metric of semantic textual similarity). We compare our AMR-based metric to mBERTscore because it has been shown to work well in cross-lingual settings when comparing system output to a reference (Koto et al., 2021). Semantic textual similarity considers the question of semantic equivalence slightly differently because it rewards semantic overlap as opposed to equivalence.

Data. To perform this comparison, we use the 301 human annotated Spanish-English test sentences from the news down of the SemEval task on semantic textual similarity (Agirre et al., 2016).

6.1 Smatch with Cross-Lingual AMR parsing

For our analysis, we use the Translate-then-Parse system (T+P; Uhrig et al., 2021). Providing the Spanish sentences as input, T+P translates them into English, and then runs an AMR parser[4] on the English translation. Because the Spanish sentence was translated into English and *then* parsed, this automatic parse can be compared against the automatic parse of the original English sentence with plain Smatch (no cross-lingual alignment added).

As we have established in §5, the noise introduced by automatic parsers can be overcome in our approach. We validate that the Smatch scores retrieved after using Uhrig et al.'s (2021) parser still bears some correlation with the Smatch scores on the aligned gold AMRs.[5]

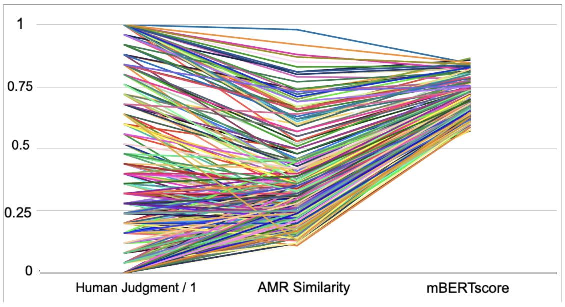


Figure 6: All data points normalized to a range of 0 to 1 for the Spanish-English sentence pairs from Agirre et al. (2016), including human judgment, AMR similarity score, and mBERTscore. This displays the decreased range of mBERTscore judgments in comparison to human judgments and AMR similarity.

6.2 Sentence Similarity Results

The average human judgment score, on a scale of 0 to 5 with 5 being exactly equivalent, for all sentence pairs which have an AMR similarity score greater than 0.8 is 4.98. The average human judgment score for all sentence pairs which have an mBERTscore similarity score greater than 0.8 is 4.89. Similarly, the average human judgment score for pairs with an AMR similarity score of greater than 0.7 is 4.86, while the average human judgment score for pairs with an mBERTscore greater than 0.7 is 3.8. This is because mBERTscore takes a much broader view of semantic equivalence. While the human judgments occupy the full range of 0 to 5, the mBERTscores of these sentences range from 0.57 to 0.87, as shown in Figure 6. The AMR similarity score ranges from 0.11 to 0.98.

This might suggest that then a higher threshold should be used for mBERTscore to achieve the same level of semantic granularity. However, our AMR similarity metric is also more correlated with human judgments for the most semantically equivalent sentences. For the top 20 items as ranked by AMR similarity, Pearson correlation with human judgments is 0.4068, while the top 20 items as ranked by mBERTScore are not correlated with human judgments (−0.0023). When looking at all items above the mBERTscore of 0.8, correlation with human judgment is 0.1645, whereas for all items above the AMR similarity score of 0.8, correlation with human judgment is 0.2675. Overall, AMR similarity score correlates with human judgment at a coefficient of 0.8367, which is slightly lower than the 0.8605 correlation between mBERTscore and human judgment. This evidence further supports that our metric is in fact a finer-

[4] Via amrlib: https://github.com/bjascob/amrlib

[5] On the 50 Spanish-English sentences mentioned in §4, the correlation between the Smatch scores (in comparison to the same gold AMRs) when using either the translation-then-parse method or the method of aligning concepts via fast_align is 0.31. This can be interpreted as a weak correlation. We find that both methods (translating the sentence first, or our pipeline algorithm aligning concepts in AMRs of different languages) work sufficiently well to capture the amount of divergence between cross-lingual AMR pairs.

grained measure of semantic equivalence, and is therefore better at identifying which sentences are exactly semantically equivalent.

7 Conclusion

In this work, we have proposed a stricter measure of semantic divergence than existing systems which rely on perceived differences at the sentence level. We have effectively demonstrated that parsing sentences into Abstract Meaning Representations and comparing those graphs facilitates a more detailed semantic comparison, when using either gold *or* automatically parsed AMR pairs.

We are excited by the numerous possible applications of this finer-grained measure of meaning (mentioned in §1), both from an engineering standpoint and the potential it has in translation and language-learning environments to highlight specific differences in language pairs.

Limitations

As the first work exploring the use of AMR for fine-grained semantic equivalence assessment, our work faces a few limitations. First, our results were limited to the language pairs we work with. In the three languages pairs, we claim that our approach is a more fine-grained measure of semantic equivalence than existing approaches. Future work on other language pairs would provide further insight into its applicability to languages less syntactically similar to English. Second, it may be worth considering the use of other semantic representations in addition to AMR. Though our results confirm that AMR captures many aspects of meaning that are important to human judgments of cross-lingual similarity, AMR does not capture all aspects of semantics. Finally, our system is limited by the performance of automatic AMR parsers. In §5, we show that, despite Smatch scores of 0.5 between the gold and automatic parses, both are usable for the task of detecting finer-grained semantic equivalence. Still, it is reasonable to expect that better parsers would lead to better performance by our system, and thus our results currently suffer due to less-than-perfect performance.

Acknowledgements

Thank you to Rexhina Blloshmi, Eleftheria Briakou, and Luigi Procopio for providing system results, as well as to anonymous reviewers for their feedback. This work was supported in part by a Clare Boothe Luce Scholarship to the first author and NSF award IIS-2144881.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2019. Abstract Meaning Representation (AMR) 1.2.6 specification. https://github.com/amrisi/amr-guidelines/blob/master/amr.md.

Susan Bassnett. 2013. *Translation studies*. Routledge.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2487–2500, Online. Association for Computational Linguistics.

Eleftheria Briakou and Marine Carpuat. 2020. Detecting Fine-Grained Cross-Lingual Semantic Divergences without Supervision by Learning to Rank. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1563–1580, Online. Association for Computational Linguistics.

Eleftheria Briakou and Marine Carpuat. 2021. Beyond noise: Mitigating the impact of fine-grained semantic divergences on neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7236–7249, Online. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Marine Carpuat, Yogarshi Vyas, and Xing Niu. 2017. Detecting cross-lingual semantic divergence for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 69–79, Vancouver. Association for Computational Linguistics.

Marco Damonte. 2019. *Understanding and Generating Language with Abstract Meaning Representation*. Ph.D. thesis, University of Edinburgh.

Marco Damonte and Shay B. Cohen. 2018. Cross-lingual Abstract Meaning Representation parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1146–1155, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bonnie Dorr. 1990. Solving thematic divergences in machine translation. In *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics*, ACL '90, page 127–134, USA. Association for Computational Linguistics.

Bonnie J. Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

Pascale Fung and Percy Cheung. 2004. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1051–1057, Geneva, Switzerland. COLING.

Valdis Girgzdis, Maija Kale, Martins Vaicekauskis, Ieva Zarina, and Inguna Skadiņa. 2014. Tracing mistakes and finding gaps in automatic word alignments for Latvian-English translation. In Andrius Utka, Gintarė Grigonytė, Jurgita Kapočiūtė-Dzikienė, and Jurgita Vaičenonienė, editors, *Human Language Technologies – The Baltic Perspective*, volume 268 of *Frontiers in Artificial Intelligence and Applications*, pages 87–94. IOS Press.

Spence Green, Jeffrey Heer, and Christopher D. Manning. 2013. The efficacy of human post-editing for language translation. In *Proc. of CHI*, pages 439–448, New York, NY, USA.

Paul Kingsbury and Martha Palmer. 2002. From Tree-Bank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Evaluating the efficacy of summarization evaluation across languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 801–812, Online. Association for Computational Linguistics.

Wai Ching Leung, Shira Wein, and Nathan Schneider. 2022. Semantic similarity as a window into vector- and graph-based metrics. In *Proc. of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 106–115, Abu Dhabi, United Arab Emirates (Hybrid).

Ha Linh and Huyen Nguyen. 2019. A case study on meaning representation for Vietnamese. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 148–153, Florence, Italy. Association for Computational Linguistics.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.

Noelia Migueles Abraira. 2017. A study towards Spanish Abstract Meaning Representation. Master's thesis, University of the Basque Country, Donostia-San Sebastián, Spain, June.

Noelia Migueles-Abraira, Rodrigo Agerri, and Arantza Diaz de Ilarraza. 2018. Annotating Abstract Meaning Representations for Spanish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Juri Opitz, Angel Daza, and Anette Frank. 2021. Weisfeiler-leman in the bamboo: Novel AMR graph metrics and a benchmark for AMR graph similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Martin Potthast, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. 2011. Cross-language plagiarism detection. *Language Resources and Evaluation*, 45(1):45–62.

Luigi Procopio, Rocco Tripodi, and Roberto Navigli. 2021. SGL: Speaking the graph languages of semantic parsing via multilingual translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 325–337, Online. Association for Computational Linguistics.

Michael Roth and Talita Anthonio. 2021. UnImplicit shared task report: Detecting clarification requirements in instructional text. In *Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language*, pages 28–32, Online. Association for Computational Linguistics.

Naomi Saphra and Adam Lopez. 2015. AMRICA: an AMR inspector for cross-language alignments. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 36–40, Denver, Colorado. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411, Los Angeles, California. Association for Computational Linguistics.

Marco Antonio Sobrevilla Cabezudo and Thiago Pardo. 2019. Towards a general Abstract Meaning Representation corpus for Brazilian Portuguese. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 236–244, Florence, Italy. Association for Computational Linguistics.

Naoko Taguchi. 2005. Comprehending implied meaning in English as a foreign language. *The Modern Language Journal*, 89(4):543–562.

Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. Translate, then parse! a strong baseline for cross-lingual AMR parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1503–1515, New Orleans, Louisiana. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2021. Classifying divergences in cross-lingual AMR pairs. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 56–65, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2022. Accounting for language effect in the evaluation of cross-lingual AMR parsers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3824–3834, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nianwen Xue, Ondřej Bojar, Jan Hajič, Martha Palmer, Zdeňka Urešová, and Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of English AMRs to Chinese and Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1765–1772, Reykjavik, Iceland. European Language Resources Association (ELRA).

Laura Zeidler, Juri Opitz, and Anette Frank. 2022. A dynamic, interpreted CheckList for meaning-oriented NLG metric evaluation – through the lens of semantic similarity rating. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 157–172, Seattle, Washington. Association for Computational Linguistics.

Yuming Zhai, Gabriel Illouz, and Anne Vilnat. 2020. Detecting non-literal translations by fine-tuning cross-lingual pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5944–5956, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuming Zhai, Aurélien Max, and Anne Vilnat. 2018. Construction of a multilingual corpus annotated with translation relations. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 102–111, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proc. of ICLR*, Online.

The Importance of Context in the Evaluation of Word Embeddings: The Effects of Antonymy and Polysemy

James Fodor
The Centre for Brain, Mind and Markets
The University of Melbourne,
Victoria 3010 Australia
jfodor@student.unimelb.edu.au

Simon De Deyne
School of Psychological Sciences
The University of Melbourne
Victoria 3010 Australia
simon.dedeyne@unimelb.edu.au

Shinsuke Suzuki
Faculty of Social Data Science
Hitotsubashi University
shinsuke.szk@gmail.com

Abstract

Word embeddings are widely used for diverse applications in natural language processing. Despite extensive research, it is unclear when they succeed or fail to capture human judgements of semantic relatedness and similarity. In this study, we examine a range of models and experimental datasets[1], showing that while current embeddings perform reasonably well overall, they are unable to account for human judgements of antonyms and polysemy. We suggest that word embeddings perform poorly in representing polysemy and antonymy because they do not consider the context in which humans make word similarity judgements. In support of this, we further show that incorporating additional context into transformer embeddings using general corpora and lexical dictionaries significantly improves the fit with human judgments. Our results provide insight into two key inadequacies of word embeddings, and highlight the importance of incorporating word context into representations of word meaning when accounting for context-free human similarity judgments.

1 Introduction

Lexical semantics seeks to provide a cognitive explanation of how word meaning is represented and how semantic relations such as hyponymy, antonymy and synonymy are encoded. Vector-space models are one of the dominant approaches to studying lexical semantics. In vector-space models, a word is associated with a vector of real numbers called a *word embedding,* which captures information about word co-occurrences in a document or sentence. Each component of this vector corresponds to an abstract feature in an underlying vector space (Almeida and Xexéo, 2019; Lieto et al., 2017). The meaning of each word is thus represented by the direction of its word embedding in semantic space. (In this paper we use 'word embeddings' loosely, referring to any vector representation of word meaning using real numbers).

Word embedding methods are widely used in natural language processing, where they are utilised by machine learning architectures that have achieved impressive performance on a range of applied language tasks (Devlin et al., 2019; Lenci, 2018; Ranashinghe et al., 2019; Young et al., 2018). Vector-space semantics models also have a natural synergy with neuroimaging techniques that measure patterns of voxel activities in response to linguistic stimuli, thus providing an interface between lexical semantics and cognitive neuroscience (Rodrigues et al., 2018; Wang et al., 2020). It is therefore of considerable interest to evaluate the performance of these methods in modelling word meanings.

Vector-space approaches to semantics hypothesise that many aspects of word meaning, including semantic relationships such as synonymy, antonymy, hyponymy, and logical inference, can be efficiently represented by the relative direction of word embeddings in semantic space (Günther et al., 2019; Clark, 2015). One way to test this hypothesis is to compare the similarity relations

[1] Our code and processed datasets are available at https://github.com/bmmlab/lexical-semantics-eval

Proceedings of the 15th International Conference on Computational Semantics pages 155–172
June 21–23, 2023. ©2023 Association for Computational Linguistics

between word embeddings with human judgements of word similarity and relatedness (De Deyne et al., 2016; Lenci et al., 2021). A high correlation between the similarity structure of word embeddings and human similarity judgements is evidence that the embeddings successfully encode information about word meaning and semantic relationships between words.

Existing literature evaluating word embeddings against human similarity judgments, however, has typically ignored the implicit context humans use to make these judgements. We hypothesise that this omission is an important factor contributing to the relatively poor performance of word embedding models when evaluated against certain experimental datasets.

In this study we focus on two specific semantic phenomena in which the effects of context are most likely to be apparent: antonymy and polysemy. In the case of antonymy, we hypothesise that humans judge the meaning of a word differently when it is presented in the context of a word opposite in meaning. Likewise, we hypothesise that humans assess the meaning of polysemous words differently than non-polysemous words due to the need to use contextual information to select the relevant sense. We therefore anticipate an investigation into polysemy and antonymy will be important for understanding the limitations of word embeddings resulting from neglecting context.

1.1 Vector-space semantics models

Word embeddings can be constructed using a variety of techniques. Predict-based embeddings are constructed by training a neural network on a word prediction task, such as predicting the next word in a text (Baroni, Dinu, & Kruszewski, 2014). Knowledge-based methods utilise human curated datasets of semantic relations such as WordNet (Pedersen et al., 2004). Transformers are the most recent class of models, which capture context-specific meaning using multilayered attention neural networks trained on very large natural language corpora (Tripathy et al., 2021). Transformers can be used to compute word embeddings which are modified based on the specific usage of the word, and hence are of particular value in assessing the effects of word context.

One of the most common methods for assessing word embeddings is *semantic similarity*. Similarity is sometimes conceptualised as the degree to which two words are interchangeable (Miller and Charles, 1991). Another metric used in the evaluation of word embeddings is *semantic relatedness*. Relatedness refers to the degree to which the words share any type of semantic relation or psychological association (Gladkova et al., 2016; Hadj Taieb et al., 2020). As an example, 'car' and 'van' have high similarity and high relatedness, whereas 'car' and 'wheel' have lower similarity but still high relatedness. See Table 1 in Appendix A for a summary of major word similarity and relatedness datasets.

In most experimental studies, participants are asked to provide judgements about the similarity or relatedness of a set of word pairs, typically measured on an ordinal scale (Hill et al., 2015; Gerz et al., 2016). The averaged ratings are then compared to the cosine similarity of the corresponding word embeddings using a correlation coefficient (Vulić, Ponti, et al., 2020).

Numerous studies have followed this approach to investigate the relationship between human judgements and word embeddings, as summarised in Table 2 in Appendix A. These analyses have typically treated such judgements as non-contextual since word pairs are presented in isolation. However, we argue that this constitutes a failure to consider the implicit context provided by the second word in each word pair. Several studies have found that presenting words within the context of a sentence affects the manner in which humans make semantic judgments (Armendariz et al., 2020; Haber and Poesio, 2021). For example, humans interpret the word 'bank' differently when presented in a sentence about aircraft compared to when presented in a sentence about money (Trott and Bergen, 2021). However, to our knowledge this effect of context on human judgements has not been investigated in experimental datasets consisting solely of word pairs presented in the absence of additional context.

As such, building on previous suggestions (Bloch-Mullins, 2021) we hypothesise that when subjects are presented with two words absent further context, they assess the meaning of each word in the pair based on the implicit context of the other word in the pair. In the present study we investigate this hypothesis by evaluating the ability of word embeddings models to represent the meaning of antonym pairs and polysemous words. These were chosen as inherently relational semantic phenomena where context is most likely to affect human similarity judgements.

1.2 Antonymy

Antonyms are words that are 'opposite in meaning'. They provide a particular challenge for word similarity measures, since words like 'happy' and 'sad' are similar in that they both describe basic emotions, however since they are roughly opposite in meaning, they tend to be given low similarity ratings by humans (Lenci, 2018). It has proven difficult to define precisely what is meant by 'opposite meaning', with different subtypes and variations of antonymy proposed for different contexts or word types (Kotzor, 2021). In this study, we use a broad definition of antonymy by identifying verb pairs with varying degrees of contrasting or opposing meanings.

There are also conflicting views about the relationship between antonymy and similarity. If similarity is defined as the extent to which words are used in similar contexts, antonyms usually are identical in meaning except for the single dimension in which they have opposite values (Etcheverry and Wonsever, 2019). Conversely, if similarity is defined as the extent to which two words can be interchanged without loss of meaning, then antonyms have very low similarity (Kliegr and Zamazal, 2018). In practise, vector-space semantic models tend to give fairly high similarity ratings to both synonyms and antonyms (Nguyen et al., 2016), making it difficult to distinguish between these two relations in such models (Dou et al., 2018).

Various methods have been proposed to improve the representation of antonyms, including training a classifier over a set of word embeddings to distinguish antonyms from synonyms (Ali et al., 2019; Etcheverry and Wonsever, 2019), combining thesaurus or other knowledge-based information with word embeddings (Dou et al., 2018), and modifying standard word embeddings so that antonyms are maximally distant in similarity space (Nguyen et al., 2016; Samenko et al., 2020).

However, if the goal is to construct a comprehensive representation of word meanings, merely being able to distinguish antonyms from synonyms is insufficient. The fundamental difficulty appears to be that humans judge the similarity of antonyms differently than they judge other words, drawing upon background knowledge about the salient features for which antonyms have opposing values, and *using the context* provided by the presentation of words in a pair to judge the salience of these opposing features (Kotzor, 2021). The goal of the present study is to explore the role of context in more depth, investigating how antonym representation in word embedding models differs from human judgements.

1.3 Polysemy

A word is *polysemous* when it has multiple distinct but related meanings. For example, the verb 'count' can be used either to describe 'calculating using numbers' or 'being included as part of a group'. Vector-space models typically do not directly incorporate polysemy, as the usual approach is to learn a single word embedding vector for each word (Boleda, 2020; Camacho-Collados and Pilehvar, 2018). A major difficulty in incorporating polysemy into vector-space models is that there is no established method for distinguishing or enumerating different senses for a given polysemous word (Emerson, 2020), or in determining how much different senses overlap (Boleda, 2020). WordNet provides one commonly-used set of senses, though these have been criticised as being too finely-grained and lacking any clear structure (Palmer et al., 2007).

Polysemy also presents a problem for evaluating word embeddings, since humans may use the context of the second word in a pair to disambiguate a polysemous word. For instance, when presented with the pair 'bank' and 'river', participants may interpret 'bank' as relating to a riverbank, while when presented with 'bank' and 'loan', they are likely to interpret 'bank' as relating to a financial institution. This differs from word embeddings, which typically represent each word as a fixed vector regardless of which other word it is being compared to. As such, comparisons between human similarity judgements and word embedding similarities may be limited in accuracy by ignoring the contextual effects that affect human judgements.

One potential solution is to replace static word embeddings with *contextual word embeddings*, where instead of being fixed for all uses, word embeddings are dynamically modified based on the context in which they occur (Ethayarajh, 2019; Ranashinghe et al., 2019). Contextual embeddings can be constructed by transformer-based architectures, which have achieved impressive results at sense disambiguation and other investigations of word similarity (Garí Soler and Apidianaki, 2021). However, the highly flexible and contextual nature of transformer embeddings makes it unclear how exactly these contextual embeddings can be interpreted (Ethayarajh, 2019), and whether it even makes sense to analyse transformer embeddings from two different sentences as existing in the same semantic space (Mickus et al., 2019). Another problem is that contextual embeddings continuously vary in meaning across senses rather than forming discrete clusters, which differs from how polysemy is typically defined (Yenicelik et al., 2020).

An approach adopted by previous studies is to use traditional dictionaries to specify different word senses, combining definitions or example sentences with transformers to produce contextualised word embeddings for each sense (Ruzzetti et al., 2021; Tissier et al., 2017). The present study aims to build on previous research by using example sentences taken from dictionaries to construct word embeddings specialised for a particular context. We use these contextualised embeddings to investigate the extent to which polysemy reduces the ability of word embeddings to account for word similarity and relatedness datasets.

2 Methods

2.1 Analysis of word embeddings

In line with previous work, datasets of similarity and relatedness judgements were used to evaluate word embeddings by computing the Spearman correlation coefficient between human judgements and cosine similarities computed by word embedding models (Baroni et al., 2014). We used Spearman correlation as this is standard practise for evaluating ordinal human judgments of world similarity (Armendariz et al., 2020). See Table 3 in Appendix A for a full description of the embeddings used in this study.

Before computing correlations, the stimuli in the experimental datasets were pre-processed:

- All capitalisation was removed for consistency across datasets.
- Proper nouns were removed, as these have different semantic properties to regular nouns (Boleda et al., 2017).
- Word conjugations were altered to be in simple present infinitive form.
- Spelling was standardised to US spelling.

For the Tr9856 dataset, pre-processing removed so many sentences (mostly due to the presence of many proper nouns) that the modified dataset was renamed to Tr1058 to reflect that this is a small subset of the original dataset. This is indicated in Figure 5 in Appendix A.

For transformer models, decontextualised word embeddings were extracted by passing a single word to the transformer, averaging over multiple tokens when necessary. Contextualised transformer embeddings were computed using ERNIE as explained in Section 3.3. Embeddings were then normalised by dividing by the standard deviation in order to mitigate the problem of 'rogue dimensions', whereby a small number of dimensions account for most of the variation (Timkey and van Schijndel, 2021).

2.2 Verb antonymy

To assess the way antonyms are represented by vector-space semantics models, we manually identified antonym and near-antonym word pairs in the verb datasets, and computed the Spearman correlation between the relevant dataset and word embedding cosine similarities, both with and without these antonym pairs. The purpose of this analysis was to determine whether antonyms are represented differently compared to other word pairs. Verb datasets were chosen for this task as it was observed that the main available noun datasets contained relatively few pairs of antonyms.

2.3 Verb polysemy

To measure the effect of polysemy on semantic similarity judgements, contextual transformer embeddings were reduced to static embeddings using procedures developed previously (Bommasani et al., 2020; Soper and Koenig, 2022). The key idea of this approach is to use a transformer to compute embeddings of the target word in a given sentence context, and then average over multiple sentences to produce a context-sensitive static word embedding. By altering the sentences used to produce the contextual embedding, the resulting static word embeddings can be tailored to particular senses of the target word.

This method was applied using the ERNIE transformer, as it performed similarly to other leading transformer models while also being small and computationally tractable (see Section 3). We produced four distinct contextualised embeddings to test a variety of methods for incorporating contextual information relevant to polysemy. These four methods differ in the amount and quality of contextual information provided, as explained below and summarised in Table 4 of Appendix A.

Note that in order to disentangle the effects of antonymy from those of polysemy, subsequent analyses are performed on the verb datasets with antonyms removed.

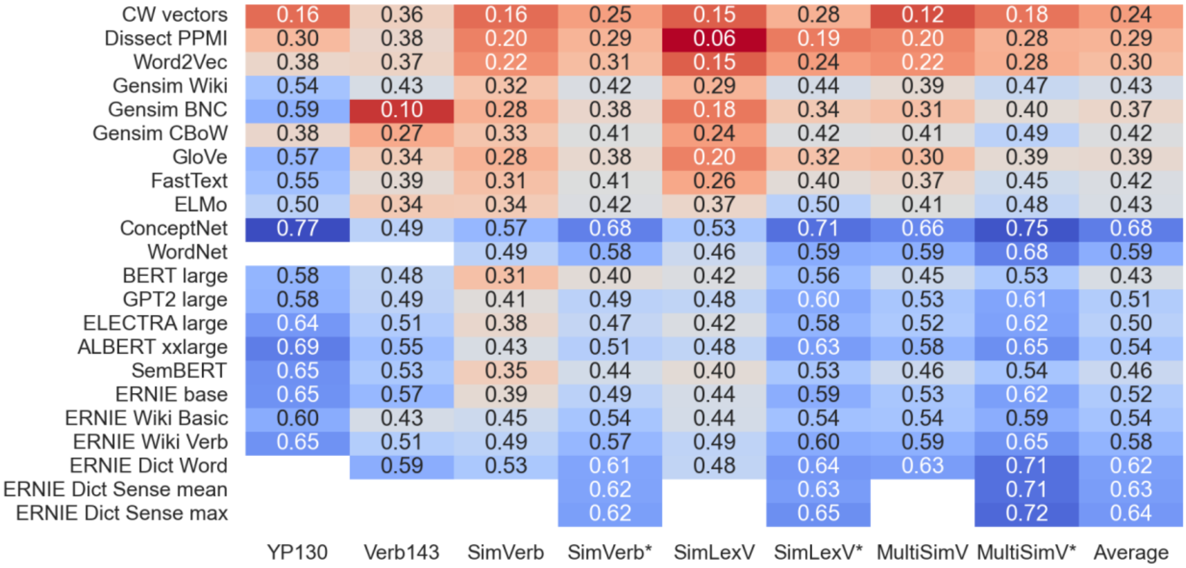
The *ERNIE Wikipedia Basic embeddings* were computed from a set of sample sentences, each containing the target word, from a custom Wikipedia corpus of 10,000 articles. These were selected using a Wikipedia list of key articles, in order to provide sentences covering a diverse range of topics while also keeping the corpus a manageable size. The text of each article was imported using a Wikipedia Python API, and then processed to remove image captions, tables, citations, and other metadata. The result was a corpus consisting of 2 million sentences.

Word embeddings were then computed by finding sentences containing each target word within the corpus, up to a maximum of 100 sentences per target word to avoid wasting computational time for very common words. To ensure a match, words in each sentence were lemmatised using the nltk WordNetLemmatizer (Loper and Bird, 2002). Contextualised embeddings were computed for each matching sentence using ERNIE, and the token embeddings of the target word averaged over all sample sentences for that word. A lemmatiser was used to automatically conjugate each word in the sentence as a noun or verb to match the target. In cases in which the target word corresponded to more than one transformer token, the embeddings for each token were averaged.

The *ERNIE Wikipedia Verb embeddings* were computed in the same way, except that words in the sample sentences were now always lemmatised as verbs, thus ensuring the sample sentences reflected cases when the target word was used as a verb. This provides a simple method for controlling for polysemy of words that are used as both nouns and as verbs. A similar approach was taken for nouns, though little gain in performance was observed (see Figure 5 in Appendix A), so subsequent analysis focused only on verb polysemy.

The *ERNIE Dictionary Word embeddings* were calculated from sample sentences extracted for each target word from the Oxford Learner's Dictionary (Turnbull et al., 2010). It was hypothesised that using sentences tailored to providing examples of usage for each word would provide better disambiguation of polysemy than a large collection of assorted Wikipedia sentences. In this case, example sentences were pooled together regardless of the sense they corresponded to.

Finally, the *ERNIE Dictionary Sense embeddings* were constructed by manually separating example dictionary sentences into up to six different senses for each target word. This was performed by the authors, using the Oxford Learner's Dictionary and Longman Dictionary of Contemporary English Online (Pearson, 2023) as guides. Senses that shared a common grouping or heading in these dictionaries were generally combined, as



	YP130	Verb143	SimVerb	SimVerb*	SimLexV	SimLexV*	MultiSimV	MultiSimV*	Average
CW vectors	0.16	0.36	0.16	0.25	0.15	0.28	0.12	0.18	0.24
Dissect PPMI	0.30	0.38	0.20	0.29	0.06	0.19	0.20	0.28	0.29
Word2Vec	0.38	0.37	0.22	0.31	0.15	0.24	0.22	0.28	0.30
Gensim Wiki	0.54	0.43	0.32	0.42	0.29	0.44	0.39	0.47	0.43
Gensim BNC	0.59	0.10	0.28	0.38	0.18	0.34	0.31	0.40	0.37
Gensim CBoW	0.38	0.27	0.33	0.41	0.24	0.42	0.41	0.49	0.42
GloVe	0.57	0.34	0.28	0.38	0.20	0.32	0.30	0.39	0.39
FastText	0.55	0.39	0.31	0.41	0.26	0.40	0.37	0.45	0.42
ELMo	0.50	0.34	0.34	0.42	0.37	0.50	0.41	0.48	0.43
ConceptNet	0.77	0.49	0.57	0.68	0.53	0.71	0.66	0.75	0.68
WordNet			0.49	0.58	0.46	0.59	0.59	0.68	0.59
BERT large	0.58	0.48	0.31	0.40	0.42	0.56	0.45	0.53	0.43
GPT2 large	0.58	0.49	0.41	0.49	0.48	0.60	0.53	0.61	0.51
ELECTRA large	0.64	0.51	0.38	0.47	0.42	0.58	0.52	0.62	0.50
ALBERT xxlarge	0.69	0.55	0.43	0.51	0.48	0.63	0.58	0.65	0.54
SemBERT	0.65	0.53	0.35	0.44	0.40	0.53	0.46	0.54	0.46
ERNIE base	0.65	0.57	0.39	0.49	0.44	0.59	0.53	0.62	0.52
ERNIE Wiki Basic	0.60	0.43	0.45	0.54	0.44	0.54	0.54	0.59	0.54
ERNIE Wiki Verb	0.65	0.51	0.49	0.57	0.49	0.60	0.59	0.65	0.58
ERNIE Dict Word		0.59	0.53	0.61	0.48	0.64	0.63	0.71	0.62
ERNIE Dict Sense mean				0.62		0.63		0.71	0.63
ERNIE Dict Sense max				0.62		0.65		0.72	0.64

Figure 1: Spearman correlations between embedding models (rows) and verb subsets of experimental datasets (columns). An asterisk denotes exclusion of antonyms from the dataset. SimLexV indicates the SimLexVerb dataset, and likewise for MultiSimV. Average is weighted by dataset size. Note that ERNIE Dict embeddings were only computed for verb datasets with antonyms removed.

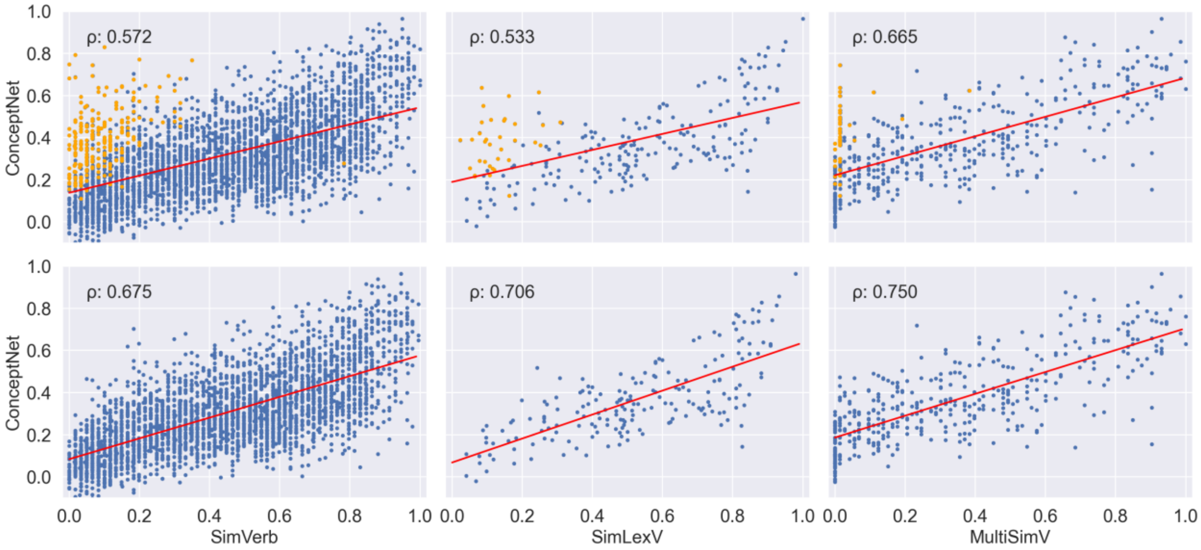


Figure 2: Effects of removing antonyms (shown in orange) from the SimVerb (left), SimLexVerb (centre), and MultiSimVerb (right) datasets, with experimental similarity judgements on the plotted on the horizontal axis against ConceptNet cosine similarities (vertical axis).

were instances where one sense is a subset of another. Rare senses containing few example sentences were excluded to focus on more common uses. We anticipated that manual consolidation of senses would improve the resulting word embeddings by allowing sample sentences to combined from the Oxford, Longman, and Collins Online Dictionaries (Collins, 2023).

Furthermore, while the previous methods pool all senses together, this approach produces embeddings for each individual sense. Such sense embeddings can be compared to the experimental datasets either by taking the average (*mean*) over all senses, or the maximum (*max*) similarity over all pairwise sense comparisons. We consider the maximum pairwise similarity because we hypothesise that participants may be sensitive to the most similar senses of two target words. Both results are shown in Figure 1.

3 Results

3.1 Analysis of word embeddings

To identify the best-performing embeddings to use in subsequent analysis, Spearman correlation coefficients between each word embedding model and the similarity ratings of all verb-based experimental datasets were computed (Figure 1). For comparison, the results for noun datasets are given in Appendix A. For both nouns and verbs, ConceptNet embeddings consistently show higher correlations with human judgements over almost all datasets. Transformers typically perform better than count- and predict-based embedding models, with GPT-2, ALBERT xxlarge, and ERNIE showing the highest correlations. We also observed some clustering of models, with static and contextualised embeddings being more similar to each other than to different types of models, as shown in Figures 5 and 6 in Appendix A.

Given its superior performance, ConceptNet was chosen as the focus of subsequent analysis of antonyms, for which static embeddings are sufficient. ERNIE was selected as a representative transformer for analysis of polysemy, as this required computing contextual embeddings which is not possible with ConceptNet.

3.2 Verb antonymy

Figure 2 shows scatterplots of ConceptNet cosine similarities against three verb-based datasets. The difference between the top and bottom rows of the subplots shows the effect of removing antonyms, which are seen to disproportionally cluster in the top left of the scatterplots. Removal of the antonyms substantially improves the fit between experimental and word embedding similarities, increasing the correlation on the SimVerb dataset from 0.572 to 0.675, from 0.533 to 0.706 on the SimLexVerb dataset, and from 0.665 to 0.750 on the MultiSimVerb dataset. This shows that humans represent the relations between antonyms very differently than do the ConceptNet embed-

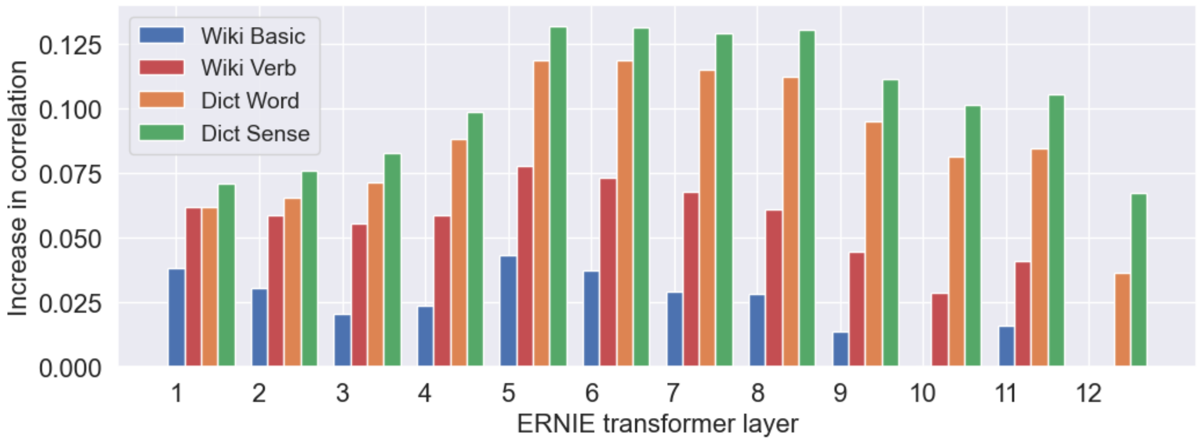


Figure 3: Comparison of the increase in correlation with SimVerb dataset relative to the ERNIE base model for the Wikipedia Basic, Wikipedia Verb, and Dictionary Word, and Dictionary Sense max embeddings. Correlations increase as more specific and fine-grained contextual information is added.

dings. Similar results were observed for ERNIE embeddings, as shown in Figure 8 of Appendix A.

3.3 Verb polysemy

Figure 3 shows the results of incorporating contextual information from corpus and dictionary sources by reducing contextual ERNIE embeddings to static embeddings, as outlined in Section 2.3. Relative to the layer 5 ERNIE base embeddings, Wikipedia Basic embeddings increase the correlation with human judgements in the SimVerb dataset by 5 percentage points, Wikipedia Verb embeddings by 8, Dictionary Word embeddings by 12, and the Dictionary Sense max embeddings by 13 percentage points.

We also examined the effect of transformer layers on the correlation with human judgements. Consistent with previous studies (Caucheteux et al., 2021; Timkey and van Schijndel, 2021), the best results are found around the middle layers of the transformer, indicating that later layers progressively incorporate relevant contextual infor-

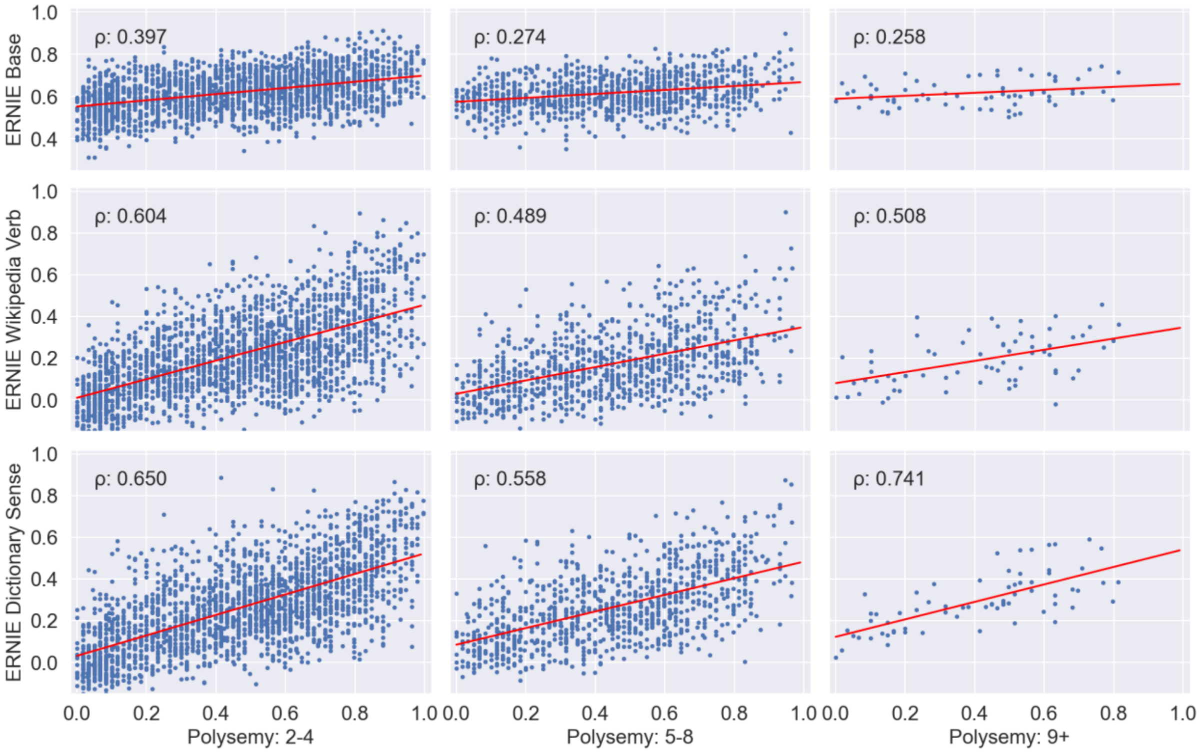


Figure 4: Comparison of the Spearman correlations of the SimVerb dataset with ERNIE Base (top), ERNIE Wikipedia Verb (middle), and ERNIE Dictionary Sense max (bottom) embeddings, split by polysemy score.

mation, but only up to a certain point. Henceforth we discuss results from layer 5.

To further investigate the effect of polysemy on the accuracy of word embeddings, SimVerb word pairs were grouped according to their total polysemy score, defined simply as the sum of the number of senses for both words in each pair. Senses were differentiated for each word during the construction of the ERNIE Dictionary Sense embeddings, as outlined in Section 2.3. As shown in Figure 4, and similarly to the results in Figure 3, the correlation with human ratings increases as more specific and fine-grained contextual information is added, with Wikipedia Verb embeddings showing higher correlations than the base model, and Dictionary Sense embeddings showing higher correlations still.

Furthermore, we found that correlations increase most for highly polysemous word pairs. Relative to the uncontextualised ERNIE base, the ERNIE Dictionary Sense embeddings increase correlations by 0.25 for the least polysemous, 0.29 for moderately polysemous, and 0.48 for the most polysemous word pairs. These results indicate that, while static word embeddings struggle to accurately represent the meaning of highly polysemous words, transformer models which incorporate contextual information perform much better.

4 Discussion

This paper has highlighted significant differences between the manner in which humans and word embedding models represent the meaning of antonyms. While it has long been known that word embeddings perform poorly in predicting antonym similarity judgements (Dou et al., 2018), we have shown the reason for this is that antonyms are given consistently low similarity ratings by humans but moderate to high cosine similarities by embedding models. This effect is consistent across datasets and large in magnitude, reducing correlations by 0.10-0.15, even though antonym or near-antonym word pairs only account for about 10% of each dataset.

Previous research has sought to rectify the low accuracy of word embedding models on antonyms by adding constraints to artificially pull antonyms further apart in semantic space (Mrkšić et al., 2016, Biesialska et al., 2020). However, we argue that this may be inappropriate, because when humans make similarity judgments between words, they may not be performing an analogous task to computing the cosine similarity between the corresponding embeddings. If this is the case, then the failure of word embedding cosine similarities to match human similarity judgments for antonyms should be interpreted as a limitation of the evaluation method, not a flaw of the word embeddings as a model of word meaning.

Relatedly, it has been argued that antonyms should have cosine similarities close to the smallest possible value of -1 (Samenko et al., 2020). In practise, however, negative cosine similarities occur mostly between unrelated words rather than antonyms, with small absolute values (up to around -0.1 for ConceptNet). This is likely because computing cosine similarity averages across all features whether salient or not, thereby computing 'property overlap' (Erk, 2016). Since antonyms share most features in common, this results in a high cosine similarity.

Why then do humans rate antonyms as having very low semantic similarity? One potential explanation is that the salience of the semantic features of a word varies depending on the context in which the word is used. This has been observed for human judgements of noun combination tasks (Bock and Clifton, 2000) and feature verification tasks (Montefinese et al., 2014). Such findings are consistent with our hypothesis that, when assessing the similarity of two antonyms, humans judge the dimension of meaning in which the two words differ as the most salient, and hence rate overall semantic similarity as low. This would also explain earlier findings that humans rate antonyms almost as similar as synonyms when asked to rate features separately, rather than providing an overall similarity score (Crutch et al., 2012).

These considerations highlight the need for a new method which enables more consistent and informative comparison between human similarity judgements and cosine similarities for antonyms. Unfortunately, in this study we were unable to develop such a method. We experimented with simple methods such as providing both words to the ERNIE transformer and extracting the contextualised embeddings of each, but this yielded no useful results. Further improvements will likely require identifying which particular features are most salient for assessment of antonyms, in line with several previous studies (Ali et al., 2019; Nguyen et al., 2016). In addition, our brief treatment of antonymy has not discussed important issues such as adjectival antonyms or the effects of discourse context on negation (Kruszewski et al., 2016). We leave such considerations for future work.

In this study we also found that polysemy significantly reduces the accuracy of word embeddings in describing the similarity of verbs. The dramatic increase in the correlation of ERNIE embeddings with human judgements when contextual information was incorporated (see Figure 4) is evidence that the quality of the embeddings is significantly impaired by the inability to properly distinguish different word senses. Our results are consistent with a strategy whereby humans assess the similarity of two words using an implicit context that maximises the aspects of meaning they share, ignoring any additional polysemous meanings. This would explain why providing ERNIE with additional information about context, like parts of speech and example sentences, improves the correlation with human judgments.

Our results also highlight the value of using contextual information from lexical dictionaries to augment contextual word embeddings. In particular, ERNIE Dictionary Sense max embeddings increase the correlation by about 5 percentage points for the full SimVerb dataset (excluding antonyms), and about 23 percentage points for the most polysemous word pairs. Similar increases in correlation were observed from the simpler automated method of aggregating all dictionary senses together, as used in the ERNIE Dictionary Word embeddings. We hypothesise that these improvements arise because example dictionary sentences represent common uses of verb, which may reflect the way that humans judge word similarities when asked to judge two words without context.

A different approach to control for the effects of polysemy used in several past studies is to ask participants to judge the similarity of words in the context of a specific sentence, thereby allowing for clearer sense disambiguation (Armendariz et al., 2020; Camacho-Collados and Pilehvar, 2018; Haber and Poesio, 2021). However, it is difficult to ensure that participants do not simply judge the overall similarity of the sentences, or conversely ignore the context and consider the target words in isolation. Furthermore, contextualised word embeddings are more difficult to interpret than static embeddings since they only apply to the word in a specific precise context. Given that a concept is typically defined as a mental representation that is reasonably invariant across contexts (Laurence and Margolis, 1999; Musz and Thompson-Schill, 2018), highly context-specific word embeddings are arguably of less value as cognitive models of concepts. As such, we believe there is also value in incorporating contextual information to improve static embeddings of polysemous words, as we have shown can be done by using example sentences from lexical dictionaries.

In this paper we have focused on ERNIE embeddings, as they showed superior performance over competing models that are purely text-based. The performance of ConceptNet embeddings provide an additional baseline that also incorporates expert linguistic knowledge. The results corroborates previous studies which found that adding expert knowledge can improve the performance of embeddings derived from word cooccurrence statistics (Peters et al., 2019; Xu et al., 2021; Zhang et al., 2020). Nevertheless, transformer models like ERNIE use much larger training corpuses and have more parameters than ConceptNet (Devlin et al., 2019), so the fact that ConceptNet still outperforms all transformer embeddings is a notable finding. However, we do not seek to determine the effect of specific architectural choices or hyperparameters, as such analysis has been conducted in previous studies (Baroni et al., 2014; Lapesa and Evert, 2014; Liu et al., 2021).

5 Conclusion

In this study we have highlighted the problems of ignoring the implicit context in which humans make word similarity judgements. Our results show that word meaning is judged in a context-dependent manner which decontextualised word embeddings struggle to adequately capture. Future work focused on improving embeddings may require better datasets specifically focused on evaluating how humans rate the similarity of different forms of antonyms. Also important is improving the representation of polysemy, which we have shown is possible by combining contextualised embeddings with carefully collated data from dictionaries and other knowledge banks. Our analysis has primarily focused on verbs, and so further work focusing on nouns is also needed. Overall, much work remains to be done to enhance the ability of vector-space semantic models to describe a wide range of semantic phenomena.

References

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. 'Antonym-synonym classification based on new sub-space embeddings.' In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6204-11.

Felipe Almeida and Geraldo Xexéo. 2019. 'Word embeddings: A survey', *arXiv preprint arXiv:1901.09069*.

Carlos Santos Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, Marko Robnik-Šikonja, Mark Granroth-Wilding, and Kristiina Vaik. 2020. 'CoSimLex: A resource for evaluating graded word similarity in context', In *Proceedings of the 12th International Conference on Language Resources and Evaluation,* pages 5878-86.

Simon Baker, Roi Reichart, and Anna Korhonen. 2014. 'An Unsupervised Model for Instance Level Subcategorization Acquisition.' *EMNLP*, 278-89.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. 'Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors.' In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238-47.

Magdalena Biesialska, Bardia Rafieian, and Marta R. Costa-jussà. 2020. Enhancing Word Embeddings with Knowledge Extracted from Lexical Resources. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 271–278.

Corinne L Bloch-Mullins. 2021. 'Similarity Reimagined (with Implications for a Theory of Concepts)', *Theoria*, 87: 31-68.

Jeannine S Bock and Charles Clifton. 2000. 'The role of salience in conceptual combination', *Memory & Cognition*, 28: 1378-86.

Gemma Boleda. 2020. 'Distributional semantics and linguistic theory', *Annual review of Linguistics*, 6: 213-34.

Gemma Boleda, Abhijeet Gupta, and Sebastian Padó. 2017. 'Instances and concepts in distributional space.' In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 79-*85.*

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. 'Interpreting pretrained contextualized representations via reductions to static embeddings.' In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758-81.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. 'Distributional semantics in technicolor.' In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136-45.

Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2018. 'From word to sense embeddings: A survey on vector representations of meaning', *Journal of Artificial Intelligence Research*, 63: 743-88.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. 'Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity.' Association for Computational Linguistics.

Charlotte Caucheteux, Alexandre Gramfort, and Jean-Remi King. 2021. 'Disentangling syntax and semantics in the brain with deep networks.' *International Conference on Machine Learning*, 1336-48. PMLR.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. 'Intrinsic evaluation of word vectors fails to predict extrinsic performance.' In *Proceedings of the 1st workshop on evaluating vector-space representations for NLP*, pages 1-6.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. 'ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators', *ArXiv*, abs/2003.10555.

Stephen Clark. 2015. 'Vector space models of lexical meaning', *The Handbook of Contemporary semantic theory*: 493-522.

Collins. 2023. 'Collins online dictionary'. https://www.collinsdictionary.com/.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. 'Natural language processing (almost) from scratch', *Journal of machine learning research*, 12: 2493–537.

Sebastian J Crutch, Paul Williams, Gerard R Ridgway, and Laura Borgenicht. 2012. 'The role of polarity in antonym and synonym conceptual knowledge: Evidence from stroke aphasia and multidimensional ratings of abstract words', *Neuropsychologia*, 50: 2636-44.

Simon De Deyne, Amy Perfors, and Daniel J Navarro. 2016. 'Predicting human similarity judgments with distributional models: The value of word associations.' In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1861-70.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', *ArXiv*, abs/1810.04805.

Zehao Dou, Wei Wei, and Xiaojun Wan. 2018. 'Improving word embeddings for antonym detection using thesauri and sentiwordnet.' *CCF international conference on natural language processing and Chinese computing*, 67-79. Springer.

Guy Emerson. 2020. 'What are the Goals of Distributional Semantics?', *arXiv preprint arXiv:2005.02982*.

Katrin Erk. 2016. 'What do you know about an alligator when you know the company it keeps?', *Semantics and Pragmatics*, 9: 17-1-63.

Mathias Etcheverry and Dina Wonsever. 2019. 'Unraveling antonym's word vectors through a siamese-like network.' In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3297-307.

Kawin Ethayarajh. 2019. 'How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings', In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* pages *55-65, Hong Kong, China.*

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. 'Placing search in context: The concept revisited.' In *Proceedings of the 10th international conference on World Wide Web*, pages 406-14.

Aina Garí Soler and Marianna Apidianaki. 2021. 'Let's play mono-poly: BERT can reveal words' polysemy level and partitionability into senses', *Transactions of the Association for Computational Linguistics*, 9: 825-44.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. 'Simverb-3500: A large-scale evaluation set of verb similarity', In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182.

Fritz Günther, Luca Rinaldi, and Marco Marelli. 2019. 'Vector-space models of semantic representation from a cognitive perspective: A discussion of common misconceptions', *Perspectives on Psychological Science*, 14: 1006-33.

Janosch Haber and Massimo Poesio. 2021. 'Patterns of Lexical Ambiguity in Contextualised Language Models', *arXiv preprint arXiv:2109.13032*.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. 'Large-scale learning of word relatedness with constraints.' In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406-14.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. 'Simlex-999: Evaluating semantic models with (genuine) similarity estimation', *Computational Linguistics*, 41: 665-95.

Tomáš Kliegr and Ondřej Zamazal. 2018. 'Antonyms are similar: Towards paradigmatic association approach to rating similarity in SimLex-999 and WordSim-353', *Data & Knowledge Engineering*, 115: 174-93.

Germán Kruszewski, Denis Paperno, Raffaella Bernardi, and Marco Baroni. 2016. 'There is no logical negation here, but there are alternatives: Modeling conversational negation with distributional semantics.' *Computational Linguistics,* 42: 637-660.

Sandra Kotzor. 2021. *Antonyms in Mind and Brain: Evidence from English and German* (Routledge).

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. 'Word vectors, reuse, and replicability: Towards a community repository of large-text resources.' In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271-76. Linköping University Electronic Press.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. 'ALBERT: A Lite BERT for Self-supervised Learning of Language Representations', *ArXiv*, abs/1909.11942.

Gabriella Lapesa, and Stefan Evert. 2014. 'A large scale evaluation of distributional semantic models: Parameters, interactions and model selection', *Transactions of the Association for Computational Linguistics*, 2: 531-46.

Stephen Laurence, and Eric Margolis. 1999. 'Concepts and cognitive science', *Concepts: core readings*, 3: 81.

Alessandro Lenci. 2018. 'Distributional models of word meaning', *Annual review of Linguistics*, 4: 151-71.

Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2021. 'A comprehensive comparative evaluation and analysis of Distributional Semantic Models', *arXiv preprint arXiv:2105.09825*.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. 'Improving distributional similarity with lessons learned from word embeddings', *Transactions of the Association for Computational Linguistics*, 3: 211-25.

Ran Levy, Liat Ein Dor, Shay Hummel, Ruty Rinott, and Noam Slonim. 2015. 'Tr9856: A multi-word term relatedness benchmark.' In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 419-24.

Antonio Lieto, Antonio Chella, and Marcello Frixione. 2017. 'Conceptual spaces for cognitive architectures: A lingua franca for different levels of representation', *Biologically inspired cognitive architectures*, 19: 1-9.

Liyuan Liu, Jialu Liu, and Jiawei Han. 2021. 'Multi-head or single-head? an empirical comparison for transformer training', *arXiv preprint arXiv:2106.09650*.

Edward Loper and Steven Bird. 2002. 'Nltk: The natural language toolkit', *arXiv preprint cs/0205028*.

Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. 'Better word representations with recursive neural networks for morphology.' In *Proceedings of the seventeenth conference on computational natural language learning*, pages 104-13.

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees Van Deemter. 2019. 'What do you mean, BERT? Assessing BERT as a Distributional Semantics Model', *arXiv preprint arXiv:1911.05758*.

George A Miller, and Walter G Charles. 1991. 'Contextual correlates of semantic similarity', *Language and cognitive processes*, 6: 1-28.

Nikola Mrkšić, Diarmuid Ó. Séaghdha, Blaise Thomson, Milica Gasic, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. 'Counter-fitting Word Vectors to Linguistic Constraints.' In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 142-148.

Maria Montefinese, Ettore Ambrosini, Beth Fairfield, and Nicola Mammarella. 2014. 'Semantic significance: a new measure of feature salience', *Memory & Cognition*, 42: 355-69.

Elizabeth Musz and Sharon L Thompson-Schill. 2018. 'Finding Concepts in Brain Patterns.' In *The oxford handbook of neurolinguistics*, New York, NY: Oxford University Press.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. 'Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction', In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* pages 454–459.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. 2007. 'Making fine-grained and coarse-grained sense distinctions, both manually and automatically', *Natural Language Engineering*, 13: 137-63.

Pearson. 2023. 'Longman Dictionary of Contemporary English Online'. https://www.ldoceonline.com/.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. 'WordNet:: Similarity-Measuring the Relatedness of Concepts.' *AAAI*, 25-29.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. 'Glove: Global vectors for word representation.' In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-43.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. 'Knowledge enhanced contextual word representations', In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing,* pages 43–54.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 'Deep Contextualized Word Representations.' *North American Chapter of the Association for Computational Linguistics*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. 'Language Models are Unsupervised Multitask Learners.' *OpenAI blog 1*, no. 8: 9.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. 'A word at a time: computing word relatedness using temporal semantic analysis.' In *Proceedings of the 20th international conference on World wide web*, pages 337-46.

Tharindu Ranashinghe, Constantin Orasan, and Ruslan Mitkov. 2019. 'Enhancing unsupervised sentence similarity methods with deep contextualised word representations.' RANLP.

João Rodrigues, Ruben Branco, Joao Silva, Chakaveh Saedi, and António Branco. 2018. 'Predicting brain activation with WordNet embeddings.' In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 1-5.

Herbert Rubenstein and John B Goodenough. 1965. 'Contextual correlates of synonymy', *Communications of the ACM*, 8: 627-33.

Elena Sofia Ruzzetti, Leonardo Ranaldi, Michele Mastromattei, Francesca Fallucchi, and Fabio Massimo Zanzotto. 2022. 'Lacking the embedding of a word? look it up into a traditional dictionary', *2022. Lacking the Embedding of a Word? Look it up into a Traditional Dictionary. In Findings of the Association for Computational Linguistics: ACL 2022,* pages 2651–2662.

Chakaveh Saedi, António Branco, João Rodrigues, and Joao Silva. 2018. 'Wordnet embeddings.' In *Proceedings of the third workshop on representation learning for NLP*, pages 122-31.

Igor Samenko, Alexey Tikhonov, and Ivan P Yamshchikov. 2020. 'Synonyms and antonyms: Embedded conflict', *ArXiv, abs/2004.12835*.

Yong Shi, Yuanchun Zheng, Kun Guo, Wei Li, and Luyao Zhu. 2018. 'Word similarity fails in multiple sense word embedding.' *International Conference on Computational Science*, 489-98. Springer.

Elizabeth Soper and Jean-Pierre Koenig. 2022. 'When Polysemy Matters: Modeling Semantic Categorization with Word Embeddings.' In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 123-31.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. 'Conceptnet 5.5: An open multilingual graph of general knowledge.' *Thirty-first AAAI conference on artificial intelligence*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, and Yuxiang Lu. 2021. 'Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation', *arXiv preprint arXiv:2107.02137*.

William Timkey and Marten van Schijndel. 2021. 'All Bark and No Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality'. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4527–46.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2017. 'Dict2vec: Learning word embeddings using lexical dictionaries.' In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254-63.

Jatin Karthik Tripathy, Sibi Chakkaravarthy Sethuraman, Meenalosini Vimal Cruz, Anupama Namburu, P Mangalraj, and Vaidehi Vijayakumar. 2021. 'Comprehensive analysis of embeddings and pre-training in NLP', *Computer Science Review*, 42: 100433.

Sean Trott and Benjamin Bergen. 2021. 'RAW-C: Relatedness of Ambiguous Words--in Context (A New Lexical Resource for English)', In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7077-87.

Joanna Turnbull, D Lea, D Parkinson, P Phillips, B Francis, S Webb, V Bull, and M Ashby. 2010. 'Oxford advanced learner's dictionary'. https://www.oxfordlearnersdictionaries.com/.

Ivan Vulić, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, and Thierry Poibeau. 2020. 'Multi-SimLex: A Large-Scale Evaluation of Multilingual and Crosslingual Lexical Semantic Similarity', *Computational Linguistics*, 46: 847-97.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. 'Probing pretrained language models for lexical semantics', In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* pages 7222-40.

Shaonan Wang, Jiajun Zhang, Haiyan Wang, Nan Lin, and Chengqing Zong. 2020. 'Fine-grained neural decoding with distributed word representations', *Information Sciences*, 507: 256-72.

Gijs Wijnholds, and Mehrnoosh Sadrzadeh. 2019. 'Evaluating Composition Models for Verb Phrase Elliptical Sentence Embeddings.' 261-71. Minneapolis, Minnesota: Association for Computational Linguistics.

Wenwen Xu, Mingzhe Fang, Li Yang, Huaxi Jiang, Geng Liang, and Chun Zuo. 2021. 'Enabling Language Representation with Knowledge Graph and Structured Semantic Information.' *2021 International Conference on Computer Communication and Artificial Intelligence (CCAI)*, 91-96. IEEE.

Dongqiang Yang, and David M. W. Powers. 2006. 'Verb similarity on the taxonomy of WordNet.'

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. 'How does BERT capture semantics? A closer look at polysemous words.' In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156-62.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. 'Recent trends in deep learning based natural language processing', *IEEE Computational Intelligence magazine*, 13: 55-75.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. 'Semantics-aware BERT for language understanding.' In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9628-35.

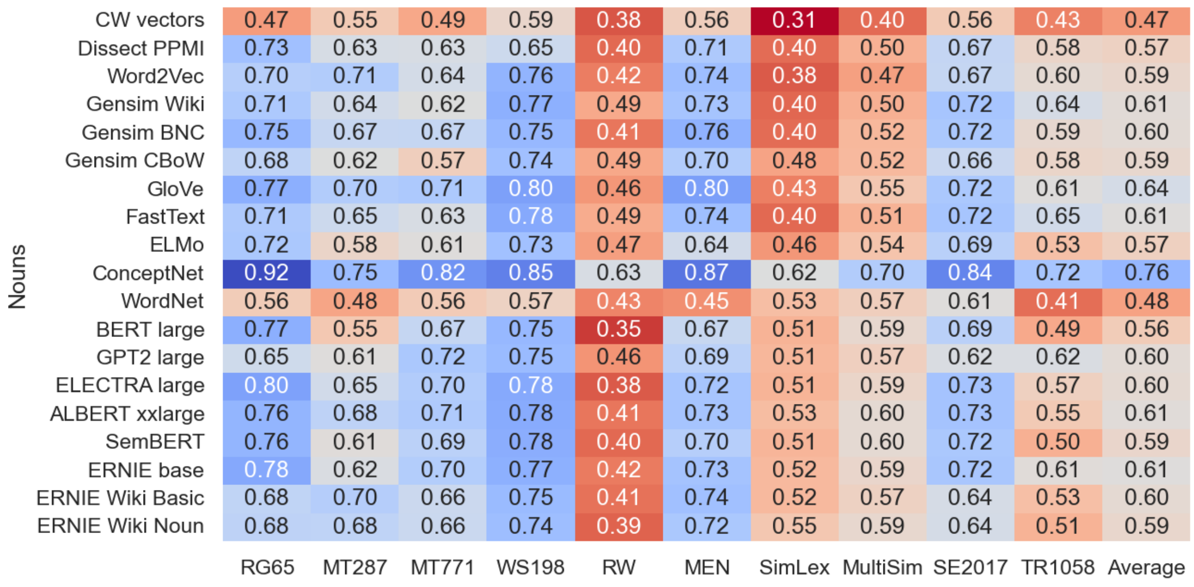
A Additional Figures and Tables

Model Name	Number Word Pairs	Part of Speech	Data Type	Citation
RG65	65	Nouns	Similarity	Rubenstein and Goodenough (1965)
WordSim-353	353	Nouns	Relatedness	Finkelstein et al. (2001)
SimLex-999	999	Mixed	Similarity	Hill et al. (2015)
YP-130	130	Verbs	Similarity	Yang and Powers (2006)
Verb-143	143	Verbs	Similarity	Baker et al. (2014)
Multi-SimLex	1,888	Mixed	Similarity	Vulić, Baker, et al. (2020)
SimVerb-3500	3,500	Verbs	Similarity	Gerz et al. (2016)
MEN	3,000	Nouns	Relatedness	Bruni et al. (2012)
MTurk-287	287	Nouns	Relatedness	Radinsky et al. (2011)
MTurk-771	771	Nouns	Relatedness	Halawi et al. (2012)
Tr9856	9,856	Nouns	Relatedness	Levy, Dor, et al. (2015)
SemEval-2017	500	Nouns	Relatedness	Camacho-Collados et al. (2017)
Stanford-RW	2,034	Mixed	Similarity	Luong et al. (2013)

Table 1: Summary of word similarity and relatedness experimental datasets.

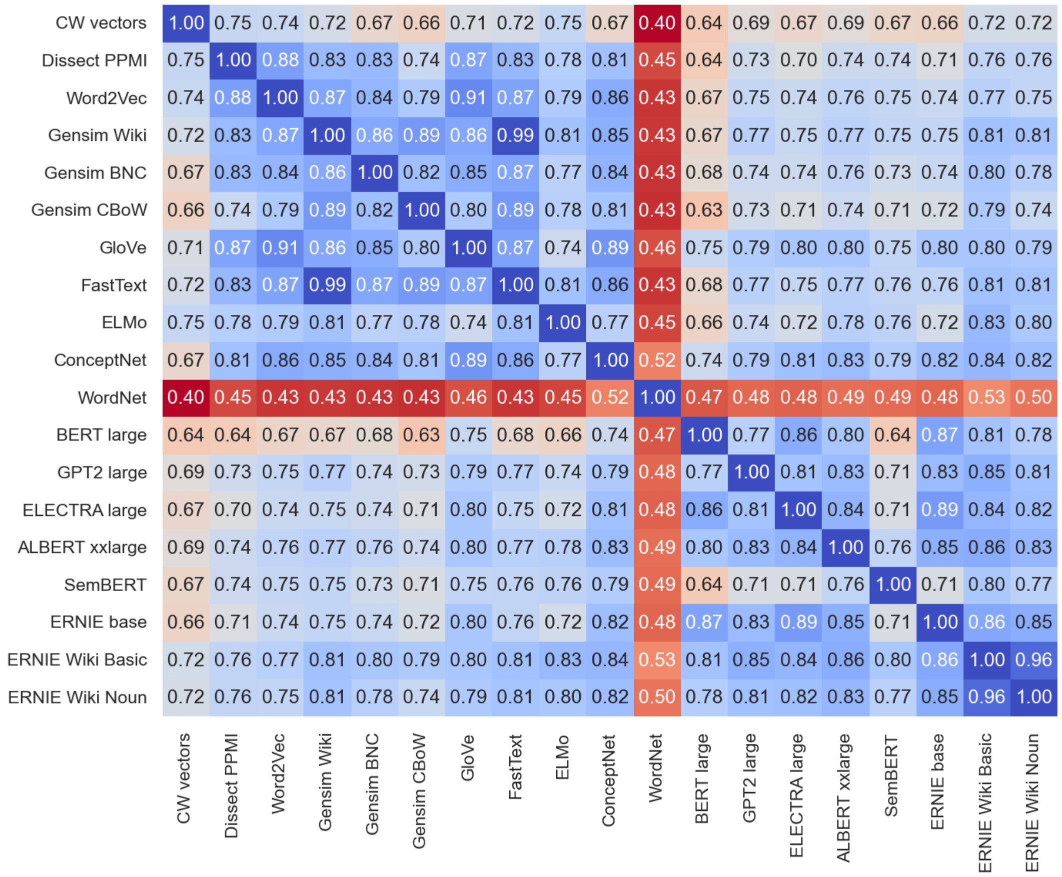
Models Tested	WS353	SL999	MEN	MT287	MT771	RW	SV	Citation
PMI model, Skip-gram, GloVe	.71	.43	.78	.69		.51		Levy, Goldberg, et al. (2015)
PMI model, CBOW	.79	.43	.79	.78	.71			De Deyne et al. (2016)
Skip-gram	.70	.34	.73	.66	.61	.40		Chiu et al. (2016)
Count-based, CBOW, GloVe, FastText	.70	.40	.78					Wijnholds and Sadrzadeh (2019)
BERT, GPT-2, RoBERTa, XLNet, DistilBERT	.72	.55					.45	Bommasani et al. (2020)
LSA, LDA, CBOW, skip-gram, GloVe, RI, FastText, BERT	.71	.49	.79	.71		.48	.41	Lenci et al. (2021)

Table 2: Summary of previous analyses of word embedding models, showing the highest Spearman correlation recorded by each paper for each analysed dataset. WS: WordSim, SL: SimLex, MT: MTurk, RW: Stanford-RW, SV: SimVerb.



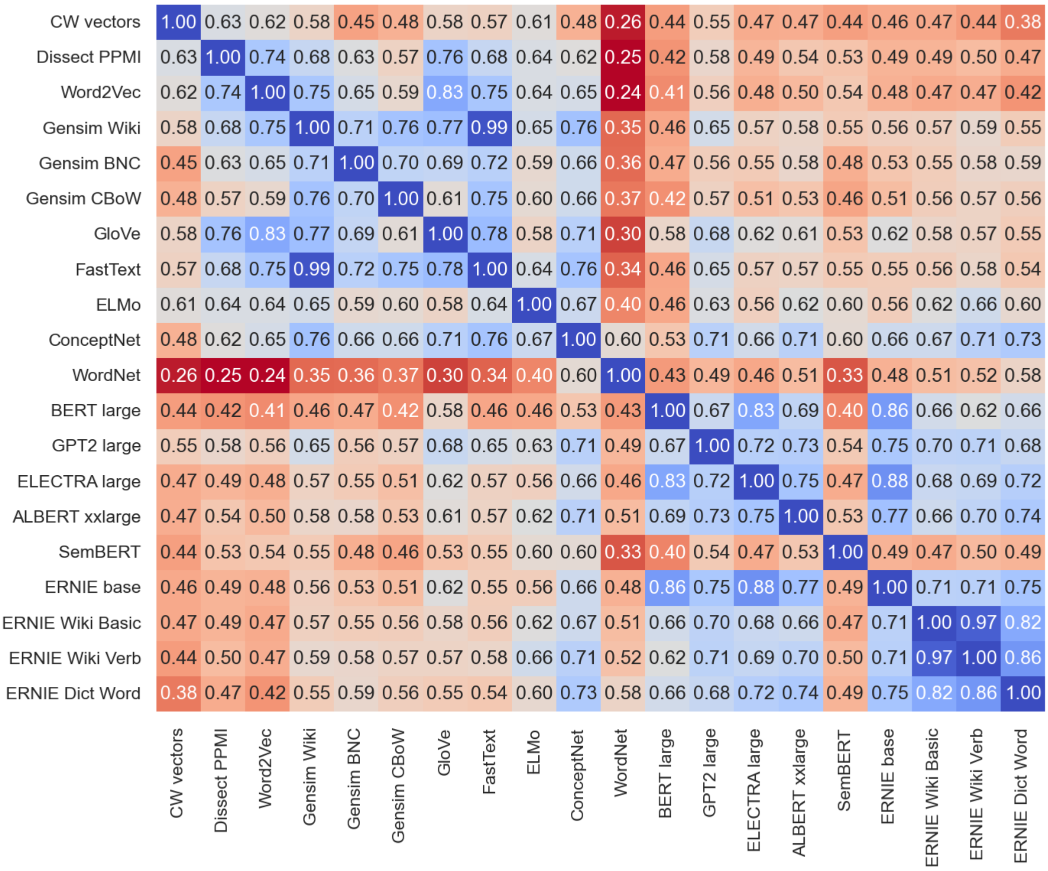
	RG65	MT287	MT771	WS198	RW	MEN	SimLex	MultiSim	SE2017	TR1058	Average
CW vectors	0.47	0.55	0.49	0.59	0.38	0.56	0.31	0.40	0.56	0.43	0.47
Dissect PPMI	0.73	0.63	0.63	0.65	0.40	0.71	0.40	0.50	0.67	0.58	0.57
Word2Vec	0.70	0.71	0.64	0.76	0.42	0.74	0.38	0.47	0.67	0.60	0.59
Gensim Wiki	0.71	0.64	0.62	0.77	0.49	0.73	0.40	0.50	0.72	0.64	0.61
Gensim BNC	0.75	0.67	0.67	0.75	0.41	0.76	0.40	0.52	0.72	0.59	0.60
Gensim CBoW	0.68	0.62	0.57	0.74	0.49	0.70	0.48	0.52	0.66	0.58	0.59
GloVe	0.77	0.70	0.71	0.80	0.46	0.80	0.43	0.55	0.72	0.61	0.64
FastText	0.71	0.65	0.63	0.78	0.49	0.74	0.40	0.51	0.72	0.65	0.61
ELMo	0.72	0.58	0.61	0.73	0.47	0.64	0.46	0.54	0.69	0.53	0.57
ConceptNet	0.92	0.75	0.82	0.85	0.63	0.87	0.62	0.70	0.84	0.72	0.76
WordNet	0.56	0.48	0.56	0.57	0.43	0.45	0.53	0.57	0.61	0.41	0.48
BERT large	0.77	0.55	0.67	0.75	0.35	0.67	0.51	0.59	0.69	0.49	0.56
GPT2 large	0.65	0.61	0.72	0.75	0.46	0.69	0.51	0.57	0.62	0.62	0.60
ELECTRA large	0.80	0.65	0.70	0.78	0.38	0.72	0.51	0.59	0.73	0.57	0.60
ALBERT xxlarge	0.76	0.68	0.71	0.78	0.41	0.73	0.53	0.60	0.73	0.55	0.61
SemBERT	0.76	0.61	0.69	0.78	0.40	0.70	0.51	0.60	0.72	0.50	0.59
ERNIE base	0.78	0.62	0.70	0.77	0.42	0.73	0.52	0.59	0.72	0.61	0.61
ERNIE Wiki Basic	0.68	0.70	0.66	0.75	0.41	0.74	0.52	0.57	0.64	0.53	0.60
ERNIE Wiki Noun	0.68	0.68	0.66	0.74	0.39	0.72	0.55	0.59	0.64	0.51	0.59

Figure 5: Spearman correlations between embedding models (rows) and noun-based experimental datasets (columns).



	CW vectors	Dissect PPMI	Word2Vec	Gensim Wiki	Gensim BNC	Gensim CBoW	GloVe	FastText	ELMo	ConceptNet	WordNet	BERT large	GPT2 large	ELECTRA large	ALBERT xxlarge	SemBERT	ERNIE base	ERNIE Wiki Basic	ERNIE Wiki Noun
CW vectors	1.00	0.75	0.74	0.72	0.67	0.66	0.71	0.72	0.75	0.67	0.40	0.64	0.69	0.67	0.69	0.67	0.66	0.72	0.72
Dissect PPMI	0.75	1.00	0.88	0.83	0.83	0.74	0.87	0.83	0.78	0.81	0.45	0.64	0.73	0.70	0.74	0.74	0.71	0.76	0.76
Word2Vec	0.74	0.88	1.00	0.87	0.84	0.79	0.91	0.87	0.79	0.86	0.43	0.67	0.75	0.74	0.76	0.75	0.74	0.77	0.75
Gensim Wiki	0.72	0.83	0.87	1.00	0.86	0.89	0.86	0.99	0.81	0.85	0.43	0.67	0.77	0.75	0.77	0.75	0.75	0.81	0.81
Gensim BNC	0.67	0.83	0.84	0.86	1.00	0.82	0.85	0.87	0.77	0.84	0.43	0.68	0.74	0.74	0.76	0.73	0.74	0.80	0.78
Gensim CBoW	0.66	0.74	0.79	0.89	0.82	1.00	0.80	0.89	0.78	0.81	0.43	0.63	0.73	0.71	0.74	0.71	0.72	0.79	0.74
GloVe	0.71	0.87	0.91	0.86	0.85	0.80	1.00	0.87	0.74	0.89	0.46	0.75	0.79	0.80	0.80	0.75	0.80	0.80	0.79
FastText	0.72	0.83	0.87	0.99	0.87	0.89	0.87	1.00	0.81	0.86	0.43	0.68	0.77	0.75	0.77	0.76	0.76	0.81	0.81
ELMo	0.75	0.78	0.79	0.81	0.77	0.78	0.74	0.81	1.00	0.77	0.45	0.66	0.74	0.72	0.78	0.76	0.72	0.83	0.80
ConceptNet	0.67	0.81	0.86	0.85	0.84	0.81	0.89	0.86	0.77	1.00	0.52	0.74	0.79	0.81	0.83	0.79	0.82	0.84	0.82
WordNet	0.40	0.45	0.43	0.43	0.43	0.43	0.46	0.43	0.45	0.52	1.00	0.47	0.48	0.48	0.49	0.49	0.48	0.53	0.50
BERT large	0.64	0.64	0.67	0.67	0.68	0.63	0.75	0.68	0.66	0.74	0.47	1.00	0.77	0.86	0.80	0.64	0.87	0.81	0.78
GPT2 large	0.69	0.73	0.75	0.77	0.74	0.73	0.79	0.77	0.74	0.79	0.48	0.77	1.00	0.81	0.83	0.71	0.83	0.85	0.81
ELECTRA large	0.67	0.70	0.74	0.75	0.74	0.71	0.80	0.75	0.72	0.81	0.48	0.86	0.81	1.00	0.84	0.71	0.89	0.84	0.82
ALBERT xxlarge	0.69	0.74	0.76	0.77	0.76	0.74	0.80	0.77	0.78	0.83	0.49	0.80	0.83	0.84	1.00	0.76	0.85	0.86	0.83
SemBERT	0.67	0.74	0.75	0.75	0.73	0.71	0.75	0.76	0.76	0.79	0.49	0.64	0.71	0.71	0.76	1.00	0.71	0.80	0.77
ERNIE base	0.66	0.71	0.74	0.75	0.74	0.72	0.80	0.76	0.72	0.82	0.48	0.87	0.83	0.89	0.85	0.71	1.00	0.86	0.85
ERNIE Wiki Basic	0.72	0.76	0.77	0.81	0.80	0.79	0.80	0.81	0.83	0.84	0.53	0.81	0.85	0.84	0.86	0.80	0.86	1.00	0.96
ERNIE Wiki Noun	0.72	0.76	0.75	0.81	0.78	0.74	0.79	0.81	0.80	0.82	0.50	0.78	0.81	0.82	0.83	0.77	0.85	0.96	1.00

Figure 6: Correlation matrices of all models computed over the vocabulary of the MEN noun dataset.



	CW vectors	Dissect PPMI	Word2Vec	Gensim Wiki	Gensim BNC	Gensim CBoW	GloVe	FastText	ELMo	ConceptNet	WordNet	BERT large	GPT2 large	ELECTRA large	ALBERT xxlarge	SemBERT	ERNIE base	ERNIE Wiki Basic	ERNIE Wiki Verb	ERNIE Dict Word
CW vectors	1.00	0.63	0.62	0.58	0.45	0.48	0.58	0.57	0.61	0.48	0.26	0.44	0.55	0.47	0.47	0.44	0.46	0.47	0.44	0.38
Dissect PPMI	0.63	1.00	0.74	0.68	0.63	0.57	0.76	0.68	0.64	0.62	0.25	0.42	0.58	0.49	0.54	0.53	0.49	0.49	0.50	0.47
Word2Vec	0.62	0.74	1.00	0.75	0.65	0.59	0.83	0.75	0.64	0.65	0.24	0.41	0.56	0.48	0.50	0.54	0.48	0.47	0.47	0.42
Gensim Wiki	0.58	0.68	0.75	1.00	0.71	0.76	0.77	0.99	0.65	0.76	0.35	0.46	0.65	0.57	0.58	0.55	0.56	0.57	0.59	0.55
Gensim BNC	0.45	0.63	0.65	0.71	1.00	0.70	0.69	0.72	0.59	0.66	0.36	0.47	0.56	0.55	0.58	0.48	0.53	0.55	0.58	0.59
Gensim CBoW	0.48	0.57	0.59	0.76	0.70	1.00	0.61	0.75	0.60	0.66	0.37	0.42	0.57	0.51	0.53	0.46	0.51	0.56	0.57	0.56
GloVe	0.58	0.76	0.83	0.77	0.69	0.61	1.00	0.78	0.58	0.71	0.30	0.58	0.68	0.62	0.61	0.53	0.62	0.58	0.57	0.55
FastText	0.57	0.68	0.75	0.99	0.72	0.75	0.78	1.00	0.64	0.76	0.34	0.46	0.65	0.57	0.57	0.55	0.55	0.56	0.58	0.54
ELMo	0.61	0.64	0.64	0.65	0.59	0.60	0.58	0.64	1.00	0.67	0.40	0.46	0.63	0.56	0.62	0.60	0.56	0.62	0.66	0.60
ConceptNet	0.48	0.62	0.65	0.76	0.66	0.66	0.71	0.76	0.67	1.00	0.60	0.53	0.71	0.66	0.71	0.60	0.66	0.67	0.71	0.73
WordNet	0.26	0.25	0.24	0.35	0.36	0.37	0.30	0.34	0.40	0.60	1.00	0.43	0.49	0.46	0.51	0.33	0.48	0.51	0.52	0.58
BERT large	0.44	0.42	0.41	0.46	0.47	0.42	0.58	0.46	0.46	0.53	0.43	1.00	0.67	0.83	0.69	0.40	0.86	0.66	0.62	0.66
GPT2 large	0.55	0.58	0.56	0.65	0.56	0.57	0.68	0.65	0.63	0.71	0.49	0.67	1.00	0.72	0.73	0.54	0.75	0.70	0.71	0.68
ELECTRA large	0.47	0.49	0.48	0.57	0.55	0.51	0.62	0.57	0.56	0.66	0.46	0.83	0.72	1.00	0.75	0.47	0.88	0.68	0.69	0.72
ALBERT xxlarge	0.47	0.54	0.50	0.58	0.58	0.53	0.61	0.57	0.62	0.71	0.51	0.69	0.73	0.75	1.00	0.53	0.77	0.66	0.70	0.74
SemBERT	0.44	0.53	0.54	0.55	0.48	0.46	0.53	0.55	0.60	0.60	0.33	0.40	0.54	0.47	0.53	1.00	0.49	0.47	0.50	0.49
ERNIE base	0.46	0.49	0.48	0.56	0.53	0.51	0.62	0.55	0.56	0.66	0.48	0.86	0.75	0.88	0.77	0.49	1.00	0.71	0.71	0.75
ERNIE Wiki Basic	0.47	0.49	0.47	0.57	0.55	0.56	0.58	0.56	0.62	0.67	0.51	0.66	0.70	0.68	0.66	0.47	0.71	1.00	0.97	0.82
ERNIE Wiki Verb	0.44	0.50	0.47	0.59	0.58	0.57	0.57	0.58	0.66	0.71	0.52	0.62	0.71	0.69	0.70	0.50	0.71	0.97	1.00	0.86
ERNIE Dict Word	0.38	0.47	0.42	0.55	0.59	0.56	0.55	0.54	0.60	0.73	0.58	0.66	0.68	0.72	0.74	0.49	0.75	0.82	0.86	1.00

Figure 7: Correlation matrices of all models computed over the vocabulary of the SimVerb verb dataset.

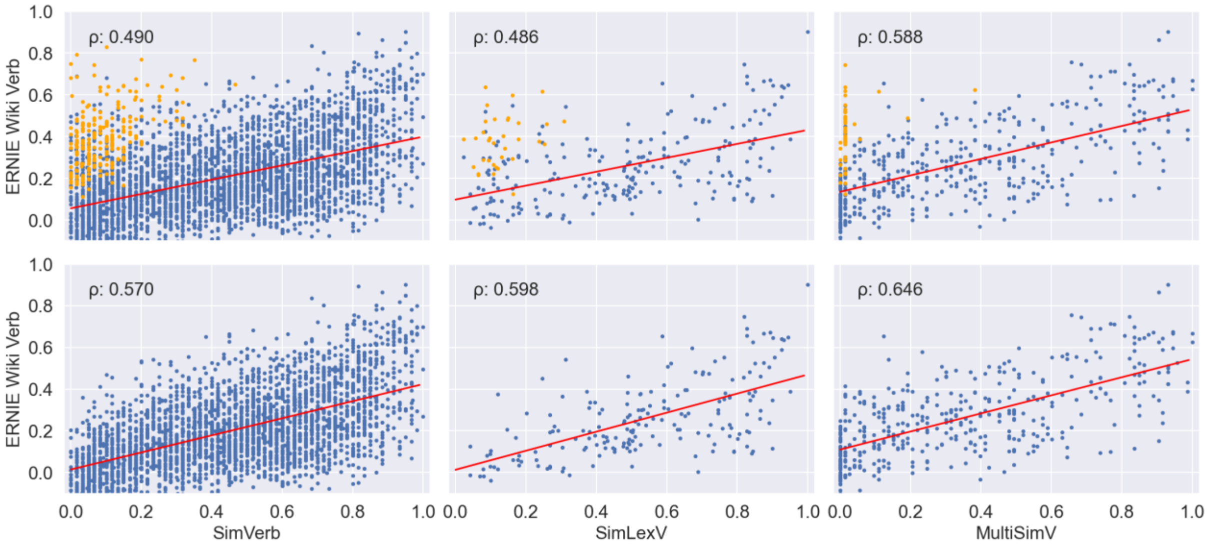
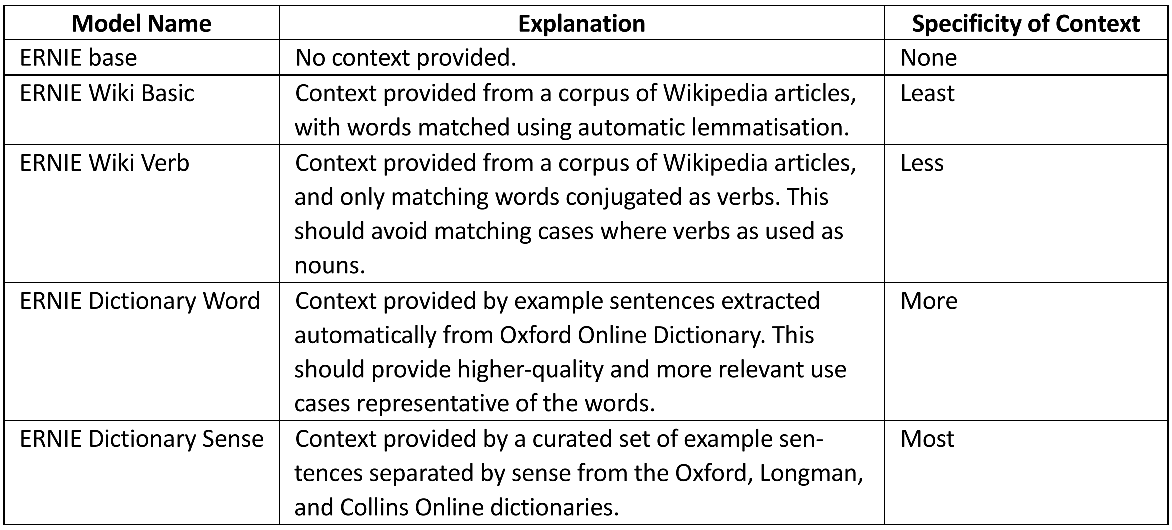


Figure 8: Effects of removing antonyms (shown in orange) from the SimVerb (left), SimLexVerb (centre), and MultiSimVerb (right) datasets.

Model Name	Type	Explanation	Citation
CW vectors	Count	Regression model trained over Wikipedia corpus.	Collobert et al. (2011)
Dissect PPMI	Count	Trained using Positive Point-wise mutual information (PPMI) over ukWaC, Wikipedia, and the British National Corpus.	Baroni et al. (2014)
Word2Vec skipgram	Predict	Skipgram model trained over Wikipedia.	Kutuzov et al. (2017)
Gensim Wiki	Predict	Skipgram model trained over Wikipedia and Gigaword corpus.	Kutuzov et al. (2017)
Gensim BNC	Predict	Skipgram model trained over British National Corpus.	Kutuzov et al. (2017)
Genism CBoW	Predict	Continuous Bag of Words (CBoW) model trained over Gigaword corpus.	Kutuzov et al. (2017)
GloVe	Predict	Custom regression model trained over 840 billion token corpus from the Common Crawl.	Pennington et al. (2014)
FastText	Predict	Skipgram model trained over Wikipedia and Gigaword corpus.	Kutuzov et al. (2017)
ELMo	Predict	A 94 million parameter bidirectional Long Short Term Memory (LSTM) trained over a 30 million word corpus.	Peters et al. (2018)
ConceptNet	Knowledge	ConceptNet relations are encoded into vectors by applying PPMI to the relation adjacency matrix, plus extra information from GloVe and word2vec.	Speer et al. (2017)
WordNet	Knowledge	WordNet relations encoded into vectors by counting number of intermediate nodes.	Saedi et al. (2018)
BERT large	Transformer	A 340 million parameter transformer model trained on a 3.3 billion token corpus from Wikipedia and BooksCorpus.	Devlin et al. (2019)
GPT2 large	Transformer	A 1.5 billion parameter transformer model trained on a web corpus of 8 million documents.	Radford et al. (2019)
ELECTRA large	Transformer	A 335 million parameter transformer model trained on a 33 billion token web corpus.	Clark et al. (2020)
ALBERT xxlarge	Transformer	A 233 million parameter transformer model trained based on BERT.	Lan et al. (2020)
SemBERT	Transformer	A 240 million parameter transformer model based on BERT and incorporating semantic role labelling.	Zhang et al. (2020)
ERNIE	Transformer	A 10 billion parameter transformer model trained on a corpus of plain text and knowledge graphs.	Sun et al. (2021)

Table 3: Summary of word embedding models used in this paper.



Model Name	Explanation	Specificity of Context
ERNIE base	No context provided.	None
ERNIE Wiki Basic	Context provided from a corpus of Wikipedia articles, with words matched using automatic lemmatisation.	Least
ERNIE Wiki Verb	Context provided from a corpus of Wikipedia articles, and only matching words conjugated as verbs. This should avoid matching cases where verbs as used as nouns.	Less
ERNIE Dictionary Word	Context provided by example sentences extracted automatically from Oxford Online Dictionary. This should provide higher-quality and more relevant use cases representative of the words.	More
ERNIE Dictionary Sense	Context provided by a curated set of example sentences separated by sense from the Oxford, Longman, and Collins Online dictionaries.	Most

Table 4: Summary of ERNIE embeddings constructed in the paper, and with an indication of how fine-grained is the context incorporated into the embeddings.

RaTE: a Reproducible automatic Taxonomy Evaluation by Filling the Gap

Tianjian Gao, Philippe Langlais
RALI/IDIRO, Université de Montréal
tianjian.gao@umontreal.ca,felipe@iro.umontreal.ca

Abstract

Taxonomies are an essential knowledge representation, yet most studies on automatic taxonomy construction (ATC) resort to manual evaluation to score proposed algorithms. We argue that automatic taxonomy evaluation (ATE) is just as important as taxonomy construction. We propose RaTE[1], an automatic label-free taxonomy scoring procedure, which relies on a large pre-trained language model. We apply our evaluation procedure to three state-of-the-art ATC algorithms with which we built seven taxonomies from the Yelp domain, and show that 1) RaTE correlates well with human judgments and 2) artificially degrading a taxonomy leads to decreasing RaTE score.

1 Introduction

A domain taxonomy is a tree-like structure that not only aids in knowledge organization but also serves an integral part of many knowledge-rich applications including web search, recommendation systems and decision making processes. Taxonomies are also inevitably used as business and product catalogs and for managing online sales. Notable taxonomy products in this domain include Amazon Category Taxonomy,[2] Google Product Taxonomy,[3] Yelp Business Category[4] and Google Content Categories.[5]

Recent years have witnessed interest in new automatic taxonomy construction (ATC) systems, but there are no systematic methods for objectively evaluating their figure of merit. For instance, TaxoGen (Zhang et al., 2018) — see Section 3 — was evaluated by asking at least three human evaluators if a taxonomy concept pair contains a hypernymy relationship, which can lead to bias and low reproducibility. It is not only difficult to compare or rank different algorithms, but changing the hyperparameters or settings of a parameterized ATC system can also result in drastically different outputs, and make optimization unfeasible.

Because ontologies and taxonomies in particular are typically created in contexts to address specific problems or achieve specific goals, e.g. classification, their evaluation is evidently context-dependent, and many researchers actually believe that a task-independent automatic evaluation remains elusive (Porzel and Malaka, 2004). Still, researchers have argued that objective evaluation metrics must be well available for significant progress in the development and deployment of taxonomies and ontologies (Brewster et al., 2004).

In this work, we propose RaTE, a Reproducible procedure for Automatic Taxonomy Evaluation. RaTE does not require external knowledge but instead depends on masked language modelling (MLM) to query a large language model for subsumption relations. We show that with some care, MLM is a valuable proxy to human judgments.

We apply RaTE to the Yelp corpus (a corpus of restaurant reviews) ranking seven taxonomies we extracted using three state-of-the-art ATC systems. We observe it correlates well with our manual evaluation of those taxonomies, and also show that artificially degrading a taxonomy leads to a decrease of score proportional to the level of noise injected.

In the remainder, we discuss related work in Section 2. In Section 3, we describe the ATC systems we used for building up our taxonomies, and their evaluation procedures. We then present RaTE in Section 4 including refinements that we found

[1] Our code repository is available at https://github.com/CestLucas/RaTE

[2] https://www.data4amazon.com/amazon-product-taxonomy-development-mapping-services.html

[3] https://support.google.com/merchants/answer/6324436?hl=en

[4] https://blog.yelp.com/businesses/yelp_category_list/

[5] https://cloud.google.com/natural-language/docs/categories?hl=fr

Proceedings of the 15th International Conference on Computational Semantics pages 173–182
June 21–23, 2023. ©2023 Association for Computational Linguistics

necessary for our approach to work. We report in Section 5 the experiments we conducted to demonstrate the relevance of RaTE, and conclude in Section 6.

2 Related Works

Systematic methods of evaluating ontologies and taxonomies are lacking. Because agreed upon quantitative metrics are lacking, research on taxonomy and ontology construction relies heavily on qualitative descriptions and the various perspectives of ontology engineers, system users or domain experts, which renders the results subjective and unreproducible (Gómez-Pérez, 1999; Guarino, 1998).

Brank et al. (2005) summarized four principle ontology evaluation methods, by (1) comparing the target ontology to a "gold standard" (ground-truth) ontology (Maedche and Staab, 2002); (2) using the target ontology in an application and evaluating the application results ("application based") (Porzel and Malaka, 2004); (3) conducting coverage analysis comparing the target with a source of data (eg., a collection of documents) about a specific domain ("data driven") (Brewster et al., 2004); (4) manual reviews done by human experts that assess how well the target ontology meets a set of predefined criteria, standards, and requirements (Lozano-Tello and Gómez-Pérez, 2004).

Gold Standard Evaluation focusses on comparing and measuring the similarity of the target taxonomy with an existing ground truth such as WordNet (Fellbaum, 1998), Wikidata and ResearchCyc (Ponzetto and Strube, 2011). Semantic similarity metrics have been proposed, including Wu-Palmer (Wu and Palmer, 1994), Leacock-Chodorow (Leacock and Chodorow, 1998) and Lin (Lin, 1998). We include in this category specific measures such as *topic coherence* (Newman et al., 2010) which scores the quality of a word cluster which rely on similarity measures. There are several issues with such a process: mapping concepts from the output system to the ground truth is not trivial and gold standards do not necessarily cover well the domains of interest.

Application-based Evaluation is an attractive alternative to gold-standard evaluation. Porzel and Malaka (2004) for instance proposed several possible applications for evaluation including concept-pair relation classification. Brank et al. (2005) underlines however that it is in fact hard to correlate ontology quality with the application performance.

Data-driven Evaluation intends to select the ontology O with the best structural *fit* to a target corpus C, which boils down into estimating $P(C|O)$ as in (Brewster et al., 2004). Practically however, it remains unclear how to approximate such conditional probability.

3 Automatic Taxonomy Extractors

In this work, we replicated results of three state-of-the-art ATC systems that are publicly available and that are producing quality results on selected datasets and domains. In this section, we describe those systems and discuss their corresponding evaluation methods.

3.1 TaxoGen

TaxoGen (Zhang et al., 2018) is an adaptive text embedding and clustering algorithm leveraging various phrase-mining and clustering techniques including AutoPhrase (Shang et al., 2018), caseOLAP (Liem et al., 2018) and spherical k-means clustering (Banerjee et al., 2005). TaxoGen iteratively refines selected keywords and chooses cluster representative terms based on two criteria: *popularity* which prefers term-frequency in a cluster and *concentration* which assumes that representative terms should be more relevant to their belonging clusters than their sibling clusters.

The system can be configured with several hyper-parameters including the depth of the taxonomy, the number of children per parent term and the "representativeness" threshold. Experiments were conducted on DBLP and SP (Signal Processing) datasets and the system is quantitatively evaluated with relation accuracy and term coherency measures assessed by human evaluators (10 doctoral students).

3.2 CoRel

CoRel (Huang et al., 2020) takes advantages of novel relation transferring and concept learning techniques and uses hypernym-hyponym pairs provided in a seeded taxonomy to train a BERT (Devlin et al., 2019) relation classifier and expand the seeded taxonomy horizontally (width expansion) and vertically (depth expansion). Topical clusters are generated using pre-computed BERT embeddings and a discriminative embedding space is learned, so that each concept is surrounded by its representative terms.

The clustering algorithms used by CoRel are *spectral co-clustering* (Kluger et al., 2003) and *affinity propagation* (Frey and Dueck, 2007), which automatically computes the optimal number of topic clusters. Compared to TaxoGen, CoRel does not require depth and cluster number specifications but a small seeding taxonomy as an input for enabling a weakly-supervised relation classifier.

CoRel is quantitatively evaluated with term coherency, relation F1 and sibling distinctiveness judged by 5 computer science students on subsets of DBLP and Yelp datasets. The system generates outputs in the form of large hierarchical topic word clusters.

3.3 HiExpan

HiExpan (Shen et al., 2018) is a hierarchical tree expansion framework that aims to dynamically expand a seeded taxonomy horizontally (width expansion) and vertically (depth expansion) and performs entity linking with Microsoft's Probase (Wu et al., 2012) — a probabilistic framework used to harness 2.7 million concepts mined from 1.68 billion web pages — to iteratively grow a seeded taxonomy. As an entity is matched against a verified knowledge base, we perceive the accuracy of terms and concept relations to be higher than that of CoRel and TaxoGen.

Authors of the HiExpan, as well as some volunteers assessed the taxonomy parent-child pair relations using ancestor- and edge-F1 scores.

3.4 Observations

Each of those taxonomy extractors face their own set of advantages and drawbacks. TaxoGen is the only parameterized systems in our experiments, and is the only one that does not require a seeded input for producing an output, which can be beneficial when prior knowledge of the corpus is lacking. It also generates alternative synonyms for each taxonomy topic, which increases the coverage and improves concept mapping between taxonomies and documents. However, it seems to depend on the keyword extraction quality and it is unclear how to determine the best hyper-parameter settings owing to the lack of automatic evaluation methods.

CoRel uses the concept pairs provided in the seed taxonomy for mining similar relations, but this has become its Achilles' heel because same-sentence co-occurrence of valid parent-child topics is rare in real-world data. As a result, CoRel may fail to produce any output at all due to insufficient training examples for the relation classifier. It is also resource-intensive for making use of neural networks for relation transferring and depth expansion. Anecdotally, the output of CoRel may also not be entirely exhaustive and deterministic.

For our experiments, HiExpan is perceived to produce the most consistent taxonomies thanks to the use of Probase for measuring topic similarities and locating related concepts. However, the set-expansion mechanism of HiExpan often ignores topic granularity and adds hyponyms and hypernyms found in similar contexts to the exact same taxonomy level (hence most HiExpan taxonomies are two-level only). It also cannot differentiate word senses such as virus as in *computer virus* and a *viral disease*.

4 RaTE

A critical part of taxonomy/ontology evaluation is knowledge about subsumptions, e.g. "is *fluorescence spectroscopy* a type of *fluorescence technology*?" or "is *CRJ200* a *Bombardier*?".

Thus, RaTE measures the accuracy of the hypernym relations present in a taxonomy we seek to evaluate. The main difference between our work and earlier ones is that we do not rely on human judgments to determine the quality of a parent-child pair, nor do we consider an external reference (that often is not available or simply too shallow). Instead, we rely on a large language model tasked to check subsumption relations.

Ultimately, an optimized language model should be able to generate an accurate list of the most canonical hypernyms for a given domain, similar to domain experts. But because we are mainly interested in domain-specific taxonomies, there is a high risk that specific terms of the domain are not well recognized by the model, and therefore, we investigate three methods for increasing the hit rate of hypernymy prediction of taxonomy subjects and reducing false negatives by (1) creating various prompts, (2) fine-tuning MLMs with different masking procedures, and (3) extending the model's vocabulary with concept names.

4.1 Core idea

We consider a taxonomy as a set of n parent-child pairs from adjacent taxonomy levels linked by single edges, denoted as $(p, c) \in \mathcal{T}$. For each parent-child pair $(p_i, c_i), i \in 1, ..., n$, we insert c_i and the "[MASK]" token into some prompts containing

c	Pred 1	Pred 2	Pred 3	Pred 4	Pred 5	Rank
Mussel	fish (0.227)	dish (0.144)	seafood (0.140)	meat (0.037)	soup (0.033)	3
Clam	fish (0.203)	dish (0.095)	seafood (0.076)	crab (0.030)	thing (0.027)	3
Lobster	seafood (0.222)	dish (0.145)	lobster (0.131)	food (0.052)	sauce (0.052)	1
Chicken	dish (0.167)	meat (0.110)	chicken (0.079)	thing (0.058)	sauce (0.052)	73
Beef	meat (0.274)	beef (0.161)	dish (0.063)	food (0.027)	thing (0.024)	57

Table 1: Top-5 hypernym predictions made by a pre-trained BERT model (Bert-large-uncased-whole-word-masking) by prompting it with "*c is a type of [MASK]*". The rank of seafood in the list is indicated in the last column.

"is-a" patterns (Hearst, 1992), then use LMs to unmask $p'_1(c_i), p'_2(c_i), ..., p'_k(c_i) \in p'(c_i)$ per query as proxy parent terms of c_i, where k is a recall threshold (we used $k = 10$ in this work). This process is illustrated in Table 1.

A good pair of taxonomy concepts is therefore if the parent concept p_i can be found among the machine predictions $p'(c_i)$. We consider a parent-child relation *positive* if and only if the parent term is recalled one or more[6] times in the top k predictions. This policy can obviously be adjusted, which we leave as future work. The measure of quality of $\mathcal{T}$ is then simply the percentage of (p, c) links in $\mathcal{T}$ that are correct according this procedure. We note that for a taxonomy with no parent-child pairs, i.e. a single-level taxonomy, our evaluation score is 0.

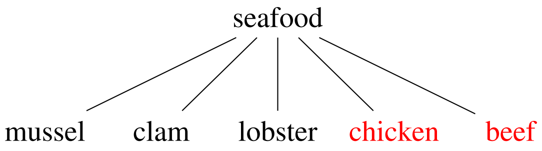


Figure 1: Excerpt from HiExpan1 for topic "seafood"

As an illustration, the taxonomy in Figure 1 would receive a score of 3/5 based on the predictions made in Table 1 where for instance, $p'_1(c_i), p'_2(c_i), ..., p'_5(c_i)$ equal *fish, dish, seafood, meat, soup* for $c_i =$ *mussel*, in which we find the real taxonomy parent $p_i = \textit{seafood} = p'_3(c_i)$.

We observe from Table 1 that not every prediction is factually correct (e.g. mussels are neither fish nor meat), and it remains evidently unreliable to depend solely upon pre-trained language models as ground-truth for all knowledge domains. Yet, we argue that we can regard the rankings of MLM predictions as a likelihood of a subsumption relation between the subject and the object of a query. In our example, the model is significantly more likely to predict "seafood" for *mussel, clam* and *lobster* (rank 3,3,1) than for *chicken* and *beef* (rank 73,57).

4.2 Diversified Prompting

Models can produce all sorts of trivial predictions, such as stop-words (e.g. "**this** is a kind of seafood"), or expressions and collocations found frequently in training samples (e.g. "seafood is a kind of **joke/disappointment**").

Differences in prompts used can actively impact a model's performance in hypernymy retrieval (Peng et al., 2022; Hanna and Mareček, 2021). Hanna and Mareček (2021) reported that prompting BERT for hypernyms can actually outperform other unsupervised methods even in an unconstrained scenario, but the effectiveness of it depends on the actual queries. For example, they show that the query "A(n) x **is a** [MASK]" outperformed "A(n) x **is a type of** [MASK]" on the Battig dataset.

As a result, instead of relying on a single query, we design five pattern groups (p1-p5) of hypernymy tests for pooling unmasking results. Those are illustrated in Table 2 for the parent-child pair (seafood,shrimp).

While p2 to p4 follow standard Hearst-like patterns (Hearst, 1992), p5a employs the "my favourite is" prompt which has demonstrated high P@1 and MRR in (Hanna and Mareček, 2021). Patterns p1 have been created specifically for noun phrases that have a tendency to be split and considered as good taxonomy edges by ATC systems.[7]

With this refined set of patterns, a topic pair has therefore a score of 1, as in the seafood-shrimp example, if the parent term is among the top-k machine predictions for any inquiries containing the child topic, and 0 vice versa. Again, more elaborate decisions can be implemented.

[6] A parent word can be predicted multiple times in singular and plural forms, misspellings, and so on, e.g. "dessert", "desserts" and "desert".

[7] For instance, extractors tend to produce (salad,shrimp) for the pair (salad,shrimp salad).

Prompt		Pred1	Pred2	Pred3	Pred4	Pred5	Rank
p1a	{shrimp} [MASK]	salad	cocktail	pasta	soup	rice	359
p1b	[MASK] {shrimp}	fried	no	garlic	coconut	fresh	117
p2a	{shrimp} is a [MASK]	joke	must	winner	favorite	hit	959
p2b	{shrimp} is an [MASK]	option	issue	experience	art	order	4407
p3a	{shrimp} is a kind of [MASK]	joke	thing	dish	treat	disappointment	146
p3b	{shrimp} is a type of [MASK]	dish	thing	food	sauce	seafood	5
p3c	{shrimp} is an example of [MASK]	that	this	shrimp	food	seafood	5
p4a	[MASK] such as {shrimp}	sides	food	seafood	fish	shrimp	3
p4b	A [MASK] such as {shrimp}	lot	variety	side	combination	protein	40
p4c	An [MASK] such as {shrimp}	ingredient	item	option	order	animal	197
p5a	My favorite [MASK] is {shrimp}	dish	thing	part	item	roll	16

Table 2: Evaluation queries for the parent-child pair (seafood,shrimp).

4.3 Fine-tuning the Language Model

To improve hypernymy predictions, we must also address two issues with pre-trained language models: (1) the models are untrained on the evaluation domain; (2) the default model tokenizer and vocabulary are oblivious of some taxonomy topics, resulting in lower recall.

Most research on MLM prompting only assessed the performance of pre-trained models. Yet, Peng et al. (2022) found an improvement when using FinBert models (Yang et al., 2020) pre-trained with massive financial corpora in retrieving financial hypernyms such as *equity* and *credit* for *"S&P 100 index is a/an __ index"*, compared to using BERT-base. Also, Dai et al. (2021) generated ultra-fine entity typing labels, e.g. "person, soldier, man, criminal" for *"**he** was confined at Dunkirk, escaped, set sail for India"* through inserting hypernym extraction patterns and training LMs to predict such patterns.

Analogously, we compared six fine-tuned models, investigating different masking protocols, model vocabulary (see next section) and training sizes. Because we want the language models to concentrate on the taxonomy entities, particularly the parent terms and their surrounding contexts, we prioritize therefore masking the main topics (shown in Table 3) and parent terms of the taxonomies to evaluate, then other taxonomy entities (e.g. leaf nodes), followed by AutoPhrase entities if no taxonomy entities are present in the sentence and other random tokens from our training samples. In addition, we test entity masking by only masking *one* taxonomy entity rather than 15% of sentence tokens to gain more sentence contexts. Our masking procedures are illustrated in Figure 2.

4.4 Extended Vocabulary

Domain-specific words such as food items are typically not predicted as a whole word, but rather as a sequence of subword units, such as *appetizer* which is treated as *'app', '##eti' and '##zer'* by the standard tokenizer. To avoid multi-unit words to be overlooked by the language model, we propose to extend its vocabulary.

Review	Everything was pretty good but the beef in the mongolian beef was very chewy and had a weird texture.	
Entities	Taxonomy	beef (CoRel1-4, HiExpan1) mongolian (CoRel1-4)
	AutoPhrase	beef, chewy, mongolian weird texture

Masking Policy

Entity	15%	Everything was pretty good but the [MASK] in the [MASK] [MASK] was very chewy and had a weird texture.
	one	Everything was pretty good but the [MASK] in the mongolian [MASK] was very chewy and had a weird texture.
Token	15%	Everything was pretty [MASK] but the beef in the mongolian beef [MASK] very chewy and had a [MASK] texture.

Figure 2: Comparison of masking strategies for a sample Yelp review where taxonomy entities or those proposed by AutoPhrase are underlined. We prioritize masking the taxonomy entities, AutoPhrase entities and random tokens, in that order.

We enrich the vocabulary of models m1 and m2, by adding the lemmas (or singular forms) of parent terms from Table 3 that were not previously included in the base tokenizer, such as “sushi", “appetizer" and “carne asada", and resizing the models’ token embedding matrices to match the size of the new tokenizer. The embedding representation of new tokens were initialized randomly before fine-tuning, although it is possible to assign them to the representation of the closest terms in the original vocabulary.

By adding only a small number of new tokens to the model and tokenizer, we also ensure similar model and tokenizer efficiencies. We believe that vocabulary extension will become a necessary step for effective hypernymy prediction in most specialized domains, though the exact optimal strategies remain to be discussed.

5 Experiments

We conducted our experiments on the Yelp corpus which contains around 1.08M restaurant reviews such as the one in Figure 2 (top box). We used the very same corpus prepared by Huang et al. (2020).[8]

5.1 Taxonomies

We created seven taxonomies for evaluation using the ATC systems mentioned in Section 3. Here our goal was to obtain meaningful taxonomies that best cover the Yelp domain using each taxonomy extractor. We did so by experimenting with different extractor settings and input. For TaxoGen, we only had to specify some parameters.[9] For CoRel and HiExpan however, we had to provide a seed taxonomy. Hence we produced 5 such taxonomies using CoRel and HiExpan by providing frequently-appearing parent-child pairs in the seeds.[10]

Table 3 reports the main topics (level 1) of the produced taxonomies. We observe that the output of one ATC system varies substantially from one parametrization to another. Also, it is noticeable that the main topic of some taxonomies do lack structure. For instance, putting *beef* and *chicken* in the category *meat* would arguably make better sense in CoRel1.

5.2 Models

We fined-tuned six language models according to the different strategies we presented in Section 4 and which characteristics are summarized in Table 4. In particular, we experiment with *entity masking* while fine-tuning model m1a, m1b and m0b, which emphasizes masking task-relevant tokens, because it has been shown to be more effective than *random masking* in (Sun et al., 2019; Kawintiranon and Singh, 2021). All models have been fine-tuned for 2 epochs by masking 15% tokens, to the exception of m1b (marked with ⋆) for which only one entity has been masked per example.

Taxonomy	Top level (main) topics
CoRel1	steak, veggies, beef, cheese, crispy, fish, rice, salad, shrimp, spicy, pork, bacon, burger, appetizer, bread, dessert, seafood
CoRel2	bacon, bread, fries, roll, soup, burger, dessert, salad, shrimp
CoRel3	chinese, seafood, dessert, steak
CoRel4	dinner, food, location, lunch, service
HiExpan1	seafood, salad, dessert, appetizer, food, sushi, desert, pizza, coffee, bread, pasta, beer, soup, wine, cheese, cocktail, taco, water, music
TaxoGen1	main_dish, south_hills, high_ceilings, était_pas
TaxoGen2	chest, tempe, amaretto, pepper_jelly, relies, travis, free_admission, exposed_brick

Table 3: Main targets of MLM evaluation.

Model	Finetuning			Masking	
name (base)	Voc.	Full	70%	Ent.	Tok
m1a (bert-base)	✓	✓		✓	
m1b (bert-base)	✓	✓		✓⋆	
m2a (bert-base)	✓	✓			✓
m2b (bert-base)	✓		✓		✓
m0a (bert-base)			✓		✓
m0b (distilbert-base)			✓	✓	

Table 4: Configurations of the fine-tuned models, with models m0a and m0b serving as baselines for training with the base tokenizer; m0b using a smaller pre-trained model and less fine-tuning material. Column Voc. indicates that main target words proposed ATC systems were injected in the model’s vocabulary.

For comparison purposes, we also selected two pre-trained models bert-large-uncased-whole-word-masking and bert-base-uncased that we did not fine-

[8] Available at: https://drive.google.com/drive/folders/13DQ0II9QFLDhDbbRcbQ-Ty9hcJETbHt9.

[9] We considered taxonomy depth, number of topics per level, and “word filtering threshold”. See the github for the specific values we used.

[10] They pretty much align with the one used by Huang et al. (2020), although we proceeded by trial-error until satisfaction.

Model	Pred1	Pred2	Pred3	Pred4	Rank
m1a	burger	dish	sandwich	steak	4
m1b	dish	burger	beer	sandwich	10
m2a	steak	dish	meat	cut	1
m2b	steak	dish	burger	meat	1
m0a	dish	burger	steak	meat	3
m0b	cut	steak	meat	beef	2
B-l	fruit	flavor	food	color	69
B-b	food	drink	color	dessert	71

Table 5: Fine-tuned (top) vs. pre-trained (bottom) models' top-4 predictions with the prompt "my favourite [MASK] is sirloin ."

tune and that we named B-l and B-b respectively.

To highlight the qualitative differences between our evaluation models, we provide a simple prompt "my favourite [MASK] is sirloin" for the models to predict the taxonomy hypernym "steak" in CoRel1. The results are shown in Table 5, where 5 out of 6 fine-tuned models and none of the pre-trained models correctly predicted the taxonomy parent in the top 4 predictions. Further, all fine-tuned models returned "steak" in the top ten predictions.

Lastly, we show the positive effects of extending the vocabulary of the language model in Table 6 where we wish to recall the parent term "appetizer" for the concept pair "appetizer-mozzarella sticks" in CoRel1, where the token "appetizer" would be split into *'app', '##eti' and '##zer'* by the standard tokenizer. Both models m1a and m1b trained with entity masking and an expanded vocabulary correctly predicted "appetizer" in their top five predictions; m2 models also recalled the term, albeit with a very low rank whereas other models are completely oblivious to it. Nevertheless, we find that expanding the model's vocabulary in conjunction with entity masking may introduce bias into the models when fine-tuning with limited training samples, i.e. over predicting the added tokens.

5.3 Ranking Results

5.3.1 Manual Ranking

The first author of this paper first manually ranked the extracted taxonomies prior to experimenting with RaTE. The main task was to manually verify the validity of the parent-child pairs of each taxonomy, while also taking into account factors like taxonomy structure.[11]

Model	Pred1	Pred2	Pred3	Pred4	Rank
m1a	sides	foods	food	apps	5
m1b	sides	food	appetizer	foods	3
m2a	sides	items	food	dessert	6089
m2b	things	items	foods	props	3111
m0a	sides	extras	items	dessert	N/A
m0e	sides	apps	foods	snacks	N/A
B-l	foods	items	products	food	N/A
B-b	foods	snacks	food	items	N/A

Table 6: Top-4 predictions of models with extended (top) or base (bottom) vocabulary for the prompt "[MASK] such as mozzarella sticks".

HiExpan1 was deemed the best taxonomy, likely because the word relations actually originate from a verified database and the coverage is extensive. It is also observably more accurate than CoRel 1-4, which have similar (overall good) quality. TaxoGen taxonomies were the least accurate, with TaxoGen1 superior to TaxoGen2. We found them trivial in the sense that the algorithm selects many insignificant topics because no seeded taxonomy indicating user interest is provided. We believe that another cause for this is the system's low sensitivity to keywords supplied by AutoPhrase, which on Yelp generates too many irrelevant terms and leads to many noisy concept pairs (e.g. "exposed brick – music video").

In fact, manually ranking the HiExpan and TaxoGen taxonomies was simple and obvious, but ranking the CoRel taxonomies was more complex. Such an assessment is delicate; after all, this was the principal motivation of RaTE.

5.3.2 RaTE Ranking

Table 7 showcases the results of MLM taxonomy relation accuracy evaluation, calculated by the number of positive relations over all unique parent-child pairs in a taxonomy.[12]

The entity-masking models m1a and m1b predicted the most positive relationships in each candidate taxonomy while the pre-trained models predicted the fewest, which was expected. It is also surprising that B-b outperforms B-l when it comes to matching more positive concept pairs. Model m2b (trained on two-thirds of the data) expectedly

[11] All HiExpan1 and TaxoGen1&2 parent-child pairs were manually examined, however due to the large size of the word clusters, we had to sample and evaluate concept pairs for CoRel 1-4.

[12] We considered word inflections and certain special cases to improve matching between taxonomy terms and machine predictions, e.g. "veggies", "vegetable" and "vegetables"; "dessert" and "desert".

	Fine-tuned Models						BERT		Majority	RaTE	Manual
	m1a	m1b	m2a	m2b	m0a	m0b	large	base	Voting	ranking	ranking
CoRel1	72.7	71.8	42.4	44.5	46.3	43.6	20.4	27.4	44.3	4	3
CoRel2	78.2	75.0	54.4	53.7	**57.2**	51.2	25.9	36.2	57.2	2	2
CoRel3	60.2	66.7	54.1	54.9	**57.2**	50.1	36.0	40.0	53.5	3	4
CoRel4	68.2	64.6	45.0	39.0	36.5	38.1	**41.0**	41.8	34.7	5	5
HiExpan1	**84.5**	**84.7**	**59.5**	**56.7**	56.9	**64.3**	34.9	**42.0**	**59.0**	1	1
TaxoGen1	13.5	14.7	5.5	6.1	1.2	2.5	3.1	3.7	1.2	6	6
TaxoGen2	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	7	7

Table 7: Relation accuracy scores evaluated by language models, calculated by the number of positive relations, or parent terms in the model predictions, divided by the number of unique parent-child pairs in each taxonomy.

underperforms model m2a, but not drastically.

However, all models produce overall similar score distributions, with the HiExpan taxonomy receiving the highest scores and the TaxoGen taxonomies receiving the lowest. This is consistent with our manual judgements in that the HiExpan concept pairs were derived from an accurate relation dataset (Probase), whereas TaxoGen1 and TaxoGen2 contain mostly noise.

We also compute the majority voting scores for each evaluation target using the six models of Table 4: a concept pair of a taxonomy is positive if and only if three or more models have successfully predicted the parent word. The resulting ranking is reported in the next column, and is shown to correlate well with our manual evaluation (last column).

5.4 Random noise Simulation

To further evaluate the good behaviour of RaTE, we conducted an experiment where we degraded the HiExpan1 taxonomy (the best one we tested). We did this by randomly replacing a percentage of concepts by others. Figure 3 shows that the score (obtained with model m1a) roughly decreases linearly with the level of noise introduced, which is reassuring.

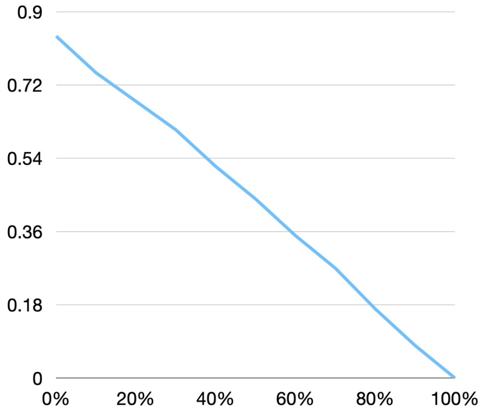


Figure 3: Relation accuracy obtained with model m1a, as a function of the percentage of noise introduced in HiExpan1.

6 Discussion

We presented RaTE, a procedure aimed at automatically evaluating a domain taxonomy without gold standard references or human evaluations. It relies on a large language model and an unmasking procedure for producing annotations. We tested RaTE on the Yelp corpus which gathers restaurant reviews, and found that it correlated well with human judgments, and (artificially) degrading a taxonomy led to a score degradation proportional to the amount of noise injected. Still, we observed that the quality of the language model predictions varies according to the strategies used to fine-tune them.

There remains a number of avenues to investigate. First, we have already identified a number of decisions that could be revisited. In particular, we must test RaTE on other domains, possibly controlling variables such as the size of the fine-tuning material or the frequency of terms. Second, RaTE is an accuracy measure, and depending on the evaluation scenario, it should eventually be coupled with a measure of recall. Last, an interesting avenue is to investigate whether RaTE can be used to optimize the hyper-parameters of an ATC system.

Acknowledgments

Our work has been done in collaboration with IATA, to whom we are truly grateful. We would like to thanks Olena Vasylchenko, Hyuntae Jung and Sorina Radu in particular for their support.

References

2011. Taxonomy induction based on a collaboratively built knowledge repository. *Artif. Intell.*, 175(9-10):1737–1756.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. 2005. Clustering on the Unit Hypersphere using von Mises-Fisher Distributions. *Journal of Machine Learning Research*, 6(9).

Janez Brank, Marko Grobelnik, and Dunja Mladenić. 2005. A Survey of Ontology Evaluation Techniques. In *Proc. of 8th Int. multi-conf. Information Society*, pages 166–169.

C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks. 2004. Data Driven Ontology Evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 641–644, Lisbon, Portugal.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-Fine Entity Typing with Weak Supervision from a Masked Language Model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1790–1799, Online.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

Fellbaum. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Asunción Gómez-Pérez. 1999. Evaluation of taxonomic knowledge in ontologies and knowledge bases. In *Banff Knowledge Acquisition for Knowledge-Based Systems (KAW'99)*, pages 6.1.1–6.1.18.

Nicola Guarino. 1998. Some ontological principles for designing upper level lexical resources. *arXiv preprint cmp-lg/9809002*.

Michael Hanna and David Mareček. 2021. Analyzing BERT's Knowledge of Hypernymy via Prompting. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 275–282.

Marti A Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang, and Jiawei Han. 2020. CoRel: Seed-Guided Topical Taxonomy Construction by Concept Learning and Relation Transferring. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1928–1936.

Kornraphop Kawintiranon and Lisa Singh. 2021. Knowledge Enhanced Masked Language Model for Stance Detection. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 4725–4735.

Yuval Kluger, Ronen Basri, Joseph T Chang, and Mark Gerstein. 2003. Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions. *Genome research*, 13(4):703–716.

Claudia Leacock and Martin Chodorow. 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. *WordNet: An electronic lexical database*, 49(2):265–283.

David A Liem, Sanjana Murali, Dibakar Sigdel, Yu Shi, Xuan Wang, Jiaming Shen, Howard Choi, John H Caufield, Wei Wang, Peipei Ping, et al. 2018. Phrase mining of textual data to analyze extracellular matrix protein patterns across cardiovascular disease. *American Journal of Physiology-Heart and Circulatory Physiology*, 315(4):H910–H924.

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 98)*, pages 296—-304, San Francisco, CA, USA.

Adolfo Lozano-Tello and Asunción Gómez-Pérez. 2004. ONTOMETRIC: A method to choose the appropriate ontology. *Journal of Database Management (JDM)*, 15(2):1–18.

Alexander Maedche and Steffen Staab. 2002. Measuring similarity between ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 251–263. Springer.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic Evaluation of Topic Coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108.

Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, and Chu-Ren Huang. 2022. Discovering Financial Hypernyms by Prompting Masked Language Models. In *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*, pages 10–16.

Robert Porzel and Rainer Malaka. 2004. A Task-based Approach for Ontology Evaluation. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*, pages 1–6. Citeseer.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated Phrase Mining from Massive Text Corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T. Vanni, Brian M. Sadler, and Jiawei Han. 2018. HiExpan: Task-Guided Taxonomy Construction by Hierarchical Tree Expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2180–2189. Association for Computing Machinery.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv preprint arXiv:1904.09223*.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A Probabilistic Taxonomy for Text Understanding. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*, pages 481–492.

Zhibiao Wu and Martha Palmer. 1994. Verb Semantics and Lexical Selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. FinBERT: A Pretrained Language Model for Financial Communications. *arXiv preprint arXiv:2006.08097*.

Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. Taxogen: Unsupervised Topic Taxonomy Construction by Adaptive Term Embedding and Clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709.

The Universal Anaphora Scorer 2.0

Juntao Yu[1], **Michal Novák**[2], **Abdulrahman Aloraini**[3], **Nafise Sadat Moosavi**[4],
Silviu Paun[5], **Sameer Pradhan**[6,7] and **Massimo Poesio**[5]
[1]Univ. of Essex; [2]Charles Univ.; [3]Qassim University; [4]Univ. of Sheffield;
[5]Queen Mary Univ.; [6]LDC, Univ. of Pennsylvania; [7]cemantix.org
j.yu@essex.ac.uk; mnovak@ufal.mff.cuni.cz;
m.poesio@qmul.ac.uk

Abstract

The aim of the Universal Anaphora initiative is to push forward the state of the art both in anaphora (coreference) annotation and in the evaluation of models for anaphora resolution. The first release of the Universal Anaphora Scorer (Yu et al., 2022b) supported the scoring not only of identity anaphora as in the Reference Coreference Scorer (Pradhan et al., 2014) but also of split antecedent anaphoric reference, bridging references, and discourse deixis. That scorer was used in the CODI-CRAC 2021/2022 Shared Tasks on Anaphora Resolution in Dialogues (Khosla et al., 2021; Yu et al., 2022a). A modified version of the scorer supporting discontinuous markables and the COREFUD markup format was also used in the CRAC 2022 Shared Task on Multilingual Coreference Resolution (Žabokrtský et al., 2022). In this paper, we introduce the second release of the scorer, merging the two previous versions, which can score reference with discontinuous markables and zero anaphora resolution.

1 Introduction

The objective of the **Universal Anaphora** initiative, or UA,[1] is to coordinate efforts to push forward the state of the art in anaphora and anaphora resolution beyond identity anaphora,[2] and also covering genres such as dialogue, exemplified by datasets such as ARRAU (Poesio et al., 2018; Uryupina et al., 2020), the CODI-CRAC 2021/2022 corpora (Khosla et al., 2021; Yu et al., 2022a) and GUM (Zeldes, 2017) for English, the Prague Dependency Treebank (its latest version in Hajič et al., 2020) for Czech, and ANCORA for Catalan and Spanish (Recasens and Martí, 2010). The initiative, modelled on Universal Dependencies (UD),[3] aims to achieve this by expanding the aspects of anaphoric interpretation which are or can be reliably annotated in anaphoric corpora, producing unified standards to annotate and encode these annotations, delivering datasets encoded according to these standards, and developing methods for evaluating this type of interpretation. The Universal Anaphora effort has proceeded in close collaboration with the COREFUD initiative (Nedoluzhko et al., 2021, 2022), whose objective is to facilitate research on coreference and anaphora (possibly along with morphology and dependency syntax) by converting corpora in various languages to a unified markup format, fully compatible with UD standards.

An essential prerequisite to make Universal Anaphora-compatible corpora usable in NLP is the availability of scorers that can evaluate the interpretation produced by a system for, e.g., bridging reference (Clark, 1977; Hou et al., 2018; Hou, 2020; Yu and Poesio, 2020; Kobayashi and Ng, 2021), discourse deixis (Webber, 1991; Marasović et al., 2017; Kolhatkar et al., 2018) or split-antecedent anaphora (Eschenbach et al., 1989; Vala et al., 2016; Zhou and Choi, 2018; Yu et al., 2020, 2021). A first step in this direction was the introduction of the Universal Anaphora scorer for anaphoric interpretation (Yu et al., 2022b), the first scorer able to evaluate system performance in all aspects of anaphoric interpretation covered by the current version of the Universal Anaphora proposal. This scorer was used in the CODI-CRAC 2021/2022 Shared Tasks in Anaphora Resolution in Dialogue (Khosla et al., 2021; Yu et al., 2022a) and a revised version supporting COREFUD was used in

[1] http://www.universalanaphora.org

[2] We use the term **identity anaphora** to refer to the subclass of anaphora in which the anaphor refers to the same discourse entity as the antecedent, also known in NLP as 'coreference'. E.g., in *[Geraint Thomas]$_i$'s Giro d'Italia challenge evaporated on the steep slopes of Monte Lussari in north-east Italy. [The Welsh rider]$_i$ was overtaken by his closest challenger, Primoz Roglic.* , the anaphor *The Welsh rider* refers to the same entity as its antecedent, *Geraint Thomas*.

[3] https://universaldependencies.org/

Proceedings of the 15th International Conference on Computational Semantics pages 183–194
June 21–23, 2023. ©2023 Association for Computational Linguistics

the CRAC 2022 Shared Task on Multilingual Coreference (Žabokrtský et al., 2022).

In this paper, we introduce the second version of the Universal Anaphora scorer. This release addresses two key limitations of the first release. The first limitation is the restriction to contiguous mentions, not allowing **discontinuous markables** such as *[a tanker] .. [of orange juice]* in (1.1), consisting of two chunks of text separated from S's uttering *yeah*. Discontinuous markables are common in spoken conversations, but are also used in the CRAFT-CR 2019 biomedical corpus (Cohen et al., 2017) and in corpora such as ARRAU to encode the conjuncts in noun phrases with coordinated heads such as *the students and lecturers from Queen Mary University*, which result in the discontinuous markables *[the students] [from Queen Mary University]* and *[the][lecturers from Queen Mary University].*

	M	:	*... [a tanker]*
Example 1.1	*S*	:	*yeah*
	M	:	*[of orange juice]*

A second limitation of the UA scorer 1.0 is the inability to score the resolution of **zero anaphora** (unrealized arguments) as in (1.2), except in the 'gold' case in which the zero is explicitly marked in the test set.

Example 1.2 (IT) *[Giovanni]$_i$ è in ritardo, così [$\emptyset$]$_i$ mi ha chiesto se posso incontrar[lo]$_i$ al cinema.*
[EN] [John]$_i$ is late so [he]$_i$ asked me if I can meet [him]$_i$ at the movies.

Zero anaphora is annotated in Arabic and Chinese ONTONOTES, and in several of the datasets in the COREFUD collection (Nedoluzhko et al., 2022). In Arabic and Chinese ONTONOTES, zeros are marked using an asterisk * to indicate the position of the empty category in the training data and in the test data in 'gold' mode, but not in the test data in 'predicted' mode, meaning that to evaluate this second mode the scorer must be able to handle 'insertion' of tokens, resulting in evaluation problems (Aloraini et al., 2022).

The new version of the scorer presented in this paper (i) incorporates the treatment of discontinuous markables developed for the COREFUD scorer, testing it also on the CRAFT-CR 2019 corpus; (ii) introduces a novel treatment for the basic form of zero anaphora; and (iii) supports both the COREFUD and UA markup formats.

2 Universal Anaphora And CorefUD

Achievements of the Universal Anaphora initiative so far include a first proposal concerning the range of phenomena to be covered, as well as a survey of the range of existing anaphoric annotations and two proposals for markup formats extending the CONLL-U format developed by **Universal Dependencies** with mechanisms for marking up the range of anaphoric information covered by UA.

2.1 Beyond Identity Anaphora

Most modern anaphoric annotation projects cover basic identity anaphora. However, many other types of identity anaphora exist, as well as other types of anaphoric relations that are annotated in a number of corpora (Novák et al., 2023).

In ONTONOTES, plural reference is only marked when the antecedent is mentioned by a single noun phrase. However, **split-antecedent anaphors** are also possible (Eschenbach et al., 1989; Kamp and Reyle, 1993). These are also cases of plural identity coreference, but to sets composed of two or more entities introduced by separate noun phrases, as in *[John]$_1$ met [Mary]$_2$. [He]$_1$ greeted [her]$_2$. Then [they]$_{1,2}$ went to the movies.*

Discourse deixis (Webber, 1991; Kolhatkar et al., 2018) is the term used to cover both event anaphora, as in *John met Mary. [It]$_1$ happened at 3pm.*, as well as more general types of anaphoric reference to abstract objects not introduced by nominals, as in *John told Mary he was at the office. She didn't believe [that]$_1$* .. Event anaphora is annotated in ONTONOTES and in corpora such as the multi-sentence AMR corpus (O'Gorman et al., 2018). The full range of discourse deixis is annotated in, e.g., ANCORA and ARRAU.

Possibly the most studied of non-identity anaphora is **bridging reference** or **associative anaphora** (Clark, 1977; Hawkins, 1978; Prince, 1981) as in *John looked at the house. [The roof] was thatched.*, where bridging reference / associative anaphora *the roof* refers to an object which is related to / associated with, but not identical to, the *the house*.

2.2 CONLL-UA

The markup format proposed in UA, called CONLL-UA,[4] is based on the CONLL-U-Plus tabular format

[4] https://github.com/UniversalAnaphora/UniversalAnaphora/blob/main/documents/UA_CONLL_U_Plus_proposal_v1.0.md

proposed in Universal Dependencies for corpora containing additional linguistic annotations.[5] The format specifies the following layers in addition to those defined in UD:

- an Identity layer, specifying the entity a markable refers to in the case of a referring markable and, optionally, whether the markable is referring or not, what its head is, and, for split antecedents, the set they belong to;
- a Bridging layer, specifying the anchor, its most recent mention, and, optionally, the associative relation;
- a Discourse_Deixis layer, whose markables specify the non-nominal antecedents of discourse deixis, represented exactly as in the Identity layer. This makes it possible to adopt for discourse deixis the same metrics used for identity anaphora.

The CONLL-UA format was designed to provide a way to specify anaphoric information independent from other layers, but compatible with the UD format. However, at present the UD parser used to validate documents included in UD datasets cannot process the CONLL-U-Plus format. Thus, UA collaborated with COREFUD to design a more 'compact' format that could be used to pack the anaphoric information representable in CONLL-UA in the 'Misc' column of the CONLL-U format, and is fully compatible with the Universal Dependencies. We discuss COREFUD next.

2.3 The CorefUD format

The COREFUD initiative (Nedoluzhko et al., 2022) was launched in parallel with UA to create a collection of corpora annotated with coreferential and other anaphoric relations using a harmonized schema and format. Its current version COREFUD 1.1 (Novák et al., 2023) consists of 17 datasets for 12 languages in its publicly available edition.[6]

Whereas UA is primarily focused on anaphora, COREFUD has another objective besides harmonization of the coreference datasets, namely, to intersect the world of coreference with the world of syntax. This is achieved by augmenting the coreference data with morpho-syntax annotation compliant with the UD standards, which has been obtained automatically for the datasets that do not contain such manual annotation. This is motivated not only pragmatically (popularity of UD and standards for numerous technical issues), but it is also grounded theoretically. For instance, entity mentions often correspond to syntactically relevant notions (e.g. noun phrase, subject), some coreference relations are manifested mainly by syntactic means (e.g. reflexive and relative constructions), and zero expressions (e.g. pro-drops) are vital for coreference in many languages.

After developing a first format in COREFUD 0.1 (Nedoluzhko et al., 2021) independently from the UA initiative[7], a new format was jointly developed and introduced with COREFUD 1.0 (Nedoluzhko et al., 2022). This format can encode essentially the same information as CONLL-UA, but this information is encoded in the Misc column, which makes it possible to pass the official UD validation at level 2 (passing the higher levels is not possible with automatically predicted POS tags and dependency relations).[8] One remaining difference is that COREFUD has been from its very beginning designed to represent existing data in datasets including dependency graphs. Thus, it can capture zero expressions by stipulating 'empty tokens' and referencing them using enhanced dependency graphs, whereas in CONLL-UA, which does not require dependency layers, empty tokens are bound to the surface tokens by their relative position.

The COREFUD collection is accompanied with API implemented within the Udapi framework[9] that facilitates manipulation with the data in COREFUD format as well as its visualization.

3 The Universal Anaphora Scorer 1.0

The Universal Anaphora (UA) 1.0 scorer (Yu et al., 2022b) is a Python scorer for the varieties of anaphoric reference covered by the Universal Anaphora guidelines: identity anaphora, split antecedent plurals, identification of non-referring expressions, bridging reference, and discourse deixis.

For identify reference, the scorer builds on the original Reference Coreference scorer [10] (Pradhan

[5] https://universaldependencies.org/ext-format.html

[6] In total, 21 datasets for 13 languages, including datasets with non-public licences, e.g. ONTONOTES and ARRAU.

[7] This format, which substantially differs from the current format, is described in: https://ufal.mff.cuni.cz/~popel/corefud-1.0/corefud-1.0-format.pdf.

[8] https://universaldependencies.org/validation-rules.html#levels-of-validity

[9] https://github.com/udapi/udapi-python

[10] https://github.com/conll/reference-coreference-scorers

et al., 2014) and its reimplementation in Python by Moosavi,[11] developed for the CRAC 2018 shared task (Poesio et al., 2018). The Reference Coreference scorer, developed for use in the CONLL 2011 and 2012 shared tasks on the ONTONOTES corpus (Pradhan et al., 2012), implemented the best known metrics for identity anaphora (coreference): MUC (Vilain et al., 1995), B^3(Bagga and Baldwin, 1998), CEAF (Luo, 2005), and BLANC (Recasens and Hovy, 2011). The Reference Coreference scorer popularized scoring by using the average F1 value of MUC, B^3 and CEAF, as originally proposed by (Denis and Baldridge, 2009)–so much so that this average, originally known as MELA, has since become known as the CONLL metric. Moosavi's CRAC 2018 scorer, apart from being written in Python, also implemented the LEA metric (Moosavi and Strube, 2016) and provided a separate score for the interpretation of non-referring expressions.

3.1 Identity Reference

In the CONLL-UA format, identity reference is specified in the `Identity` column, which specifies the cluster id (`EntityID`), markable id (`MarkableID`), the minimum span (`Min`) and the semantic type (`SemType`) (non-referring types, discourse new (`dn`) and discourse old (`do`)) of the mention. Split-antecedent information is annotated on the antecedents's row using an '`ElementOf`' attribute that specifies the cluster id of the split antecedent plural anaphor. This is illustrated in the following example:

```
(EntityID=10|\
MarkableID=markable_11|\
Min=5|\
SemType=do|\
ElementOf=23)
```

The UA 1.0 scorer computes all major metrics for identity reference including MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF (Luo, 2005), CONLL (the unweighted average of MUC, B^3, and CEAF) (Pradhan et al., 2014), BLANC (Luo et al., 2014; Recasens and Hovy, 2011), and LEA (Moosavi and Strube, 2016) scores.

Three score-reporting options are available: The first option mirrors the evaluation used in the CONLL shared tasks (Pradhan et al., 2012) which excludes singletons and split-antecedents from evaluation. The second option is the one used in the identity anaphora sub-task of the CRAC 2018 shared task (Poesio et al., 2018). This evaluation includes singletons, but not split-antecedents. Finally, the scorer can include both singletons and split-antecedent anaphors; this is the format used in CODI-CRAC 2021/2022 (Khosla et al., 2021; Yu et al., 2022a). Clusters include both split-antecedents and singletons. For split antecedents, a generalization of the existing coreference metrics was developed (Paun et al., 2023).

3.2 Split Antecedent Anaphora

The UA scorer implements a new method proposed by Paun et al. (2023), for scoring split-antecedent anaphora based on treating the antecedents of split-antecedent anaphors as a new type of mention, **accommodated sets**–set denoting entities which have the split antecedents as elements.

3.3 Non-referring expressions

A key aspect of anaphoric interpretation is correctly determining whether nominal phrases like markable *it* in Example 3.1 are referring or not, and to distinguish such noun phrases from singletons.

Example 3.1 *[It] was late at night.*

The semantic type (`SemType`) attribute is used to specify the non-referring type in detail for corpora such as ARRAU or CODI-CRAC 2021/2022 in which such distinctions are made (e.g. predicate, idiom). The new UA scorer follows the scorer developed for the CRAC 2018 shared task in that non-referring expressions are not treated as singletons in the evaluation of identity reference. Instead, non-referring expressions are separated from identity references when inputted to the scorer. More specifically, the collection of non-referring expressions in both the key and the response is identified and the scorer computes an F1 score for non-referring expressions only. The F1 score for non-referring expression is reported separately from the F1 scores for identity reference.

3.4 Discourse Deixis

The UA scorer supports the extension to discourse deixis proposed in version 1.0 of the Universal Anaphora specification of anaphoric phenomena by implementing an entirely new approach to evaluation of discourse deixis supporting the evaluation. This new approach is enabled by the way discourse deixis is encoded in the UA markup.

In the UA markup, discourse deixis is specified in the `Discourse_deixis` column of the 'exploded' format, and the same attributes are used as

[11] https://github.com/ns-moosavi/coval

for the Identity column. The only difference is that the cluster id (EntityID) and the markable id (MarkableID) of the segments are highlighted with a '-DD' suffix and 'dd_' prefix respectively, to avoid confusion in visual inspection.

This representation enables the application of coreference metrics to evaluate discourse deixis. Particularly given that our new scorer provides a way to incorporate split-antecedents into the standard metrics, which therefore are discourse deixis-ready. This is exactly how the UA scorer evaluates discourse deixis: it computes the same MUC, B^3, CEAF, CONLL, BLANC and LEA metrics as for identity anaphora.

3.5 Bridging References

In UA format, bridging references are specified in the Bridging column of the 'exploded' format. The attributes for bridging include the markable ID (MarkableID), a mention of anchor entity (MentionAnchor), the cluster id of the antecedent (EntityAnchor) and the bridging relationship (Rel). For example:

```
(MarkableID=markable_9|\
Rel=subset-inv|\
MentionAnchor=markable_1|\
EntityAnchor=3)
```

For bridging references, the scorer reports three scores: the two metrics computed by the scorer used for CRAC 2018 shared task – mention-based F1 and entity-based F1 – and, in addition, anaphora recognition F1. Mention-based F1 for bridging evaluates a system's ability to predict the correct anaphora and the mention of the anchor specified in the annotation (this is usually the closest or most suitable mention). Entity-based F1 is more relaxed than mention-based F1, and does not require the system to predict exactly the same mention as the gold annotation. Finally, anaphora recognition F1 is used to assess the system's ability to identify bridging anaphors.

4 The CorefUD Scorer 1.0

CorefUD scorer 1.0 was used in the CRAC 2022 Shared Task on Multilingual Coreference Resolution (Žabokrtský et al., 2022). It is based on the Universal Anaphora Scorer 1.0, reusing the implementations of all generally used coreferential measures without any modification. This guarantees that the measures are computed in exactly the same way. Nevertheless, CorefUD scorer is capable of processing the coreference annotation files in the CorefUD 1.0 format.

Among other things, it allows evaluation of coreference for zeros. Nonetheless, its version 1.0 is not able to handle a response document whose tokens are not completely identical to the tokens in the key document. This holds also for empty tokens, which virtually prevents the scorer to evaluate response documents where the zero expressions are automatically predicted.

Moreover, the CorefUD scorer re-defines matching of key and response mentions in the way to be able to process potentially discontinuous mentions, which are present in some CorefUD datasets. Instead of comparing mention boundaries, matching is based on set/subset relations between the tokens of the mentions in question.

Last but not least, the CorefUD scorer introduced two new scores. The MOR score measures to what extent key and response mentions match, no matter to which coreference entity they belong. The CorefUD scorer also implements the anaphor-decomposable scoring schema introduced by Tuggener (2014) and applies it to zeros. This allows for measuring the quality of predicting any of the antecedents of zero anaphors.

5 The UA Scorer 2.0

The UA scorer 2.0 merges the functions of the UA scorer 1.0 and CorefUD scorer 1.0 to make them a unified scorer. It also optimises/extends the scorer's ability on handling discontinuous markables and zeros, e.g. the new scorer can handle zeros in the predicted setting and can reproduce the CRAFT-CR 2019 shared task results. We introduce the details of the implementations in the next subsections.

5.1 Discontinuous Markables

In CONLL-UA, discontinuous markables can be used in both the Identity and Discourse_Deixis columns by sharing the MarkableID between the different sub-spans of a discontinuous markable. The scorer can then recognise the discontinuous markables from the text. For example, if a discontinuous markable consists of two continuous spans, the two spans will have the same Identity column, e.g. same EntityID, MarkableID, Min and SemType.

COREFUD format does not assign IDs to markables. Each continuous part of a discontinuous markable is thus labeled by its ordinal number and the total number of parts in square brackets just after the cluster ID: `Entity=(10[1/2] ... Entity=10[1/2]) ... Entity=(10[2/2] ... Entity=10[2/2])`.

Since coreference evaluation metrics are developed based on the assumption that mentions in the key and response are aligned implicitly, the scorer provides two mention alignment strategies during the evaluation: 'strict' and 'partial'. In a 'strict' setting mentions are aligned only if all parts of the discontinuous markables are recognised correctly by the system. In the 'partial' setting, mentions can be aligned using a specified fuzzy matching algorithm. To use the 'partial' matching, the `Min/head` span for each mention needs to be specified in the key files. The `Min/head` span is specified as the minimum string that a coreference resolver must identify for the corresponding markable (either discontinuous or continuous). Allowing 'partial' mention alignment is especially useful for evaluating discontinuous mentions, given that it is more complex to predict, and most of the current coreference systems cannot predict the discontinuous markables.

To be more specific, the scorer provides two algorithms to align the mentions in 'partial' settings. By default, a mention in the response is considered a candidate for a gold mention if it contains the MIN/head string and does not go beyond the annotated maximum boundary. To align the mentions in the key and response, we first align the mentions based on the exact matching to exclude them from the partial matching step. Secondly, to align the remaining mentions, we compute the recall (the precision will always be 100% according to our definition of partial matching) between all remaining mention pairs between key and their corresponding candidates in the response to create a recall matrix. Finally, the recall matrix is used with the Kuhn-Munkres algorithm (Kuhn, 1955; Munkres, 1957) to find the best alignment between those mentions. After the alignment between the mentions is found, the coreference evaluation metrics can be used as normal.

To facilitate the research in the biomedical domain we also provide an option to align the mentions using the same algorithm as in CRAFT-CR 2019 shared task (Baumgartner et al., 2019) The CRAFT-CR 2019 corpus consists of biomedical files with coreference relations (including discontinuous markables) annotated. The algorithm used to align mentions in CRAFT-CR 2019 shared task considers a predicted mention correct if any continuous span of the predicted mention overlaps with and does not go beyond the first span of the key mention. Their algorithm does not impose a one-to-one alignment between mentions hence one key mention might be aligned with multiple predicted mentions and vice versa.

By default, if a corpus consists of discontinuous markables the system will use the 'strict' setting to evaluate them. The `-p|--partial-match` option can be used to enable the default partial matching algorithm. To use the CRAFT-CR 2019 algorithm, the `--partial-match-method` option needs to be set to `craft`.

5.2 Zeros

In both 'exploded' and 'compact' format, zeros are represented using the UD standard of empty nodes, in which the first column (`ID`, word index) is indicated using the decimal numbers. For instance, if we have a zero anaphora right after a token whose `ID` is 5, we index the zero with 5.1 instead of 6 used for a normal token. The scorer identifies the zeros by the decimal indexing and has the option to include zeros in the evaluation.

When zeros are included in the evaluation, again we need to align them between the key and response. Currently, the scorer performs the alignment based on the position of the zeros, i.e. zeros are aligned if they are located in the same position in the sentences. This is based on the assumption that the position of the zeros is not random, and the corpus which have zeros annotated has a consistent guideline on where should the zeros be positioned. We are also considering another approach that uses dependency relations to align the zeros, in which the position of zero does not need to follow a certain rule. However, due to the complication of this approach, we are not able to include it in this release and are planning to make it available in the next version of the scorer.

By default zeros are excluded in the evaluation, to include them the `-z|--keep-zeros` options can be specified.

5.3 Formats

The scorer supports three formats: CONLL 2012, CONLL-UA (UA 'exploded') and COREFUD (UA 'compact'). The CONLL-UA format is the default format for the scorer that support all anaphora relations assessed by the scorer e.g. singletons, non-referring expressions, split-antecedents, bridging reference and discourse deixis. The parser of the COREFUD format supports identity relations including discontinuous markables and zeros but does not support split-antecedents and non-referring expressions. The CONLL 2012 format only support continuous markables in the identity relation.

5.4 Shared Tasks Support

As the number of shared tasks supported by the scorer grows, the options also increase. To simplify the usage of the scorer we provide shortcuts for all coreference shared tasks supported by the scorer. The `-t|--shared-task` option can be used to specify the evaluation settings for the shared task in question. In total, the scorer supports 7 different settings used in 5 shared tasks:

- `conll12`: This evaluation mode is compatible with the coreference evaluation of the CONLL 2012 shared task in which only coreferring markables are evaluated.

- `crac18`: The evaluation method used in CRAC 2018 shared task. In this evaluation setting, coreference relations, singletons and non-referring mentions are taken into account for evaluation.

- `craft19`: This evaluation mode is used by the CRAFT 2019 shared task, it includes coreference relations, singletons and discontinuous markables.

- `crac22`: The evaluation method used as the primary metric by the CRAC 2022 shared task on multilingual coreference resolution. The evaluation applies partial matching and includes coreference relations, discontinuous markables, and zeros but excludes singletons and split-antecedents

- `codicrac22ar`: The evaluation method used by the anaphora resolution track of the CODI-CRAC 2021/2022 shared tasks. In this mode, both coreferring markables, split-antecedents and singletons are evaluated by the specified evaluation metrics.

- `codicrac22br`: The evaluation method used by the bridging resolution track of the CODI-CRAC 2021/2022 shared tasks. In this evaluation setting only bridging references will be evaluated.

- `codicrac22dd`: The evaluation method used by the discourse deixis track of the CODI-CRAC 2021/2022 shared tasks. The discourse deixis column is evaluated using the same method as `codicrac22ar`.

6 Results

In this section we demonstrate the scorer in practice by using it to score the submissions to two shared tasks that involved discontinous markables and zeros, CRAC 2022 and CRAFT-CR 2019.

6.1 CRAC 2022 Shared Task

We tested the new UA scorer on the submissions to the CRAC 2022 Shared Task on Multilingual Coreference Resolution (Žabokrtský et al., 2022), namely on the predictions of the winning setup of the CorPipe system (Straka and Straková, 2022).

Table 1 shows the performance of the winning submission evaluated on the shared task testset in terms of F-scores of multiple standard coreferential metrics macro-averaged over all datasets in the testset. We compare the measured performance to the scores calculated by the COREFUD scorer 1.0, the official scorer of the shared task, using ‘strict’ and ‘partial’ setting (denoted as exact and partial matching, respectively, in the CRAC 2022 shared task). Apart from the standard scores, it also compares the values of the anaphor-decomposable score for zeros and the MOR score, calculating the average overlap of key and response markables.

Firstly, note that all scores obtained with the ‘strict’ setting are significantly lower than those calculated with the ‘partial’ setting. It results from artificial reduction of system mentions to their heads done by the CorPipe system. They pursued this strategy in order to perform better in terms of the official metric, computed using partial matching.

Secondly, the comparison of pairs of corresponding scores measured by the two scorers confirms that the UA scorer implements processing of the COREFUD format including discontinuous markables correctly, exemplified by the identical scores with respect to the ‘strict’ setting. On the other hand, it also shows that partial matching is treated in a slightly different way, leading to consistently lower scores measured by UA scorer. The reason is that, unlike COREFUD scorer 1.0, the new UA scorer imposes one-to-one alignment when matching potentially overlapping markables.

Finally, the only mismatch for the ‘strict’ setting occurs in the MOR score. The two scorers in

	Exact		Partial	
Metrics	CorefUD	UA	CorefUD	UA
MUC	34.20	34.20	74.18	73.98
B^3	29.40	29.40	68.34	68.08
CEAF$_e$	35.93	35.93	69.64	69.40
CEAF$_m$	40.86	40.86	71.24	71.04
BLANC	28.39	28.39	64.86	64.35
LEA	22.68	22.68	65.02	64.78
CoNLL F1	33.18	33.18	70.72	70.49
Zero	60.42	60.42	83.65	83.15
MOR	45.37	26.76	45.37	44.75

Table 1: Comparison between the UA scorer and the COREFUD scorer.

fact use different mapping between key and system mentions. Whereas UA scorer uses the same mapping as for the other scores, which is based either on exact or partial matching, COREFUD scorer employs one-to-one mapping that maximizes the number of overlapping tokens regardless of the chosen matching. Two mentions that do not match even partially may still overlap. Consequently, the MOR scores outputted by COREFUD scorer are the same for each of the matching type as well as higher than those produced by the UA scorer.

6.2 CRAFT-CR 2019 Shared Task

Since the system outputs of the CRAFT-CR 2019 shared task are not publicly available, we have to find the system outputs elsewhere. We obtained the system output of the best-performing system from Lu and Poesio (2021) to compare the evaluation results between our scorer and the CRAFT-CR 2019 scorer[12] in both ‘strict’ and ‘partial’ mention matching settings.

Table 2 shows the comparison, as we can see from the ‘strict’ evaluation setting our scorer has the same results as their scorer. For the ‘partial’ setting we find their original scorer produces slightly different results if we run the scorer multiple times, whereas our scorer always produces the same results. The difference between the two scorers is within the range of the difference between two different runs of the original scorer. Hence we are convinced that the new scorer follows the same algorithm as the original scorer and can be used as a replacement for the original scorer.

	Strict		Partial	
Metrics	CRAFT	UA	CRAFT	UA
MUC	57.69	57.69	59.74	59.78
B^3	45.43	45.43	48.03	48.02
CEAF$_e$	39.89	39.89	42.89	42.89
CEAF$_m$	51.26	51.26	53.19	53.20
BLANC	46.29	46.29	49.68	49.76
LEA	42.34	42.34	44.15	44.14
CoNLL F1	47.67	47.67	50.22	50.23

Table 2: The comparison between the UA scorer and the CRAFT-CR 2019 scorer.

7 Conclusion and Future Work

The new version of the Universal Anaphora scorer presented in this paper makes further progress towards the goal of providing the community with methods for evaluating systems carrying the full range of anaphoric interpretation. This version builds on the results of three separate shared tasks and additional research that enabled the Universal Anaphora community to test the scorer not only for a variety of types of anaphoric interpretation, but also for a range of genres covering dialogue (Khosla et al., 2021; Yu et al., 2022a) and biomedical text (Lu and Poesio, 2021), and for a variety of languages including Arabic (Aloraini et al., 2022) and the 13 languages covered in COREFUD (Žabokrtský et al., 2022). It revealed a number of limitations with the previous version of the scorer that needed addressing. We hope the community will take advantage of the new scorer to broaden the range of research on multilingual, multi-genre anaphoric interpretation.

Acknowledgements

Juntao Yu, Silviu Paun and Massimo Poesio were funded in part by the DALI project, ERC Grant 695662, and in part by the ARCIDUCA project, EPSRC grant EP/W001632/1. Michal Novák was funded by Grant 20-16819X (LUSyD) of the Czech Science Foundation (GAČR) and by FW03010656 of the Technology Agency of the Czech Republic. The work described herein has also been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

References

Abdulrahman Aloraini, Sameer Pradhan, and Massimo Poesio. 2022. Joint coreference resolu-

[12] https://github.com/bill-baumgartner/reference-coreference-scorers

tion for zeros and non-zeros in Arabic. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 11–21, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation (LREC) - Workshop on linguistics coreference*, volume 1, pages 563–566. ACL.

William Baumgartner, Michael Bada, Sampo Pyysalo, Manuel R. Ciosici, Negacy Hailu, Harrison Pielke-Lombardo, Michael Regan, and Lawrence Hunter. 2019. CRAFT shared tasks 2019 overview — integrated structure, semantics, and coreference. In *Proceedings of the 5th Workshop on BioNLP Open Shared Tasks*, pages 174–184, Hong Kong, China. Association for Computational Linguistics.

Herbert H. Clark. 1977. Bridging. In P. N. Johnson-Laird and P.C. Wason, editors, *Thinking: Readings in Cognitive Science*, pages 411–420. Cambridge University Press, London and New York.

Kevin Bretonnel Cohen, Arrick Lanfranchi, Miji Joo-young Choi, Michael Bada, William A. Baumgartner Jr., Natalya Panteleyeva, Karin Verspoor, Martha Palmer, and Lawrence E. Hunter. 2017. Coreference annotation and resolution in the Colorado Richly Annotated Full Text (CRAFT) corpus of biomedical journal articles. *BMC Bioinformatics*, 18(372).

Pascal Denis and Jason Baldridge. 2009. Global joint models for coreference resolution and named entity classification. *Procesamiento del Lenguaje Natural*, 42:87–96.

Carola Eschenbach, Christopher Habel, Michael Herweg, and Klaus Rehkämper. 1989. Remarks on plural anaphora. In *Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics*, pages 161–167. Association for Computational Linguistics.

Jan Hajič, Eduard Bejček, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpá nková. 2020. Prague Dependency Treebank - Consolidated 1.0. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, pages 5208–5218, Marseille, France. European Language Resources Association.

John A. Hawkins. 1978. *Definiteness and Indefiniteness*. Croom Helm, London.

Yufang Hou. 2020. Bridging anaphora resolution as question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1428–1438, Online. Association for Computational Linguistics.

Yufang Hou, Katja Markert, and Michael Strube. 2018. Unrestricted bridging resolution. *Computational Linguistics*, 44(2):237–284.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic*. D. Reidel, Dordrecht.

Sopan Khosla, Juntao Yu, Ramesh Manuvinakurike, Vincent Ng, Massimo Poesio, Michael Strube, and Carolyn Rosé. 2021. The codi-crac 2021 shared task on anaphora, bridging, and discourse deixis in dialogue. In *Proc. of the CODI/CRAC Shared Task Workshop*.

Hideo Kobayashi and Vincent Ng. 2021. Bridging resolution: Making sense of the state of the art. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1652–1659, Online. Association for Computational Linguistics.

Varada Kolhatkar, Adam Roussel, Stefanie Dipper, and Heike Zinsmeister. 2018. Anaphora with non-nominal antecedents in computational linguistics: a Survey. *Computational Linguistics*, 44(3):547–612.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Pengcheng Lu and Massimo Poesio. 2021. Coreference resolution for the biomedical domain: A survey. In *Proc. of the CRAC Workshop*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British

Columbia, Canada. Association for Computational Linguistics.

Xiaoqiang Luo, Sameer Pradhan, Marta Recasens, and Eduard Hovy. 2014. An extension of BLANC to system mentions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 24–29, Baltimore, Maryland. Association for Computational Linguistics.

Ana Marasović, Leo Born, Juri Opitz, and Anette Frank. 2017. A mention-ranking model for abstract anaphora resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 221–232, Copenhagen, Denmark. Association for Computational Linguistics.

Nafise S. Moosavi and Michael Strube. 2016. A proposal for a link-based entity aware metric. In *Proc. of ACL*, pages 632–642, Berlin.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, Amir Zeldes, and Daniel Zeman. 2022. CorefUD 1.0: Coreference meets Universal Dependencies. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4859–4872, Marseille, France. European Language Resources Association.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, and Daniel Zeman. 2021. Coreference meets universal dependencies – a pilot experiment on harmonizing coreference datasets for 11 languages. ÚFAL Technical Report TR-2021-66, Charles University, Prague.

Michal Novák, Martin Popel, Zdeněk Žabokrtský, Daniel Zeman, Anna Nedoluzhko, Kutay Acar, Peter Bourgonje, Silvie Cinková, Gülşen Cebiroğlu Eryiğit, Jan Hajič, Christian Hardmeier, Dag Haug, Tollef Jørgensen, Andre Kåsen, Pauline Krielke, Frédéric Landragin, Ekaterina Lapshinova-Koltunski, Petter Mæhlum, M. Antònia Martí, Marie Mikulová, Anders Nøklestad, Maciej Ogrodniczuk, Lilja Øvrelid, Tuğba Pamay Arslan, Marta Recasens, Per Erik Solberg, Manfred Stede, Milan Straka, Svetlana Toldova, Noémi Vadász, Erik Velldal, Veronika Vincze, Amir Zeldes, and Voldemaras Žitkus. 2023. Coreference in universal dependencies 1.1 (CorefUD 1.1). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. Amr beyond the sentence: the multi-sentence amr corpus. In *Proc. of COLING*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Silviu Paun, Juntao Yu, Nafise Moosavi, and Massimo Poesio. 2023. Scoring coreference chains with split-antecedent anaphors and other entities constructed from a discourse model. *Dialogue and Discourse*.

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. 2018. Anaphora resolution with the ARRAU corpus. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 11–22, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.

Ellen F. Prince. 1981. Toward a taxonomy of given-new information. In P. Cole, editor, *Radical Pragmatics*, pages 223–256. Academic Press, New York.

Marta Recasens and Ed Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*.

Marta Recasens and M. Antònia Martí. 2010. AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan. *Language Resources and Evaluation*, 44(4):315–345.

Milan Straka and Jana Straková. 2022. ÚFAL CorPipe at CRAC 2022: Effectivity of multilingual models for coreference resolution. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*, pages 28–37, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Don Tuggener. 2014. Coreference resolution evaluation for higher level applications. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 231–235, Gothenburg, Sweden. Association for Computational Linguistics.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J. Rodriguez, and Massimo Poesio. 2020. Annotating a broad range of anaphoric phenomena, in a variety of genres: the ARRAU corpus. *Journal of Natural Language Engineering*.

Hardik Vala, Andrew Piper, and Derek Ruths. 2016. The more antecedents, the merrier: Resolving multi-antecedent anaphors. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2287–2296, Berlin, Germany. Association for Computational Linguistics.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Bonnie L. Webber. 1991. Structure and ostension in the interpretation of discourse deixis. *Language and Cognitive Processes*, 6(2):107–135.

Juntao Yu, Sopan Khosla, Ramesh Manuvinakurike, Lori Levin, Vincent Ng, Massimo Poesio, Michael Strube, and Massimo Poesio. 2022a. The CODI/CRAC 2022 shared task on anaphora resolution, bridging and discourse deixis in dialogue. In *Proc. of CODI/CRAC Shared Task*.

Juntao Yu, Sopan Khosla, Nafise Moosavi, Silviu Paun, Sameer Pradhan, and Massimo Poesio. 2022b. The universal anaphora scorer 1.0. In *Proc. of LREC*.

Juntao Yu, Nafise Sadat Moosavi, Silviu Paun, and Massimo Poesio. 2020. Free the plural: Unrestricted split-antecedent anaphora resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6113–6125, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Juntao Yu, Nafise Sadat Moosavi, Silviu Paun, and Massimo Poesio. 2021. Stay together: A system for single and split-antecedent anaphora resolution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Juntao Yu and Massimo Poesio. 2020. Multitask learning based neural bridging reference resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3534–3546, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zdeněk Žabokrtský, Miloslav Konopík, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondřej Pražák, Jakub Sido, Daniel Zeman, and Yilun Zhu. 2022. Findings of the shared task on multilingual coreference resolution. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*, pages 1–17, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Ethan Zhou and Jinho D. Choi. 2018. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 24–34, Santa Fe, New

Mexico, USA. Association for Computational Linguistics.

The Sequence Notation: Catching Complex Meanings in Simple Graphs

Johan Bos
Center for Language and Cognition
University of Groningen
johan.bos@rug.nl

Abstract

Current symbolic semantic representations proposed to capture the semantics of human language have served well to give us insight in how meaning is expressed. But they are either too complicated for large-scale annotation tasks or lack expressive power to play a role in inference tasks. What I propose is a meaning representation system that it is interlingual, model-theoretic (by translation to first-order logic), and variable-free. It divides the labour involved in representing meaning along three levels: concept, roles, and contexts. As natural languages are expressed as sequences of phonemes or words, the meaning representations that I propose are likewise sequential. However, the resulting meaning representations can also be visualised as directed acyclic graphs.

1 Introduction

There are many proposals for representing meaning of natural language expressions in a formal way. These originate from various disciplines, including formal semantics (Thomason, 1974; Dowty et al., 1981; Heim, 1982; Kamp, 1984; Groenendijk and Stokhof, 1990; Chierchia, 1992), artificial intelligence (Schubert, 1976; Sowa, 1984, 1995; Schubert, 2015), and computational linguistics (Copestake et al., 2005; Banarescu et al., 2013; Abzianidze et al., 2017; Martínez Lorenzo et al., 2022). Although most of these do a tremendous job in analysing meaning, I think none of them offers a meaning representation that is the ideal candidate for large-scale annotation tasks in computational semantics requiring supervised machine learning: some of them lack expressive power, some of them are only partially interpretable, some of them are tailored to specific natural languages, and yet others are featured with a complex syntax that makes them unsuitable for human annotation tasks.

What nearly all of these semantic formalisms have in common is that they all share the property of using *variables* ranging over (first-order or higher-order) entities. Representations without variables have potential advantages and benefits when we think of human annotation efforts, machine learning approaches, and meaning visualisations techniques. Hence, the question I take at heart is whether it is possible to eliminate variables from formal meaning representations without losing expressive power required to interpret linguistic expressions.

The goal of this paper is to propose a meaning representation that is a healthy mixture of interlinguality, simplicity, and expressiveness. With interlinguality I mean a meaning representation that is not designed to support a single language. With simplicity I mean a kind of semantic representation that supports an intuitive way of drawing a graphical representation of the meaning that it is supposed to represent. With expressiveness I mean at least the expressive power of first-order logic (i.e., quantification, negation, and conjunction) and support for discourse phenomena such as co-reference and discourse structure.

Current graph-based meaning representations such as AMR, Abstract Meaning Representation (Banarescu et al., 2013) lack expressive power. Current logic-based meaning representations such as DRS, Disourse Representation Structure (Kamp and Reyle, 1993) are unattractive to represent as graphs as they require substantial reification (Abzianidze et al., 2020). What I propose is a formalism that combines AMR with DRS while removing notational redundancies such as variables and punctuation symbols. It takes the attractive and simple graph-based visualisation of AMR but adds the "boxes" of DRS, arriving at a formalism that includes negation and quantification as in predicate logic. The formalism accommodates two ways

Proceedings of the 15th International Conference on Computational Semantics pages 195–208
June 21–23, 2023. ©2023 Association for Computational Linguistics

of represent meaning: the variable-free sequential notation, and directed acyclic graphs. The variable-free sequence notation is expected to be advantageous for human annotation efforts and language technology applications that require machine learning (e.g., applying neural networks for the tasks of semantic parsing or natural language generation). This is because it doesn't require the process of using variables nor explicit indication of scope for logical operators like negation. The graph representation is convenient for human readers and for software designed to work with graphs.

2 Simplifying Meaning Representations

In this section I will present a new meaning representation system. Using this formalism, annotation can be done with a simple text editor. There are no logical variables but there is still support for negation and scope. The primary encoding of meaning is done in sequence notation. But the meanings can be visualised as directed acyclic graphs. The sequence notation can be applied to various meaning representation formalisms including AMR and DRS. In this paper I focus on the latter.

2.1 The Sequence Notation

I will introduce the sequence notation by first explaining what the elementary building block are. Then I explain how sequences can be constructed, visualised, and interpreted. The sequence notation has the following ingredients (with examples in brackets):

- Concepts (`cat.n.01`, `see.v.03`, ...)
- Constants (`"Mary"`, `speaker`, `20`, π, ...)
- Roles (`Agent`, `Theme`, `Patient`, ...)
- Operators (=, $\neq$, $\approx$, <, $\leq$, $\prec$, ...)
- Indices (..., `-2`, `-1`, `+1`, `+2`, ...)
- Contexts
- Separators (`NEGATION`, `CONJUNCTION`, `EXPLANATION`, `NARRATION`, ...)
- Connectors (..., `<2`, `<1`, `>1`, ...)

Concepts identify an entity or event as belonging to a certain class within a domain ontology. Concepts are always written in lower case and are represented as interlingual WordNet synsets as triplets comprising of a lemma, part of speech (n, v, a, or r) and a sense number, e.g., `cat.n.01` represents the first sense of the noun cat. I view a WordNet synset as language-neutral, even though in this paper I will use the synsets as defined in Princeton's American English WordNet 3.0 (Fellbaum, 1998) because of its common use in the NLP community. Adoption of a multi-lingual wordnet (Navigli and Ponzetto, 2012; Bond and Foster, 2013) would eventually be the target in a large-scale multi-lingual implementation.

Constants comprise proper names (of people, animals, organisations, locations, artifacts), numerical values (integers and reals), times and dates, literal mentions. They also include deictic references: the speaker of the utterance (`speaker`), the addressee (`hearer`), the utterance time (`now`) and location (`here`).

Roles connect an event to an entity (or relate two entities to each other). Roles always start with an uppercase character followed by lowercase to distinguish them from concepts. The roles used in this paper are by and large based on thematic role inventory provided by VerbNet and LIRICS (Kipper et al., 2008; Bonial et al., 2011). The connections between events and entities are established with *indices* (see § 3.3). The operators are used to express comparisons between entities and are written in mathematical notation or with three uppercase letters (EQU, NEQ, SIM, LES, LEQ, TPR, and so on).

All concepts are introduced in a *context*. Contexts are not explicit in sequence notation. A *separator* introduces a new context connecting it to a previously introduced context as indicated by its *connector* (see § 3.4). Separators are always written in all uppercase to distinguish them from roles and concepts.

2.2 Forming Sequences

A role followed by a constant is an *anchor*. So, `Name "Mary"` is an anchor. A role followed by an index is a *hook*. Hence, `Owner +1` is a hook. A *simple sequence* is a sequence of one or more concepts, where a concept can be followed by zero or more anchors or hooks. Therefore, `dog.n.01` is a simple sequence, and so are `cat.n.01 dog.n.01`, and `cat.n.01 Owner +1 person.n.01 Name "Mary"`. A simple sequence represents a single *context*. A context is similar to a box in Discourse Representation Theory (Kamp and Reyle, 1993). They set the stage for the entities that play a part of the context.

A *complex sequence* is formed by combining two (simple or complex) sequences using a separator and connector. For instance, `person.n.01 NEGATION <1 smile.v.01 Theme -1` is a complex sequence, constructed from the simple sequences `person.n.01` and `smile.v.01 Theme -1` using the separator `NEGATION` and connector `<1` as glue. A complex sequence represents two or more contexts.

2.3 Graph Visualisation

A meaning in sequence notation can be visualised as a directed acyclic graph, where the vertices denote concepts, contexts or constants, and the edges are decorated by roles or comparison operators. Concept nodes are drawn as ovals and context nodes as boxes. Figure 1 shows how the sequence `male.n.02 Name "Tom" time.n.08 TPR now cry.v.02 Agent -2 Time -1` is visusalised as a graph.

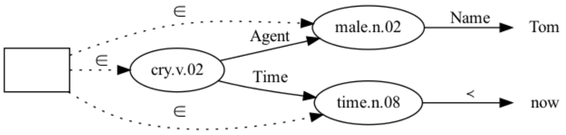


Figure 1: Graph for "Tom was crying."

Although contexts are implicit in the sequence notation, drawn as a graph the contexts become explicit as boxes. Each concept is related to a context with a membership edge connected to its context, as Figure 1 shows.

Note that the sequence notation corresponds to a topological ordering of its graph. As a directed acyclic graph can give rise to one or more topologic orderings, the preferred ordering is one that resembles the linguistic realisation. As a consequence, a meaning-preserving translation from a sentence in one language to another language could result in a single meaning representation that would show different orders in sequence notation for the two languages. This is exemplified for a simple English sentence (1) and its translation in Dutch (2) with a different word order.

(1) a. (that) a boy bought a book.
b. `boy.n.01 buy.v.01 Agent -1 Theme +1 book.n.02`

(2) a. (dat) een jongen een boek kocht.
b. `boy.n.01 book.n.01 buy.01 Agent -2 Theme -1`

2.4 Role Inversion

A role connects two entities, but can only be hooked to one. This could cause unwanted side-effects such as cycles in the corresponding graph (see previous section) or imperfect linguistic alignment (see next section). The mechanism of *role inversion*, as introduced in description logics, AI approaches of knowledge representation and AMR, is therefore a useful one to have at one's disposal because of the added flexibility in creating meanings.

Role inversion is defined as follows: $\forall R \forall x \forall y (R(x,y) \leftrightarrow \overleftarrow{R}(y,x))$, where $\overleftarrow{R}$ is the inversion of R. In words: every role, a binary relation, has a dual, and if you want to swap the arguments of a role, you can do so using the dual without changing the overall meaning. Following the convention in AMR, I use the `Of` suffix to indicate inverted roles. Consider (3) with an inverted role and compare it to the earlier (1).

(3) a. A boy bought a book.
b. `boy.n.01 buy.v.01 Agent -1 book.n.02 ThemeOf -1`

Role inversion does not affect the truth-conditional meaning, and for checking syntactic equivalence of graphs inverted roles are normalised (Cai and Knight, 2013). Role inversion gives us flexibility in the sequence notation, which is useful in semantic annotation tasks where linguistic alignment is important.

2.5 Linguistic Alignment

For practical purposes (human annotation and verification and natural language processing technologies using machine learning) it is convenient to get a close alignment between the meaning representation and the natural language expression that it forms the interpretation of. It is hard to align meaning graphs with text, which is linear by nature (Pourdamghani et al., 2014; Liu et al., 2018; Anchiêta and Pardo, 2020; Blodgett and Schneider, 2021). I show how a reasonably fine-grained alignment can be provided using the sequence notation. (Appendix B shows an elaborated example.)

Because the sequence notation is simply a succession of hooked or anchored concepts, possibly divided by context separators, it gives us a lot of freedom in the way it can be encoded. As most writing systems in western cultures possess a left-to-right direction, it is convenient to follow this convention when describing languages following this direction, as I have done in the examples above. However, for annotation purposes a top-to-bottom organisation is handy and perhaps also the most neutral seen from the perspective of the various writing systems used for natural languages. It is also used in computational linguistics to annotate text with labels classifying word tokens in categories for tasks such as part-of-speech or named entity tagging, known as the column-based format (Buchholz and Marsi, 2006). Figure 2 gives us the idea.

```
boy.n.01                            % A boy
bought.v.01 Agent -1 Theme +3 % bought
quantity.n.01 2 QuantityOf +1 % two
box.n.03 MeasureOf +1           % boxes of
bonbon.n.01                         % bonbons.
```

Figure 2: Aligning a sequence meaning with text.

Even though there is no one-to-one mapping between words and elements of the meaning representation, the alignment is reasonably executed, with all concepts in line with a noun, adjective, or verb. Prepositions, determiners, and particles aren't always directly alignable, and nor are discontinuous expressions. The alignment could be further improved using the machinery introduced by Blodgett and Schneider (2021).

2.6 Evaluation

Evaluation of meaning representation becomes important and interesting when one wants to compare two meanings that are independently produced for the same input. This could be a comparison between computer output and gold standard annotation (curated by a semanticist), or a comparison between two human-created meanings in order to calculate inter-annotator agreement. A simple proposal using existing software is put forward in Poelman et al. (2022) who convert sequential meanings to PENMAN format (Kasper, 1989) and then use SMATCH to compute overlap of triples (Cai and Knight, 2013). Therefore no new machinery is required to evaluate meanings in sequence notation, and improved evaluation metrics such as SEMBLEU can also be adopted easily (Song and Gildea, 2019).

3 Interpreting Sequences

In the previous section I showed how sequential meanings can be constructed. In this section I explain how they are interpreted. Appendix A illustrates how sequential meanings can be converted to Discourse Representation Structures from DRT.

3.1 Concepts

A concept in a sequence has a dual purpose: it (a) introduces an entity within its context, and (b) classifies it to a particular concept. Hence, every entity has a corresponding one-place predicate, a "guard", that classifies it within some background knowledge ontology.[1] Roughly speaking, a simple sequence of concepts $[\![C_1 \ldots C_n]\!]$ corresponds to the first-order formula $\exists \mathrm{x}_1 \ldots \exists \mathrm{x}_n(C(\mathrm{x}_1)\ldots C(\mathrm{x}_n))$. In the terminology of Discourse Representation Theory (Kamp and Reyle, 1993), a concept C that is part of a context B introduces a fresh discourse referent x in the domain of B and a basic condition $C(x)$ in the set of conditions of B.

3.2 Anchors

Anchors connect a concept in a meaning representation with an external entity. It can be seen as a means of grounding or anchoring abstract units of meaning with concrete entities present in the real world. The denotation of an anchored concept is defined as follows: $[\![C\ Rc]\!] = \exists x(C(x) \wedge R(x,c))$.

3.3 Hooks

A hook connects ("hooks") a concept to another concept by a two-place relation. Recall that a hook is always (a) attached to a concept and (b) ends with an index. The indices replace the variables found in traditional meaning representation, inspired by work of Nicolaas Govert de Bruijn (1972). There are negative and positive indices. As concepts are strictly ordered in the sequential notation, we can refer to a concept by refering to the relative position the relation is situated: the index 0 refers to the current concept, -1 to the concept introduced before the current concept, -2 to the concept before that, and so on. Negative indices refer to entities introduced before, and positive indices refer to entities that are available later in the sequence: $+1$ refers to a concept that is introduced after the current index. This mechanism is crucial to understand how hooks work, and bears also resemblance with how co-reference is implemented in Lexical Functional

[1] This is reminiscent of guarded quantifiers (Andréka et al., 1998), and it is equivalent to a many-sorted first-order logic, where sorts, sometimes called types, denote subsets of the domain. Instead of assigning a sort to a variable directly, I do this by adding a one-place predicate (a concept).

Grammar (Kaplan and Bresnan, 1982). The first-order logic interpretation of a concept with hooks is thus roughly defined as follows: $[\![C\ H_1 \cdots H_n]\!] = \exists x(C(x) \wedge H_1(x, y_1) \wedge \cdots \wedge H_n(x, y_n))$. Indices without an antecedent concept correspond to free variables in first-order logic.

3.4 Separators

A separator divides a sequential meaning representation into two contexts: the context before, and the context after the separator. Hence, a sequential meaning representation with n separators has exactly $n + 1$ contexts. There are various kinds of separators. The type of separator tells us what logical or rhetorical relationship exists between the two contexts. A key application of separators is the treatment of negation, disjunction and universal quantification, but separators also find use in assigning discourse structure and rhetorical relations in text.

A separator decorated with a connector <1 means that the separator connects two local contexts. A connector <2 means that the context following the separator is attached to an earlier introduced context: not the previous context but the one just before that. Newly introduced contexts always connect to a previously introduced context. A new context cannot be linked to more than one context. Usually, a separator connects two adjacent contexts. But it is possible that a separator connects two contexts that are not adjacent. This happens with wide-scope interpretations, presuppositional accommodation, non-local discourse relations, and disjunction.

4 Semantic Phenomena

4.1 Negation and Disjunction

Negation has impact on the structure of meaning: it doesn't introduce a new conceptual entity or hook, but rather packages the information in what is asserted as positive information and what is negative. In sequence notation, negation introduces the separator `NEGATION`, stating that the negated information following the separator is attached to the context just before the separator (Figure 3). Its first-order equivalent is ∃x(person.n.01(x) ∧¬∃y∃z(book.n.02(z) ∧ buy.v.01(y) ∧ Agent(y,x) ∧ Theme(y,z))). In DRT parlance, the corresponding DRS would have a nested box with a unary negation operator (see Figure 9 in Appendix A).

```
person.n.01                      % Somebody
  NEGATION <1                    %
buy.v.01 Agent -1 Theme +1 % bought
book.n.02                        % no book.
```

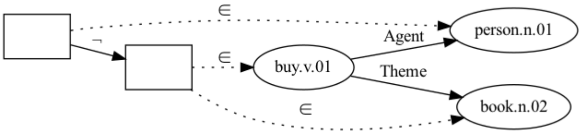


Figure 3: Graph for "Somebody bought no book."

Another example with negation is given in Figure 4, displaying a sequential meaning with three contexts, where the contextual index <2 ensures that the second negation is correctly attached to the main context, rather than the first negated context.

```
female.n.02                   % She
  NEGATION <1                 % is neither
rich.a.01 AttributeOf -1      % rich
  NEGATION <2                 % nor
famous.a.01 AttributeOf -2 % famous.
```

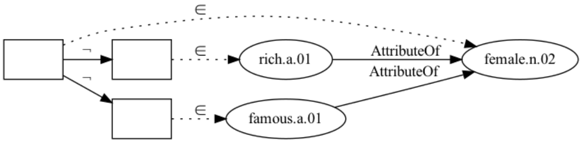


Figure 4: Graph for "She is neither rich nor famous".

Disjunction is represented in sequential meanings using the equivalence $(p_1 \vee p_2 \vee ... \vee p_n) \equiv \neg(\neg p_1 \wedge \neg p_2 \wedge ... \wedge \neg p_n)$. This representation has the advantage that no new separators are required, and that there is no limit to the number of disjuncts, as shown in Figure 5.

```
person.n.01 EQU speaker         % I
 NEGATION <1                    %
 NEGATION <1                    %
bake.v.02 Agent -1 Patient +1 % bake
bread.n.01                      % bread,
 NEGATION <2                    %
listen.v.01 Agent -3 Theme +1 % listen
music.n.01                      % to music,
 NEGATION <3                    % or
read.v.01 Agent -5 Source +1  % read comic
comic-book.n.01                 % books.
```

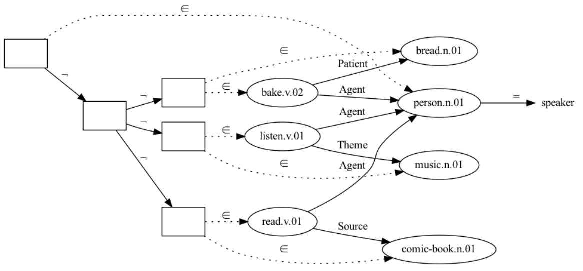


Figure 5: Graph exemplifying disjunction.

4.2 Universal Quantification

Universal quantification is encoded in sequential meanings by making use of the logical equivalence $\forall x(P(x) \rightarrow Q(x)) \equiv \neg\exists x(P(x) \wedge \neg Q(x))$. For instance, the sentence "Everyone smoked." is analysed as: it is not the case that there is a person that is not smoking. In sequence notation this would be `NEGATION <1 person.n.01 NEGATION <1 smoke.v.01 Agent -1`. The reason to use nested negation rather than a conditional is because this way there is no need to add two new separator relations—that would need to be coordinated as well, because unlike negation, a unary operator, implication and disjunction are binary operators—to the vocabulary.

Universal quantifiers in object position pose a challenge to meaning-text alignment in the sequence notation because of the scope they take over the transitive verb. An example is given in Figure 6, where the `CONJUNCTION` separator performs a merge of semantic information akin to merging of Discourse Representation Structures (Zeevat, 1991). This representational technique effectively gives the object wider scope, and is similar to presuppositional accommodation (Van der Sandt, 1992).

```
female.n.02                    % She
  NEGATION <1
  NEGATION <1
buy.v.01 Agent -1 Theme +1 % bought
  CONJUNCTION <2               % every
book.n.02                      % book.
```

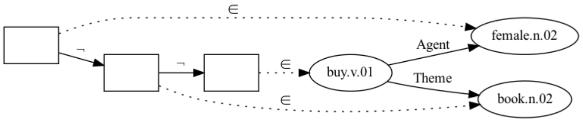


Figure 6: Graph displaying universal quantification.

4.3 Discourse Relations

Rhetorical relations are also encoded in sequential meanings by separators. Here I adopt the inventory of discourse relations as proposed in SDRT (Asher, 1993). Figure 7 shows an example where the rhetorical relation `EXPLANATION` connects two contexts. In sequential meanings discourse relations are always between single contexts. In SDRT, however, this is not necessarily the case because of the recursive nature of the segmented discourse representation structures. Yet sequential meanings can still capture rhetorical structure (Figure 8).

```
person.n.01                 % Someone
smile.v.01                  % smiles.
 EXPLANATION <1             %
male.n.01 EQU -2            % He
happy.a.01 Experiencer -1 % is happy.
```

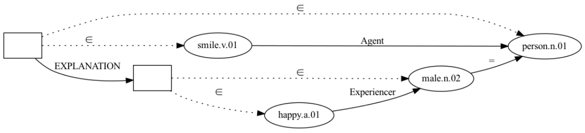


Figure 7: Graph visualisation for a short text.

As Asher and Lascarides (2003) have shown, anaphoric reference to compound discourse units is possible. The sequence notation would require additional machinery to catch this phenomenon. This could be something like a summation operation similar to handling split antecedents of plural pronouns in Discourse Representation Theory (Kamp and Reyle, 1993). This is probably also needed to cover the `CONTRAST` and `PARALLEL` discourse relations of SDRT.

```
person.n.01 Name "Max"          % Max
have.v.01   Pivot -1 Theme +2 % had
lovely.a.01 AttributeOf +1      % a lovely
evening.n.01                    % evening.
   ELABORATION <1
male.n.02   EQU -4              % He
have.v.01   Pivot -1 Theme +2 % had
great.a.01  AtttributeOf +1     % a great
meal.n.01                       % meal.
   ELABORATION <1
male.n.02   EQU -4              % He
eat.v.01  Agent -1 Patient +1 % ate
salmon.n.01                     % salmon.
   NARRATION <1
male.n.02     EQU -3              % He de-
devour.v.01 Agent -1 Patient +2 % voured
quantity.n.01 EQU +               % lots of
cheese.n.01   Quantity -1         % cheese.
   NARRATION <3
male.n.02   EU -11              % He
win.v.01    Agent -1 Theme +2 % won
dancing.n.01                    % a dancing
competition.n.01 Theme -1     % competition.
```

Figure 8: Sequential meaning for Asher and Lascarides (2003)'s celebrated example.

In SDRT, a `NARRATION` of a discourse unit U″ of U′, where U′ is an `ELABORATION` of U, would automatically invoke an `ELABORATION` relation of U″ to U, given the way SDRSs are constructed. This is not the case in sequence notation for the reason mentioned above. To capture such indirect discourse relations, some background inference rules would be needed.

5 Related Formalisms

The development of the sequence notation found inspiration from a wide spectrum of semantic representation systems, ranging from the classic semantic networks, Discourse Representation Theory, and Abstract Meaning Representations. In this section we will discuss how they are related: what do they have in common and how do they differ?

5.1 Semantic Networks

Semantic networks were introduced in the early 1970s to represent meaning (Simmons, 1973). Typically in these networks, a distinction is made between entity types and tokens (for instance, "a dog" would introduce two nodes in the network, one describing the set of dogs, and the other a particular member of that set, whereas in AMR just one node would be introduced in the semantic graph). The need for a richer network formalism was already recognised back then by Gary Hendrix, to cover linguistic phenomena such as universal quantification, hypothethical and imaginary situations. Hendrix (1975) introduced a method for partioning a semantic network into *spaces*. His use of spaces in semantic nets is strongly reminiscent to the way we employ contexts in the sequence notation, and is also similar to the Scoped Semantic Networks proposed by Power (1999).

A yet even more elaborative proposal was made around the same time by Len Schubert, who extended the expressive power of semantic nets with negation, disjunction and lambda expressions (Schubert, 1976). The resulting networks became rather cumbersome, and even Schubert himself remarks "I hasten to add that I am not urging universal adoption of this notation." These bunglesome additions might have been the reason why the extended networks never became mainstream in later years of AI and NLP, with the exception of the Conceptual Graphs proposed by Sowa (1984).

5.2 Discourse Representation Structures

One of the most elaborated semantic formalisms is probably Discourse Representation Theory (Kamp, 1984). Proposed in the early 1980s, it has seen many improvements, extensions, modifications, and reincarnations (Klein, 1987; Roberts, 1989; Zeevat, 1991; Van der Sandt, 1992; Kamp and Reyle, 1993; Asher, 1993; Reyle, 1993; Bos et al., 1994; Muskens, 1996; Van Eijck and Kamp, 1997; Frank and Kamp, 1997; Piwek, 2000; Kadmon, 2001; Beaver, 2002; Asher and Lascarides, 2003; Bos, 2003; Geurts and Maier, 2013; Kamp et al., 2011; Geurts et al., 2020). A wide range of linguistic phenomena are covered by DRT, among them conditionals, negation, modals, disjunction, presupposition, plurals, tense, aspect, and quantifier scope.

The contexts in sequence notation can be compared directly to the DRSs in Discourse Representation Theory. But sequential meanings discard representational redundancies: discourse referents are implicitly introduced by concepts. DRT has separate types of DRS conditions to model conditionals and disjunction, whereas the sequence notation only uses negation to cover these.

Standard DRT (Kamp and Reyle, 1993) follows a Davidsonian event semantics, whereas in this paper a neo-Davidsonian semantics is adopted that gives us the binary relations that enables simple graphical visualusation. Several features of DRT can be transferred to sequential meanings: blocking of anaphoric links by inaccessibility, merging of DRSs (Zeevat, 1991), and presuppositional accommodation (Van der Sandt, 1992).

5.3 Abstract Meaning Representations

The Abstract Meaning Representation formalism (Langkilde and Knight, 1998) represents meaning of natural language sentences as rooted, directed acyclic graphs. It took the clarity of the early semantic networks, and techniques introduced by AI researchers such as role inversion. Large semantically annotated corpora were developed based on AMR (Banarescu et al., 2013), encoded by using the PENMAN notation introduced by Kasper (1989). These corpora sparked a lot of interest in computational linguistics, and gave rise to many new approaches to semantic parsing and generating text from meaning representations.

Drawing a parallel with the semantic networks introduced in the 1970s, history repeats itself, when many scholars realized that AMR has incomplete inference capabalities for negation (and other logical devices such as universal quantification). Several proposals for extending AMR were published (Bos, 2016; Stabler, 2017; Pustejovsky et al., 2019; Bos, 2020; Lai et al., 2020; Stein and Donatelli, 2021). However, none of these proposals were widely adopted.

Several features of AMR are also present in the sequence notation: the binary relations that support attractive graphical visualisation, the use of role inversion, and being agnostic to grammar. But there are also notable differences: the sequence notation is closer to surface wording because there is not as much decomposition as in AMR. The sequence notation supports logical quantification and negation, which AMR lacks. And the sequence notation adopts WordNet (Fellbaum, 1998) and VerbNet (Kipper et al., 2008) to interpret the non-logical symbols, whereas AMR is based on PropBank (Palmer et al., 2005), but not all non-logical symbols are interpreted (verb-based symbols are, noun-based symbols aren't). This makes AMR partly specific to English, even though there have been AMR corpora constructed for other languages.

6 Discussion

6.1 No Overdose of Variables

Variables require some kind of naming convention, effectively an arbitrary way of blessing entities with a unique identifier. It is this resort to a naming system that makes variables unattractive for applications such as machine learning and human annotation. Usually, there are some informal conventions involved in naming variables, such as giving a variable an index that is increased by every new concept introduced in the meaning representation, or using the next letter of the alphabet. Alternatively, as is done in AMR, the variable name is based on the name of the concept that it names (Banarescu et al., 2013). This works well for short sentences, but as soon as longer texts need to be taken into account, the naming system gets cumbersome in practice.

The system of indices in sequential meaning does not suffer from these issues. Furthermore, the indices are *relative*—not absolute—capturing local "distances" between concepts. This enables a generalisation of catching argument structure, independent of sentence or text length. Even for short sentences meaning representations with indices yield better results in neural parsing than those resorting to variables (Van Noord et al., 2018). Hence, using indices rather than variables has the potential to offer advantages respect to human annotation and machine learning. And even though in this paper the sequence notation is used to encode DRS-based meanings, it can also be used to produce AMRs, as the AMR in (4) and its translation in sequence notation (5) show.

(4)
```
(w / want-01 :arg0 (b / boy)
   :arg1 (g / go-01 :arg0 b))
```

(5)
```
boy want-01 :arg0 -1 :arg1 +1
go-01 :arg0 -2
```

The sequence notation results in shorter and compact meaning representations, because no space is wasted on brackets and variables.

6.2 Compositionality

I don't say much about *compositionality* from the perspective of the syntax-semantics interface. This is a deliberate choice. Compositionality—the study of how meanings of complex expressions are derived from meanings of their parts—is a fascinating problem in formal and computational semantics (Montague, 1973; Dowty et al., 1981) in which many attempts have been formulated and implemented, in particular within the Montagovian tradition (Bos et al., 1996; Bender et al., 2015).

The assumption in any implementation of compositionality is that there are atomic units of expressions carrying meaning that cannot be further decomposed. But what these atomic units are is unclear in general, and can range from simple inflectional markers to multi-word expressions. An extreme direction in this tradition, however never been explored in computational semantics, is Natural Semantic Metalanguage, defining a small set of semantic primes of which meanings can be composed (Wierzbicka, 1996).

A theory of syntax that supports semantic theory is therefore not sufficient to completely uncover compositionality, and moreover, makes the formalism language dependent. Arguably, large semantic annotation efforts have been shipwrecked exactly on the dependence of a computational grammar (Bos et al., 2017; Abzianidze et al., 2017).

Instead, sequential meanings do not require a lexical theory of meaning, such that one could, for instance, give an interpretation for a preposition, article or adverb in isolation. It assumes the expressions that it maps meanings to are complete utterances. Giving up strong compositionality is, from one perspective, certainly attractive, as it makes the formalism language-neutral and opens the door for multi-lingual computational semantics. Having said this, there are natural ways to break down sequantial meanings into smaller pieces (concepts, hooked/anchored concepts, contexts, and so on).

7 Conclusion

The meaning represention that I proposed has much in common with AMR (Banarescu et al., 2013) and DRS (Kamp and Reyle, 1993). But there are notable differences. Like AMR but unlike DRS, sequential meanings are agnostic to any method or theory of syntax. Like AMR, but unlike DRS, sequential meanings can be viewed as simple graphs. Like DRS, but unlike AMR, there is an explicit way of assigning scope to logical operators. Unlike AMR and DRS, there are no variables in sequential meanings.

The quote "make everything as simple as possible, but not simpler", often attributed to Albert Einstein, is perhaps what summarises the sequence notation. It provides a language that I think cannot be simpler than it is, at the same time making it possible to describe complex meaning representations (including negation, disjunction, quantification, and discourse structure) with a formal interpretation. As there are only binary relations, and the binary relations can be inverted, a sequential meaning can be visualised as a directed acyclic graph, resulting in graphs that are simpler than those previously proposed for Discourse Representation Theory (Basile and Bos, 2013; Abzianidze et al., 2020). The sequence notation therefore offers a visual aid for verification of meanings.

I think the sequence notation is also a convenient way of annotating text with meaning representations. The notation is simple, no logical variables are needed, meanings can be manually entered and corrected in a standard text editor. The sequence notation supports the alignment between meaning representations and corresponding linguistic realisation in an approximate manner, where at least the order of the concepts corresponds with the order as they are introduced in the text by nouns, verbs, adjectives and adverbs. Yet I understand that not everyone is convinced that annotation with the sequence notation would be simpler than say AMR or DRS. This paper has no evidence for this claim and is solely based on personal experience. Additionally, I have observed that researchers with logic background have become accustomed to the use of variables, making it considerably challenging for them to abandon the familiarity of such notation.

Currently the sequence meaning notation has been put in practice in the Parallel Meaning Bank (Abzianidze et al., 2017). In future work the idea is to take advantage of the sequence notation and annotate larger (multi-sentence) multi-lingual documents with meaning representations that include rhetorical structure.

Acknowledgments

Previous versions of this work have been presented on various occasions, including the online ILFC seminar on 18 January 2022, the Workshop in honor of the 80th birthday of Hans Kamp (which was held from 23–25 June 2022 in Stuttgart), the Workshop on Ten Years of Babelnet and Multilingual Neurosymbolic Natural Language Understanding (Rome, 4–5 July 2022), the kick-off workshop of the MWE-SEMPRE project in Düsseldorf on 8 September 2022, and the ALiAS-'23 workshop in Amsterdam on 6 January 2023. I would like to thank the participants of all these events for the valuable feedback they provided. In particular, I am grateful to Anette Frank for pointing out the connection between LFG and the relative indices in sequential meanings (see § 3.3); to Alex Lascarides and Nicholas Asher for demonstrating the differences between SDRT and the way discourse is structured in sequentual meanings (see § 4.3); and to Nathan Schneider for showing me the differences between AMR and sequential meanings with respect to meaning-text alignment (see § 2.5).

Additionally, the comments of the three anonymous IWCS-2023 reviewers helped me to further improve this article. Reviewer #1 with the encouraging words "an interesting proposal that might go someplace" suggested to look at approaches proposed in description logic, as it reminded them of sequential meanings, and yes thanks for the suggestion, I will explore this area of research in the future. And yes I don't present a model theory directly but instead do so by translation to DRT. And no, I am not bothered by the lack of support for generalised quantifiers (see Appendix 5), I am more worried about *kinds* and subsective adjectives. Reviewer #2 "a notation which has some attractive properties" made me rewrite several parts of the paper to make the presentation clearer. And indeed large language models might make all these efforts obsolete but who knows, perhaps explainability might become fashionable. Reviewer #3 was "unconvinced that annotating in sequence notation would be easier, faster and more reliable" and yep you might be right about this and perhaps it is just a matter of taste of what one is used to. However, the expressive capacity goes well beyond the state-of-the-art in large-scale semantic annotation efforts.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 242–247, Valencia, Spain.

Lasha Abzianidze, Johan Bos, and Stephan Oepen. 2020. DRS at MRP 2020: Dressing up discourse representation structures as graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 23–32, Online. Association for Computational Linguistics.

Rafael Anchiêta and Thiago Pardo. 2020. Semantically inspired AMR alignment for the Portuguese language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1595–1600, Online. Association for Computational Linguistics.

Hajnal Andréka, István Németi, and Johan Van Benthem. 1998. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274.

Nicholas Asher. 1993. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publishers.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Studies in natural language processing. Cambridge University Press.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria.

Valerio Basile and Johan Bos. 2013. Aligning formal meaning representations with surface strings for wide-coverage text generation. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 1–9, Sofia, Bulgaria.

David I. Beaver. 2002. Presupposition Projection in DRT: A Critical Assesment. In *The Construction of Meaning*, pages 23–43. Stanford University.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London, UK. Association for Computational Linguistics.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, structure-aware algorithms for improved variety, accuracy, and coverage of AMR alignments. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3310–3321, Online. Association for Computational Linguistics.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

Claire Bonial, William Corvey, Martha Palmer, Volva V. Petukhova, and Harry Bunt. 2011. A hierarchical unification of lirics and verbnet semantic roles. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 483–489.

J. Bos. 2003. Implementing the Binding and Accommodation Theory for Anaphora Resolution and Presupposition Projection. *Computational Linguistics*, 29(2):179–210.

Johan Bos. 2016. Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535.

Johan Bos. 2020. Separating argument structure from logical structure in AMR. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 13–20, Barcelona Spain (online). Association for Computational Linguistics.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer.

Johan Bos, Björn Gambäck, Christian Lieske, Yoshiki Mori, Manfred Pinkal, and Karsten Worm. 1996. Compositional Semantics in Verbmobil. In *The 16th International Conference on Computational Linguistics*, pages 131–136, Copenhagen, Denmark.

Johan Bos, Elsbeth Mastenbroek, Scott McGlashan, Sebastian Millies, and Manfred Pinkal. 1994. A Compositional DRS-Based Formalism for NLP-Applications. In *International Workshop on Computational Semantics*. University of Tilburg, The Netherlands.

Nicolaas Govert de Bruijn. 1972. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. In *Indagationes Mathematicae (Proceedings)*, volume 75, pages 381–392. Elsevier.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Gennaro Chierchia. 1992. Anaphora and dynamic binding. *Linguistics & Philosophy*, 15:111–183.

Ann Copestake, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

David R. Dowty, Robert E. Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Studies in Linguistics and Philosophy. D. Reidel Publishing Company.

Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

Anette Frank and Hans Kamp. 1997. On context dependence in modal constructions. In *Proceedings of the 7th Semantics and Linguistic Theory Conference*, pages 151–168.

Bart Geurts, David I. Beaver, and Emar Maier. 2020. Discourse Representation Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, spring 2020 edition. Metaphysics Research Lab, Stanford University.

Bart Geurts and Emar Maier. 2013. Layered discourse representation theory. In Alessandro Capone, Franco Lo Piparo, and Marco Carapezza, editors, *Perspectives on Linguistic Pragmatics*, Perspectives in Pragmatics, Philosophy & Psychology, pages 311–327. Springer.

Jeroen Groenendijk and Martin Stokhof. 1990. Dynamic Montague Grammar. In *Papers from the Second Symposium on Logic and Language*, pages 3–48.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts.

Gary G. Hendrix. 1975. Expanding the utility of semantic networks through partitioning. In *Proceedings of IJCAI*, pages 115–121.

Nirit Kadmon. 2001. *Formal Pragmatics*. Blackwell.

Hans Kamp. 1984. A Theory of Truth and Semantic Representation. In Jeroen Groenendijk, Theo M.V. Janssen, and Martin Stokhof, editors, *Truth, Interpretation and Information*, pages 1–41. FORIS, Dordrecht – Holland/Cinnaminson – U.S.A.

Hans Kamp, Josef van Genabith, and Uwe Reyle. 2011. Discourse Representation Theory. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 15, pages 125–394. Elsevier, MIT.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA. Reprinted in Mary Dalrymple, Ronald M. Kaplan, John Maxwell, and Annie Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, 29–130. Stanford: Center for the Study of Language and Information. 1995.

Robert T. Kasper. 1989. A flexible interface for linking applications to penman's sentence generator. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 153–158, Philadelphia.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation*, 42(1):21–40.

Ewan Klein. 1987. VP Ellipsis in DR Theory. In Jeroen Groenendijk et al., editors, *Studies in Discourse Representation Theory and the Theory of Generalised Quantifiers*, volume 8, pages 161–187. FLORIS, Dordrecht.

Kenneth Lai, Lucia Donatelli, and James Pustejovsky. 2020. A continuation semantics for abstract meaning representation. In *The Second International Workshop on Designing Meaning Representations (DMR 2020)*, Barcelona, Spain.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, pages 704–710.

Yijia Liu, Wanxiang Che, Bo Zheng, Bing Qin, and Ting Liu. 2018. An AMR aligner tuned by transition-based parser. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2430, Brussels, Belgium. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.

Richard Montague. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, Dordrecht.

Reinhard Muskens. 1996. Combining Montague Semantics and Discourse Representation. *Linguistics and Philosophy*, 19:143–186.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Paul Piwek. 2000. A formal semantics for generating and editing plurals. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022. Transparent semantic parsing with Universal Dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192, Gyeongju, Republic of Korea.

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English strings with Abstract Meaning Representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Doha, Qatar. Association for Computational Linguistics.

R. Power. 1999. Controlling logical scope in text generation. In *Proceedings of the 7th European Workshop on Natural Language Generation*, Toulouse, France.

James Pustejovsky, Nianwen Xue, and Kenneth Lai. 2019. Modeling quantification and scope in abstract meaning representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33, Florence, Italy. Association for Computational Linguistics.

Uwe Reyle. 1993. Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics*, 10:123–179.

Craige Roberts. 1989. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12(6):683–721.

Rob A. Van der Sandt. 1992. Presupposition Projection as Anaphora Resolution. *Journal of Semantics*, 9:333–377.

Lenhart K. Schubert. 2015. Semantic representation. In *AAAI Conference on Artificial Intelligence*.

L.K. Schubert. 1976. Extending the expressive power of semantic networks. *Artificial Intelligence*, 7:163–198.

R.F. Simmons. 1973. Semantic networks: Their computation and use for understanding english sentences. In R. Schank and K. Colby, editors, *Computer Models of Thought and Language*, pages 63–113. W.H. Freeman & Co.

Linfeng Song and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

John F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

John F. Sowa. 1995. Syntax, semantics, and pragmatics of contexts. In *ICCS*, pages 1–15.

Ed Stabler. 2017. Reforming AMR. In *Formal Grammar 2017. Lecture Notes in Computer Science*, volume 10686, pages 72–87. Springer.

Katharina Stein and Lucia Donatelli. 2021. Representing implicit positive meaning of negated statements in AMR. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 23–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Richmond Thomason. 1974. *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, New Haven.

Jan Van Eijck and Hans Kamp. 1997. Representing discourse in context. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 179–237. Elsevier.

Rik Van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Anna Wierzbicka. 1996. *Semantics: Primes and Universals*. Oxford Universiy Press.

Hendrik Willem Zeevat. 1991. *Aspects of Discourse Semantics and Unification Grammar*. Ph.D. thesis, University of Amsterdam.

A Translation to DRS

Here I sketch a translation from sequential meaning notation to DRT's Discourse Representation Structure (DRS). Although the sequential meaning system presented here bears strong similarities with Discourse Representation Theory (Kamp and Reyle, 1993), it is significantly different from it:

1. Events are represented in a neo-Davidsonian way whereas in DRT a Davidsonian way is assumed (i.e., without adopting an inventory of thematic roles);

2. All non-logical synbols are interpreted using WordNet as supporting ontology, whereas in DRT these remain uninterpreted;

3. A single NEGATION relation is used to capture negation, disjunction and conditionals, whereas DRT has special complex conditions for them in the DRS language;

4. There is no syntactic check for free and bound variables, whereas the geometrical structure of DRS immediately shows accessibility of referents.

5. There is no support for generalised quantifiers unlike DRT that has duplex conditions to accommodate them. If one were to incorporate generalised quantifiers into sequential meanings one would likely resort to adding new separators to the inventory. For instance, for "A guitar has six strings", we would arrive at something like GENERALISATION <1 guitar.n.01 MOST < have.v.02 Pivot -1 Theme +1 string.n.03 Quantity 6. These two separators would need to be coordinated though: one cannot exist without the other.

6. There is no different in representation of singular and plural noun phrases—the model theory behind sequential meanings allows entities in the domain to range over plural noun phrases as well.

Despite these differences, the similarities with DRT become immediately clear when one sketches the translation from sequential meanings to DRS (Kamp and Reyle, 1993). Only *closed* sequential meanings can be translated to DRS, so each index needs to have an antecedent context, each connector needs to link to an existing context, and in the resulting DRS no free variables should occur.

The easiest way to explain the translation from sequential meanings to DRS is to take the corresponding rooted directed acyclic graphs as starting point. The root node is always a context. The translation to DRS starts with this context, initiated as an empty DRS. Recall that a DRS consists of a domain (a set of discourse referent) and a set of (basic and complex) DRS-conditions. All entities with concept C that are members of this context are added to the domain of the DRS with a fresh discourse referent. The concept is translated as unary predicate applied to this discourse referent and added to the conditions of the DRS. All hooks and anchors of this concept are added to the conditions as binary predicates, where the internal argument is the same as the discourse referent.

Once this is completed for all members of a context, the process is recursively repeated for contexts that are connected to the current context. There are two main cases here: (1) NEGATION adds a complex unary condition $\neg B$ to the DRS, where B will be the result of the translation of the context associated to the negation; and (2) CONJUNCTION does not start a new DRS, but instead continues adding information to the current DRS. The other separators build up a structure as in SDRT (Asher, 1993). To illustrate the procedure, I show in Figure 9 the DRSs that are the result of translating two sequential meanings presented earlier in this paper.

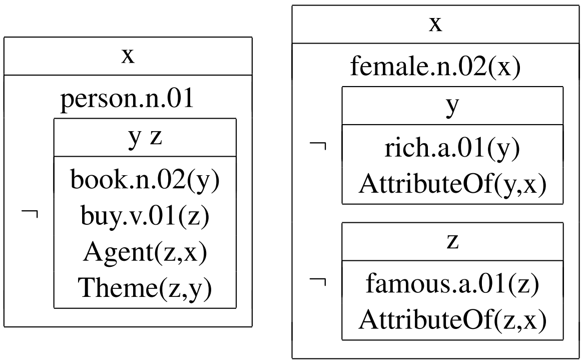


Figure 9: DRS equivalents of the sequential meanings shown in Figure 3 and Figure 4.

B Semantic Annotation Example

Figure 10 shows an elaborated example in sequence notation aligned with its textual input. Figure 11 visualises the corresponding graph.

```
male.n.02 Name "Pierre Vinken"                     % Pierre Vinken,
  APPOSITION <1
quantity.n.01 EQU 61                               % 61
measure.n.02 Quantity -1 Unit "year"               % years
old.a.01 AttributeOf -3 Value -2                   % old,
  CONJUNCTION <2
time.n.08 TSU now                                  % will
join.v.01 Theme -5 CoTheme +1 Role +3 Time -1      % join
board.n.01                                         % the board
nonexecutive.a.01                                  % as nonexecutive
director.n.02 Attribute -1                         % director
time.n.08 MonthOfYear 11 DayOfMonth 29 TOV -5      % Nov. 29.
  ELABORATION <1
male.n.02 Title "Mister" Name "Vinken" EQU -10     % Mr. Vinken
be.v.03 Theme -1 Co-Theme +2 Time +1               % is
time.n.08 EQU now
chairman.n.01 Of +1                                % chairman of
company.n.01 Name "Elsevier N.V."                  % Elsevier N.V.,
  APPOSITION <1
country.n.02 Name "The Netherlands"                % the Dutch
publishing_group.n.01 Source -1 EQU -2             % publishing group.
```

Figure 10: Meaning in sequence notation aligned for the first text of the Wall Street Journal corpus (Marcus et al., 1993). The text is here included as comments on each line following a percentage sign, and is not part of the actual meaning representation. Three different comparison operators are used here: EQU (equality), TSU (temporally succeeds), and TOV (temporally overlaps). The resulting graph is shown in Figure 11.

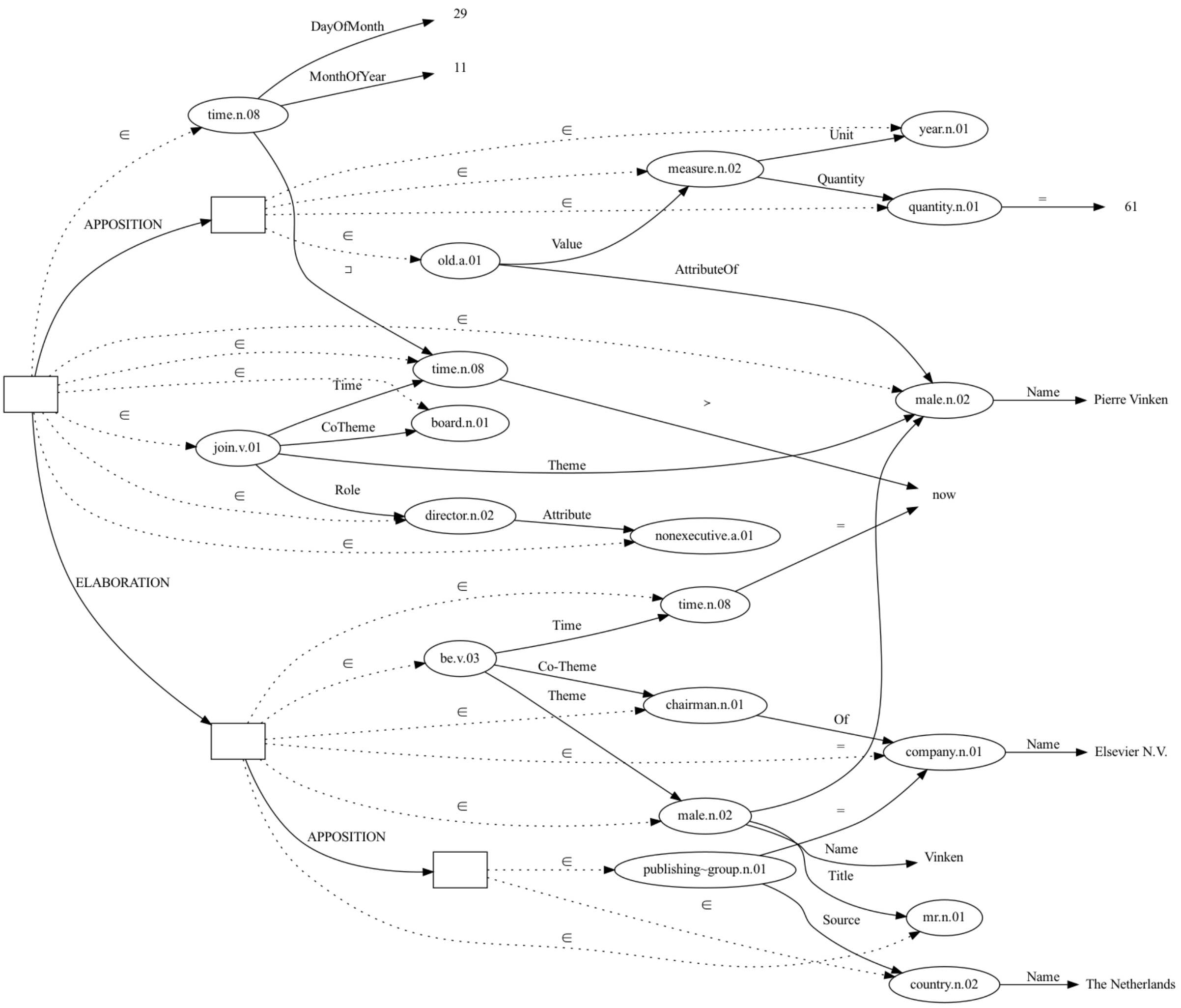


Figure 11: Graph visualisation of the WSJ corpus text “Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.”

Bridging Semantic Frameworks: mapping DRS onto AMR

Siyana Pavlova, Maxime Amblard, Bruno Guillaume
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{firstname.lastname}@loria.fr

Abstract

A number of graph-based semantic representation frameworks have emerged in recent years, but there are few parallel annotated corpora across them. We want to explore the viability of transforming graphs from one framework into another to construct parallel datasets. In this work, we consider graph rewriting from Discourse Representation Structures (Parallel Meaning Bank (PMB) variant) to Abstract Meaning Representation (AMR). We first build a gold AMR corpus of 102 sentences from the PMB. We then construct a rule base, aided by a further 95 sentences. No benchmark for this task exists, so we compare our system's output to that of state-of-the-art AMR parsers, and explore the more challenging cases. Finally, we discuss where the two frameworks diverge in encoding semantic phenomena.

1 Introduction

Many semantic representation frameworks have emerged over the years (Kamp and Reyle, 1993; Copestake et al., 2005; Banarescu et al., 2013; Abend and Rappoport, 2013), at varying levels of abstraction in terms of encoding semantic phenomena. We want to be able to compare frameworks empirically across phenomena, with the goal to understand, unify and extend them. Unfortunately, this is difficult to do in a data-driven manner as there are few freely available parallel datasets. As manual annotation is laborious, it is important to develop automatic tools to create and expand datasets. One way to approach this is by transforming annotations across frameworks. In this work, we take a look at Abstract Meaning Representation (AMR) (Banarescu et al., 2013), and Discourse Representation Structures (DRS) (Kamp and Reyle, 1993), as expressed in the Parallel Meaning Bank (PMB) (Abzianidze et al., 2017), to see how much of the former can be constructed from the latter.

We show a significant portion of AMR can be constructed from DRS and provide a discussion on our insights as to where the process is not possible. To achieve this we build a graph rewriting system from DRS to AMR. As there is no parallel data between the two, we also annotate a small part of the PMB into AMR.

Our motivation for this work is twofold. Our first goal is to get more parallel annotated data between semantic formalisms in general, and between AMR and DRS for this particular study, in order to foster empirical cross-formalism comparison. A natural question to ask here is, since (as we will see in section 5) automatic parsers based on machine learning techniques seem to perform better than rule-based transformation systems on this task, why do we bother with such an experiment. We have a few reasons: (i) rule-based transformation systems may still perform quite well, especially for more closely-related formalisms (as we show in this study) and we do not know how well exactly until we test such a system; (ii) it is possible that the two approaches make different kinds of mistakes, which opens the possibility for hybrid solutions that combine their strengths; (iii) with a rule-based system, tracking the decision-making process is possible, rendering the method explainable.

Our second goal is to better understand the differences between formalisms with a view to extend and unify them. This is difficult to do in a non-data-driven manner as the formal definitions of formalisms are rarely complete. More importantly, within the community, it is not clear what a *complete* semantic representation should consist of. Thus, while not as direct as the first, an outcome we hope to get from this work is a deeper insight into what is needed in a semantic representation and what are the missing links between formalisms, as a step towards defining a unifying framework.

The rest of the paper is structured as follows:

Proceedings of the 15th International Conference on Computational Semantics pages 209–222
June 21–23, 2023. ©2023 Association for Computational Linguistics

in section 2, we present the two frameworks; in section 3 – our graph rewriting system (GRS); section 4 is about our annotation procedure for a small gold AMR dataset; in section 5, we present our experiments, discuss the results, and compare them to those of SoTA AMR parsers; in section 6, we provide a discussion and future work directions. Our code and data are publicly available[1].

2 Background

In this section we present the Parallel Meaning Bank as an instance of a large corpus of Discourse Representation Structures, and Abstract Meaning Representation.

2.1 DRS in the PMB

The Parallel Meaning Bank (PMB) is a semantically annotated corpus, with parallel annotations available for four languages – English, German, Italian and Dutch. The portion of the PMB that contains gold annotations for English is significantly larger than the other three: 10,715 sentences vs 2,844 (German), 1,686 (Italian) and 1,467 (Dutch). The formalism behind the PMB semantic representations is Discourse Representation Theory (DRT) (Kamp and Reyle, 1993) and in particular Projective DRT (PDRT) (Venhuizen, 2015), which differs from DRT in the way it accounts for presuppositions and conventional implicatures. DRT expressions are called Discourse Representation Structures (DRS). DRS are typically represented as boxes with variables defined at the top of the box and the entities and relations between them in the bottom. The boxes are used to label scopes and discourse units. Similar to (Muskens, 1996), the PMB "dialect" of DRS is compositional and it allows to embed boxes into one another, specifying the relations between them. Sentences from the PMB can be viewed on the PMB explorer[2]. There, DRS's can be seen in three kinds of notation: the traditional box notation (Figure 1a), clause notation (Figure 1b), and the recently proposed Simplified Box Notation (SBN) (Bos, 2021) (Figure 1d).

The PMB uses WordNet (Fellbaum, 1998)[3] to encode senses (e.g. `attack.v.04`, `shark.n.01`) and VerbNet/LIRICS (Bonial et al., 2011) for semantic roles (e.g. `Agent`, `Patient`, etc.).

For the purposes of our work, as the three notations available in the PMB are equivalent[4], we use SBN as a starting point, as it is simplest to process. We transform SBN representations into graphs for easier manipulation and visualisation (Figure 1c).

2.2 AMR

Abstract Meaning Representation (AMR) represents "who did what to whom" in a sentence. It is meant to be rather abstract in order to be easily-readable by humans and easier for annotators to work with. The simplification is achieved by not encoding phenomena such as tense, plurality or scope, though this can also be seen as a disadvantage.

AMR abstracts away from the surface representation, allowing multiple sentences with the same meaning to have the same representation. The AMR in Figure 2 is the representation of the sentence "*He was attacked by a shark.*", but also of "*A shark attacked him.*". Furthermore, as AMR does not encode various semantic phenomena, sentences with similar (but not the same) meanings can also get the same representation. The AMR in Figure 2 also represents the sentences "*The shark attacked him.*" and "*Sharks will attack him.*", among others.

AMR is centered around predicate-argument structure and, for English, makes extensive use of PropBank predicates (Palmer et al., 2005). Predicates are used to annotate verbs in a sentence, but also adjectives, and sometimes even nouns, if the appropriate PropBank frames exist. Each predicate has a set of arguments which are called *core roles* and appear as numbered arguments in AMRs (see `ARG0` and `ARG1` in Figure 2). Additionally, *non-core roles* such as `time`, `domain`, `duration` make up the rest of the AMR relations.

AMRs are directed acyclic graphs (DAGs) with a single root. Respecting both of these properties does not always come naturally. To preserve both, an AMR role can be inverted by changing its direction and adding `-of` to its label. Inverse roles are also useful for highlighting the *focus* of a sentence.

Unlike for DRS, the larger, more commonly used AMR datasets, are only available via a paid license from the Linguistic Data Consortium[5]. Still, a smaller portion of the so-called AMR Bank is freely available[6], namely the Little Prince corpus and the BioAMR corpus. However, for the purposes of our work, we need parallel data between DRS and

[1] `https://gitlab.inria.fr/semagramme-public-projects/drs2amr`

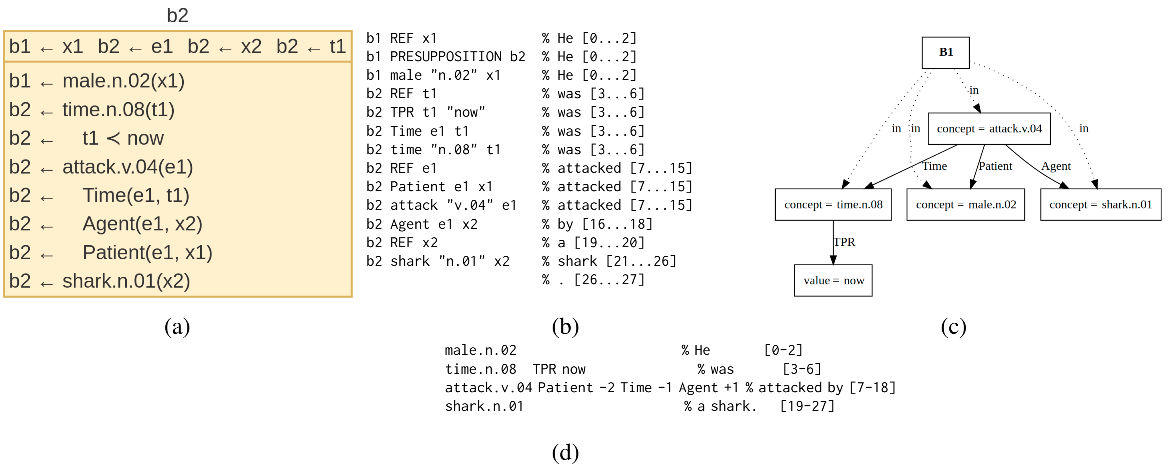
[2] `https://pmb.let.rug.nl/explorer/explore.php`; data freely available under ODC-BY 1.0

[3] `https://wordnet.princeton.edu/`

[4] with the exception of PRESUPPOSITION

[5] `https://www.ldc.upenn.edu/`

[6] `https://amr.isi.edu/download.html`



(a)

```
b1 REF x1                 % He [0...2]
b1 PRESUPPOSITION b2      % He [0...2]
b1 male "n.02" x1         % He [0...2]
b2 REF t1                 % was [3...6]
b2 TPR t1 "now"           % was [3...6]
b2 Time e1 t1             % was [3...6]
b2 time "n.08" t1         % was [3...6]
b2 REF e1                 % attacked [7...15]
b2 Patient e1 x1          % attacked [7...15]
b2 attack "v.04" e1       % attacked [7...15]
b2 Agent e1 x2            % by [16...18]
b2 REF x2                 % a [19...20]
b2 shark "n.01" x2        % shark [21...26]
                          % . [26...27]
```

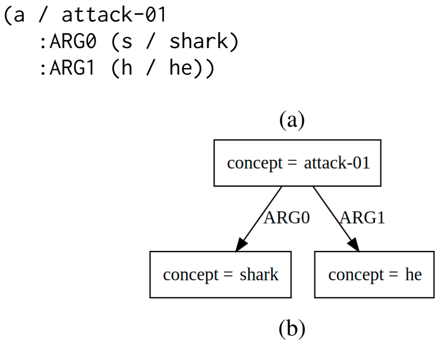
(b)

(c)

```
male.n.02                                % He        [0-2]
time.n.08   TPR now                        % was       [3-6]
attack.v.04 Patient -2 Time -1 Agent +1 % attacked by [7-18]
shark.n.01                                % a shark.   [19-27]
```

(d)

Figure 1: The sentence “He was attacked by a shark.” in box notation (a), clause notation (b), as a graph (c), and in simplified box notation (SBN) (d).



```
(a / attack-01
   :ARG0 (s / shark)
   :ARG1 (h / he))
```

(a)

(b)

Figure 2: AMR annotation of the sentence “*He was attacked by a shark.*”, among others, in (a) Penman notation and (b) as a graph.

AMR. Since that, to the best of our knowledge, does not exist, we chose to annotate a small portion of the PMB into AMR (section 4).

3 System

We use GREW[7] (Guillaume, 2021; Bonfante et al., 2018) to build a graph rewriting system (GRS) for rewriting SBN graphs into AMR ones. GREW is a tool that allows the user to define rules to match patterns in a graph and apply a set of commands that transform the matched part of the graph. GREW also allows for the use of lexicons, which lets us map sets of values and assign a value to a variable based on the value of another variable.

3.1 Lexicons

AMR and the PMB use different lexical resources, so our system relies extensively on lexicons to map them: WordNet verbs to PropBank predicates, and VerbNet semantic roles to PropBank arguments.

SemLink[8] (Palmer, 2009) is an existing effort aimed at linking English linguistic resources, among which PropBank and WordNet. We scraped the SemLink verb groupings[9] to collect mappings between the WordNet senses used in our dataset and the corresponding PropBank predicates. We note that this is a many-to-many mapping, as can be seen in the sample below.

```
wn_pred    pb_pred
%=================
try.v.01   try-01
try.v.01   try-04
play.v.01  play-01
play.v.07  play-01
```

We collected 133 such mappings in total for the 197 (95 from *dev* set + 102 from *test* set) sentences in our dataset[10]. These span across 110 WordNet senses and 119 PropBank predicates. The mappings do not cover all the predicates present in our dataset, as either the WordNet or PropBank predicate does not exist in its respective resource or the mapping between the two is not a part of SemLink. We found 22 such WordNet predicates, 13 of which correspond to phrasal verbs.

Next, for each PropBank predicate in our lexi-

[7] https://grew.fr/

[8] https://verbs.colorado.edu/semlink/

[9] https://verbs.colorado.edu/html_groupings/

[10] For this experiment, we wanted to simulate having a complete lexicon mapping. Constructing such a mapping is out of the scope of this work. Instead, we chose to have a lexicon that is “complete” at least for our dataset by pulling the predicate senses appearing in both the *dev* and *test* sets.

con, we manually[11] went over the corresponding PropBank entry and collected the VerbNet role for each argument where that was present. This way, we produced the first version of our lexicon, which we will refer to as *incomplete lexicon*. Out of the 119 PropBank predicates, 61 (around 51%) were missing a VerbNet role for some or all arguments.

Finally, we produce the final version of our lexicon, which we will refer to as *complete lexicon*. We do this by going over all the predicates again and deciding on a VerbNet role for each of the arguments that do not have one.

For example, in PropBank, for try-01, we have the following:

Arg0-PAG: *Agent/Entity Trying* (vnrole: 61.1-agent)
Arg1-PPT: *thing tried* (vnrole: 61.1-theme)

whereas for try-04, we have:

Arg0-PAG: *tryer*
Arg1-PPT: *thing tried (hand, patience)*
Arg2-PRD: *attribute of Arg 1*

As can be seen, for try-01 the corresponding VerbNet roles are explicitly specified in the brackets, whereas for try-04 they are not. Thus, while the *incomplete lexicon* contains entries for both try-01 and try-04, it specifies the PropBank numbered arguments only for try-01.

```
wn_pred  pb_pred Agent Theme ...
%===============================
try.v.01 try-01  ARG0  ARG1  ...
try.v.01 try-04  -     -     ...
```

The *complete lexicon*, on the other hand, specifies the roles for try-04 as well. As can be seen below, based on the descriptions from PropBank, we have decided to link ARG0 to Agent, ARG1 to Theme and ARG2 to Attribute.

```
wn_pred  pb_pred Agent Theme Att.
%================================
try.v.01 try-01  ARG0  ARG1  -
try.v.01 try-04  ARG0  ARG1  ARG2
```

The PMB typically uses WordNet's measure.n.02 as a node when talking about quantities. In AMR, this is more fine-grained, with concepts such as temporal-quantity or distance-quantity. Many of these can be deduced based on the :unit of said quantity, e.g. if the :unit is day, then the concept should be temporal-quantity. To address this, we also produce and use a lexicon which maps unit types to quantity types.

3.2 Our Graph Rewriting System

Our Graph Rewriting System (GRS) includes a few groups of rules, centered around different types of roles or structures in both AMR and the PMB. We selected partition *00* of our split of the PMB (see section 4 for explanation on partitions) as the set used for constructing rules, referred to hereupon as our *dev* set. AMR annotations for it were produced by annotator D (see section 4). All our data comes from the English section of the PMB.

Core roles with lexicon. This set of rules encompasses a rule for picking a PropBank predicate for the WordNet verbs in the input SBN graph if a mapping for that WordNet verb is present in our lexicon, and rules for rewriting the VerbNet roles from the input graph into PropBank numbered arguments. This category contains 27 rules – one for sense picking and one each for the 26 VerbNet predicates in our lexicon.

Core roles without lexicon. Here, we include a set of rules that rewrite the most common VerbNet roles, Agent, Patient, Theme, Stimulus and Experiencer, into PropBank numbered arguments in case they were not present in the lexicon for the relevant PropBank entry. For each, we select the most common numbered argument that that role has in our lexicon. These are later referred to as our *fallback* rules. This category contains 5 rules.

Non-core roles. This set of rules covers rewriting of PMB roles, such as Duration, Manner, Beneficiary, etc., to their AMR counterparts (:duration, :manner, :beneficiary, etc.). This category contains 21 rules.

Structures. Another set of rules deals with what we call structures. As structures, we consider a set of nodes and edges (as opposed to just a single node or a single edge) that can be rewritten into another set of nodes and edges or an individual edge. One such example is the structure used by the PMB when we have person.n.01 -EQU-> speaker. This corresponds to using either the concept I or the concept we as a single node in place of the

[11] This seems like a lot of work for a small dataset, but it is a one-off effort. Once done for the entire sense bank for a given language, it can be used for all datasets for that language.

whole structure. Here we also include rules where a single node or edge is rewritten into a set of nodes and edges. An example of this is the rule we use for named entities that rewrites the edge `Name` from the SBN graph into a structure that encompasses the `name`, `wiki` and their corresponding values in the AMR graph. We have 25 rules in this category.

Special words. A small set of rules deals with special concepts and relations. One such example is the concept `be-03` which is most often used to refer to spatial location and therefore invokes the special AMR concept `be-located-at-91`. There are 12 rules in this category.

Boxes. As described in subsection 2.1, the PMB groups nodes in boxes. When there is a single box in the SBN representation of a sentence, this generally does not bring any new information for the AMR graph. However, when more than one box is present, for example to introduce phenomena such as negation or universal quantification, this can be informative for the AMR graph as well. For example `B1 -NEGATION-> B2` can introduce a `:polarity -` relation to AMR. Our final set of rules deals with the different types of relations between boxes when more than one box is present. A final rule removes all the boxes that are left at the end. This category contains 36 rules.

The different sets of rules presented here are applied in the following order: special words, core roles with lexicon, non-core roles, structures, core roles without lexicon, boxes, except the two rules dealing with the `AttributeOf` SBN role, which are applied after boxes. An additional rule for removing cycles with three nodes by inverting one of the relations in the cycle is applied at the end.

Some of the rules are combined into non-deterministic strategies. For example, since there is no way to tell from the SBN graph only (i.e. without referring to the text) whether `person.n.01 -EQU-> speaker` refers to `I` or `we`, both versions are produced by our GRS. Similarly, as we mentioned in subsection 3.1, the WordNet to PropBank predicate mapping is many-to-many. In case a WordNet predicate maps to multiple PropBank ones, all possible graphs are produced.

3.3 Post-processing

After applying the GRS to our data, we do some post-processing on the GREW graphs. For named entities, our GRS only produces an `:op1` property for the `name` of the entity even if the name consists of multiple words. This is addressed in the post-processing step by adding `:op2` to `:opN` accordingly. Additionally, for any remaining WordNet concepts (be it verbs, nouns or adjectives) we remove the trailing part starting from the first dot, i.e. `piano.n.01` becomes `piano`. Finally we produce the PENMAN notation for the output AMR graph (or graphs in the case of non-determinism).

4 Gold Data

To evaluate our system, we produced gold AMR annotations for 102 sentences of the English part of the PMB. In order to make sure that there were no specific phenomena concentrated in certain partition of the PMB data, instead of picking a random partition and risking having a non-representative sample, we applied an algorithm to "randomise" that[12]. We created 100 new partitions, by finding the sum of the part and document number of each sentence and applying modulo 100 to get a new partition number. This approach groups the data randomly, but is reproducible and as the PMB expands, the partitions should grow in a fairly uniform manner. Version 4.0.0 of the PMB contais 10,715 gold English sentences, so 107 sentences on average per partition.

We picked partition 25 (i.e. all the documents for which $(p + d)\%100$ is 25) to annotate manually. It contains 102 sentences. Our four annotators – **A**, **B**, **C** and **D** – annotated half of the sentences (51) each. Every sentence was annotated by two annotators. To ensure that each pair of annotators had the same number of overlapping sentences, we split the 102 sentences into six groups of 17 and distributed the groups among the six different pairings.

The annotators consulted the following resources during the annotation process:

- AMR Specifications[13] as the primary source for examples and explanations on how to annotate different phenomena

[12] As per (Abzianidze et al., 2017), the corpora used in the PMB are balanced across parts. It is difficult to verify whether the dataset is also balanced across various semantic phenomena. In case it is, our randomization step is not necessary. We do, however, want to point out that the English gold data in the PMB 4.0.0 is not distributed uniformly across parts. Those towards the beginning and those with a number divisible by 10 have significantly more sentences than the rest (see subsection C.1 in the Appendix).

[13] `https://github.com/amrisi/amr-guidelines/blob/master/amr.md` (points to version 1.2.6 of the specifications at the time of writing)

- AMR Annotation Dictionary[14] for additional annotation examples grouped by specific roles, concepts, words and constructions

- PropBank Searchable Frame Files[15] for PropBank predicates and their argument structures

- A full list of PropBank frames from the AMR website[16] to find "hidden" AMR frames (e.g. "strong-02" is hidden in `strengthen.html` in the Searchable Frame Files). PropBank Searchable Frame Files took precedence in case of conflict.

- GREW-MATCH[17] to search for examples of different concepts or structures, in graph format. For AMR, GREW-MATCH currently contains all the examples from the AMR Specifications, AMR Annotation Dictionary, The Little Prince corpus, and the BioAMR corpus.

We used Smatch (Cai and Knight, 2013) to compute the inter-annotator agreement (IAA). Smatch uses a hill-climbing algorithm to find the maximum number of triples between two graphs. There are three types of triples: instance, relation, and attribute. Instance triples match nodes in the graph, counting exact matches between the node concepts. Relation triples match edges in the graph. Attribute triples match properties of the nodes. Each type has equal weight in the overall score count.

The results of our IAA are reported in Table 1. Annotator **A** appears to have the lowest agreement with the other three annotators. One reason for this may be that annotator **A** correctly observed that named entities in AMR always get a `:wiki` property, even if they do not have an existing Wikipedia page[18] and added them accordingly. The other three annotators only added a `:wiki` property to Wikipedia named entities. We have adopted annotator **A**'s approach for the gold data.

To produce the final version of the gold data, the four annotators gathered in groups (two, three, or four) over the course of a few sessions. For each sentence, the two existing annotations were compared and after a discussion, one was chosen or a modification that combines elements of both annotations was selected. In a small number of cases, the annotators agreed on an entirely different annotation from the two proposed ones.

	A	**B**	**C**	**D**
A	–	0.76	0.82	0.81
B	0.76	–	0.83	0.86
C	0.82	0.83	–	0.84
D	0.81	0.86	0.84	–

Table 1: Inter-annotator agreement – Smatch f-score.

5 Evaluation

As with our IAA, we use Smatch to evaluate our system's output against the gold annotations. As mentioned in section 4, Smatch takes into account not only the graph structure, but also the exact match of concepts between graphs. Thus, we expected that the predicate lexicon and the second post-processing step (removing trailing part of WordNet concepts) would have a substantial impact on the final score. To evaluate this, we run the experiment with *no lexicon* (1), with the *incomplete lexicon* (2), and with the *complete lexicon* (3). Additionally, we also run a version with the *complete lexicon*, but without the second post-processing step (4). Finally, while adding a lexicon of senses increases the results for both the *test* and *dev* sets significantly, we see that the difference in results between the *incomplete lexicon* and the *complete lexicon* setting is very small. Our hypothesis is that this is due to the fallback rules for core roles that have not been rewritten. To verify this, we run the three different lexicon versions (*no lexicon* (5), *incomplete lexicon* (6), and *complete lexicon* (7)) also without the fallback rules. We run each of these experiments on both the *dev* and *test* sets.

The results from our experiments are reported in Table 2. As can be seen our hypothesis about the benefit of a lexicon and the post-processing step is justified: we get an increase of $6-7\%$ on both the *dev* and *test* sets for all scores. When we consider the fallback rules, we can compare experiments (1), (2) and (3) with experiments (5), (6) and (7). We see that the fallback rules do a lot of the groundwork, but more so when we have no lexicon or an incomplete one. They have less of an impact when working with a complete lexicon.

Due to non-deterministic rules, for some sen-

[14] https://www.isi.edu/~ulf/amr/lib/amr-dict.html

[15] http://verbs.colorado.edu/propbank/framesets-english-aliases/

[16] https://amr.isi.edu/doc/propbank-amr-frames-arg-descr.txt

[17] http://semantics.grew.fr/

[18] Indeed, we observe that all the named entities in the AMR annotated data, except from one sentence from the BioAMR corpus have a `:wiki` property.

tences we get more than one output graph. As we want to see what is the biggest part of the AMR structure that can be built from DRS, the scores we report take the graph with highest overlap (according to Smatch F1-score) with the gold graph[19].

Comparing the scores on the *dev* and *test* sets, we notice the big disparity in the scores between the two. This is due to our building the rules based on the *dev* set and thus missing out on structures that do not appear in it, but appear in the *test* set. This suggests that our rule set is incomplete and a larger *dev* set may be necessary to ensure broader structure coverage. One such example is that the structure used in the PMB for expressions such as *instead of* and *rather than* needs special treatment which requires either the duplication of a specific node or the introduction of the predicate *prefer-01*. However, since our rule base was built from the *dev* set and that does not include such an example, we do not have a rule to address it. We do, however, have such an example in the *test* set and it cannot be addressed properly.

5.1 Comparison to AMR parsers

As far as we are aware, transforming DRS graphs into AMR ones is a new task. There is, therefore, no benchmark against which we can compare our outputs. For the sake of argument, however, we got predictions from two SoTA AMR parsers – an ensemble one, and a single-model one.

MBSE. Maximum Bayes Smatch Ensemble (MBSE) (Lee et al., 2022) is an ensemble distillation model that combines knowledge from a number of models to produce a single prediction. MBSE is currently the SoTA AMR parser.

AMRBART. AMRBART (Bai et al., 2022) uses graph-to-graph pre-training to improve pre-trained language models' awareness of the graph structure of AMRs. It is currently the best single-model and fifth best overall parser on the AMR2.0 and AMR3.0 datasets.

We sent our dataset to the MBSE authors and obtained the predictions from the Ensemble-5 MBSE model back from them. As for AMRBART, we ran the fine-tuned on AMR parsing AMRBART-large (AMR2.0)[20] on our dataset. The granular Smatch scores[21] (Damonte et al., 2017) for these two as well as for our system on both our *test* and *dev* sets are in Table 3. MBSE predictions have not been wikified. We expect that after wikification, MBSE's score will be on par with AMRBART's.

As can be seen in the table, the AMR parsers perform better overall compared to our system. We believe there are two main reasons for this. Firstly, the AMR parsers have been trained on a lot more data: tens of thousands of sentences versus 95 for our system. Secondly, we are limited by the information that is present in the DRS and parts of the AMR structure simply cannot be predicted from it (see section 6 for further discussion).

A closer look at the granular scores indicates that the areas where our system performs particularly poorly is when dealing with negations and reentrancies, both of which are the hardest areas for the parsers as well. For negation, we owe this to the fact that in DRS, when negation is morphological, but there is a corresponding WordNet concept, as is the case with `unhealthy.a.01` in "I knew it was unhealthy" (*26/2674*), this is expressed in one node, whereas in AMR, we have a node for `healthy-01` and a node that negates that[22].

In some cases, our system performs better than the parsers. For example, in sentences that use comparison (e.g. "This car is bigger than that one" (*67/2333*)). However, this is likely because our rule for handling these cases was built following the AMR guidelines, as was our gold dataset. The data that the two parsers have been trained on uses a different than in the guidelines structure, leading them to learn that instead. To their credit, our hypothesis is that if they were trained on the same structure, they would be more likely to predict it correctly.

5.2 Error Analysis

As discussed earlier, some of our errors are due to certain structures not being present in our *dev* set. These do not, however, account for the errors on the *dev* set itself. There are a number of other aspects which come at play here.

Missing predicates from lexicon. A number of the predicates in our sentences, while present in both WordNet and PropBank, do not appear in Semlink. Therefore we have not been able to add them to our lexicon. This leads to a non-overlap

[19] For completeness, if we take the worst graph instead, the F1-score for experiment (3) is 0.71 for *dev* and 0.65 for *test*.

[20] `https://github.com/goodbai-nlp/AMRBART`

[21] `https://github.com/mdtux89/amr-evaluation`

[22] It would be possible to address this via a rule that captures such words. However, only words where the negative particle is indeed a morpheme, need to follow this rule (it would not apply to *"uniform"*, for example). This would require the construction of a lexicon of words with negative morphemes. This is ultimately a task that requires morphological analysis and, as such, is out of the scope of this work.

	Dev set			Test set		
	Precision	Recall	F1-score	Precision	Recall	F1-score
(1) No lexicon	0.72	0.71	0.72	0.66	0.62	0.64
(2) Incomplete lexicon	0.79	0.77	0.78	0.72	0.68	0.70
(3) Complete lexicon	**0.79**	**0.78**	**0.78**	**0.73**	**0.68**	**0.70**
(4) Complete lexicon, no concept fix	0.65	0.63	0.64	0.61	0.57	0.69
(5) No lexicon, no fallback	0.61	0.60	0.60	0.55	0.51	0.53
(6) Incomplete lexicon, no fallback	0.75	0.74	0.75	0.69	0.65	0.67
(7) Complete lexicon, no fallback	0.77	0.75	0.76	0.70	0.66	0.68
MBSE – no wiki	0.84	0.83	0.83	0.85	0.80	0.82
AMRBART	**0.85**	**0.83**	**0.84**	**0.86**	**0.86**	**0.86**

Table 2: Smatch scores. Where "no fallback" is not specified, it means that the fallback rules have been applied. Where "no concept fix" is not specified, it means that the post-processing concept addition has been applied.

		Smatch	Unlabeled	No WSD	Concepts	NE	Neg.	Wiki	Reent.	SRL
Dev	MBSE	0.83	0.87	0.84	0.88	0.92	0.55	–	0.66	0.84
	AMRBART	0.86	0.89	0.87	0.87	0.94	0.70	0.85	0.66	0.83
	Our system	0.78	0.84	0.79	0.78	0.91	0.40	0.68	0.53	0.75
Test	MBSE	0.82	0.86	0.83	0.86	0.94	0.60	–	0.65	0.81
	AMRBART	0.84	0.87	0.84	0.86	0.94	0.55	0.88	0.59	0.84
	Our system	0.70	0.78	0.71	0.70	0.82	0.48	0.73	0.37	0.65

Table 3: Granular Smatch scores.

between instance nodes for those predicates as well as a wrong argument structure. This is especially true in the case of adjectives since many are PropBank predicates. However, there are no adjectives in the Semlink groupings so we have not been able to add them to our lexicon.

Divergence between AMR and DRS. AMR and DRS differ in the way in which they encode certain semantic phenomena, notably scope. There are specific AMR structures for which it is not possible to decide on the correct structure, given only the DRS. We discuss some of these cases in more detail in section 6.

Inconsistencies in the PMB data. Finally, while a much smaller number, some errors are propagated from wrong annotations in the PMB dataset. An example of this can be seen in subsection C.2.

6 Discussion

Our goal with this work was to see what portion of AMR can be constructed from DRS and where that is not possible, to understand why. While constructing our rule base, we observed that the way the two frameworks encode predicate-argument structure is very similar, differing mostly in semantic role labels, where DRS relies on VerbNet roles and AMR on PropBank predicates. With an exhaustive lexicon that contains a mapping between all senses and their arguments in the two lexical resources, it will be possible to rewrite these correctly.

The most notable difference between the two frameworks is the lack of scope in AMR, whereas that is present in DRS. Some phenomena linked to scope are encoded differently in the two. E.g., universal quantification is typically encoded in the PMB in the same way as generics: the sentences "All the seats are booked." (*50/2764*) and "A cat has two ears." (*60/0913*) have a similar structure, despite the different phenomena. In AMR the two are encoded differently as the quantifier "all" is present on the surface in one and not in the other. Similarly, the quantifiers "the" and "this" are expressed in the same way in DRS: by neither being present in the representation, while in AMR "this" is expressed as a separate node and "the" is not.

DRS, as the name suggests, is centered around discourse (as opposed to dialogue) and is not meant to encode questions very well. We observe that in the PMB, wh-questions can be derived from the representation. However, this is not the case for yes-no questions, which, in the PMB are encoded exactly as their declarative counterparts. This is not the case in AMR, thus preventing us from dis-

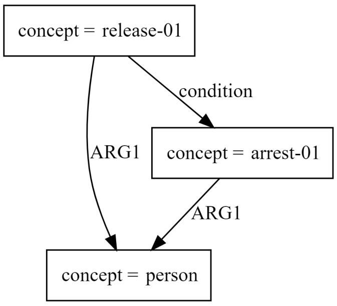


Figure 3: AMR annotation of the sentence "All who were arrested have been released." *(99/1243)*, the way it would look like if we were to follow the "logical" reading as in DRS.

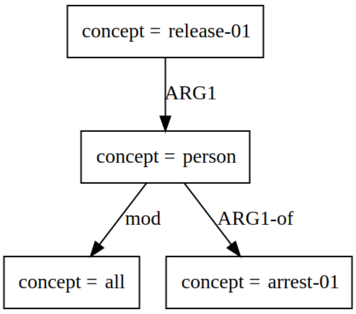


Figure 4: AMR annotation of the sentence "All who were arrested have been released." *(99/1243)*, the way a human annotator using the AMR guidelines and datasets as examples would likely annotate it.

tinguishing between the two without referring to the text of the sentence.

That being said, for a number of our rules in the **Boxes** category, we do make use of GREW's ability to check for specific regular expressions in the original sentence. This works for short sentences where there is a small or no risk of having a specific structure appear more than once. However, it is not a valid solution for longer texts. A future version of this system may benefit from using the SBN notations comments (part after % in Figure 1d).

Finally, we want to discuss a broader issue in relation to universal quantification in DRS. In the PMB, sentences such as "All who were arrested have been released." (*99/1243*) have a structure which corresponds to the reading "if a person has been arrested, they have been released". This is the way to express the semantics of universal quantification in logic. It is achieved in the PMB by using a combination of a `CONDITION-CONSEQUENCE` box embedding. This can be rewritten into AMR by making use of the non-core role `condition`, obtaining the graph in Figure 3. This is a correct reading of the sentence. However, if an annotator was to follow the AMR guidelines, we would get the graph in Figure 4. We believe this is also a correct representation of the sentence. While logically the two may be equivalent, the graph representations are structurally different. This raises the question of whether we can have more than one correct AMR per sentence. If so, then this opens the door for future considerations on how to take that into account in our evaluation metrics.

There are a number of other improvements to our system that are worth exploring in the future. Expanding the rule base can happen in two main ways (1) by expanding the *dev* set so that more varying structures are present and (2) thoroughly going though the different expressions in the AMR guidelines and AMR dictionary and designing rules for each of them. Ideally, a combination of the two should be considered. Furthermore, as we have seen with our experiments in section 5, having a lexicon that maps WordNet senses to PropBank predicates improves the score significantly. Our lexicon is still incomplete and can be further improved by adding adjectives, for instance. It would also be interesting to explore how our system performs on other languages (see Appendix A).

Our effort in trying to transform frameworks is not unique for the semantic representations community. In an exploration to better understand what linguistic semantic phenomena formalisms encode, Hershcovich et al. (2020) propose a rule-based conversion system from syntax and lexical semantics into Universal Conceptual Cognitive Annotation. Closer to our work in terms of formalisms used, Bos (2020) proposes AMR+ (an AMR extension to deal with scope) and a formal procedure to convert AMR+ into DRS. As a future work, we are interested in seeing how much AMRs obtained by applying the reverse procedure (from DRS to AMR+), then dropping the scope information, would differ from what we obtained with our system.

7 Conclusion

The goal of this work was to build a graph rewriting system from DRS (as in the PMB) to AMR to discover what portion of the latter can be constructed from the former. To do so, we first constructed a small AMR dataset from PMB sentences and built a lexicon mapping WordNet senses to PropBank predicates and arguments. We showed a significant part of the AMR structure is contained in DRS. Finally, we discussed their divergences.

Acknowledgements

We would like to thank: the anonymous reviewers for their feedback and suggestions, we have tried to incorporate all their remarks in the current version of the paper; the MBSE team, and Ramón Astudillo in particular, for kindly running MBSE on our dataset and providing us with the predictions; Guy Perrier for being our fourth annotator; Priyansh Trivedi for helping us run AMRBART. Part of this work has been funded by *Agence Nationale de la Recherche* (ANR, fr: National Research Agency), grant number ANR-20-THIA-0010-01.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. Wiley Online Library.

Claire Bonial, William Corvey, Martha Palmer, Volha V. Petukhova, and Harry Bunt. 2011. A hierarchical unification of lirics and verbnet semantic roles. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 483–489. IEEE.

Johan Bos. 2020. Separating argument structure from logical structure in AMR. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 13–20, Barcelona Spain (online). Association for Computational Linguistics.

Johan Bos. 2021. Variable-free discourse representation structures.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.

Daniel Hershcovich, Nathan Schneider, Dotan Dvir, Jakob Prange, Miryam de Lhoneux, and Omri Abend. 2020. Comparison by conversion: Reverse-engineering UCCA from syntax and lexical semantics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2947–2966, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Dordrecht. Kluwer.

Young-Suk Lee, Ramón Astudillo, Hoang Thanh Lam, Tahira Naseem, Radu Florian, and Salim Roukos. 2022. Maximum Bayes Smatch ensemble distillation for AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5379–5392, Seattle, United States. Association for Computational Linguistics.

Reinhard Muskens. 1996. Combining montague semantics and discourse representation. *Linguistics and philosophy*, pages 143–186.

Martha Palmer. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*, pages 9–15. GenLex-09, Pisa, Italy.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Noortje Venhuizen. 2015. *Projection in Discourse: A data-driven formal semantic analysis*. Ph.D. thesis, Rijksuniversiteit Groningen.

A Limitations

There are a number of limitations of our work that we address in this section.

We work with semantics, and it can be argued that the meaning representation of a sentence should be identical regardless of the language. However, empirical experiments are necessary to verify that this is indeed the case when we work with real-world data and that our system still works for languages which are structurally very different from English.

That being said, reproducing this experiment for another language is not as straightforward as simply running our system on a dataset in a different language. For our system we rely heavily on lexical resources in English. The same are not as well-developed for most other languages.

Furthermore, as there is no parallel data between DRS and AMR, to run an evaluation on such a system for another language, requires the construction of a corpus in one or both frameworks. This comes at the cost of either training or having access to a skilled annotator who is also a speaker of the language for which the system is to be constructed.

Finally, relating to subsection 3.1, the missing VerbNet arguments for the PropBank predicates were decided on by one of the authors, after carefully reading descriptions for each numbered argument of the given predicate in PropBank. However, as none of the authors is an expert in semantic role labeling, we have to note that the decisions may not have always been what an expert in this field may have chosen.

B Ethical considerations

Our system is entirely rule-based: it does not rely on heavy computational power and takes a few seconds to run on a standard computer.

Our code and data are freely available and it is not necessary to obtain any paid resources to be able to reproduce our experiments.

C PMB data

C.1 Source distribution for English gold

Figure 5 shows that the sources where data comes from in gold English section of the PMB 4.0.0 is balanced across parts. The total number of sentences per part, however, is not evenly distributed, with parts towards the beginning and those with a sequence number divisible by 10 having more sentences than the rest.

C.2 Inconsistencies in PMB data

Though not very frequent, there are errors in the PMB annotations, which, in turn, propagate to the AMR annotations produced by our system. One such example is for the sentence “Since I didn’t receive a reply, I wrote to her again” (*75/3043*). Its PMB annotation, in graph format, can be seen in Figure 6. This is incorrect, as this is the DRS for the sentence “I didn’t receive a reply because I wrote to her”. For the correct version of this sentence, the `NEGATION` and `EXPLANATION` labels have to be reversed, like they are in Figure 7 for the sentence “I am hungry because I did not eat lunch” (*86/1591*).

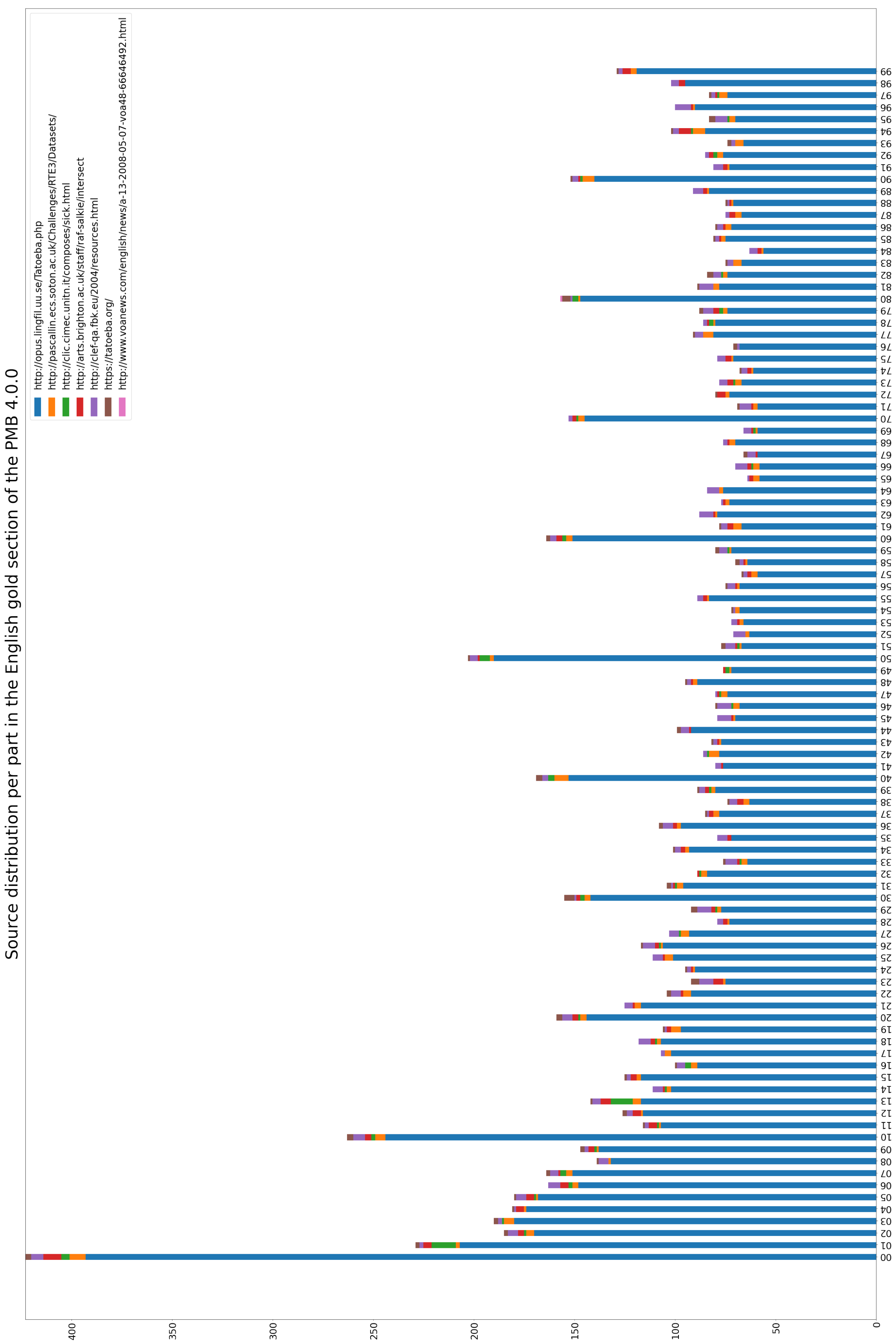


Figure 5: Distribution of the sources across the English gold part of the PMB, release 4.0.0.

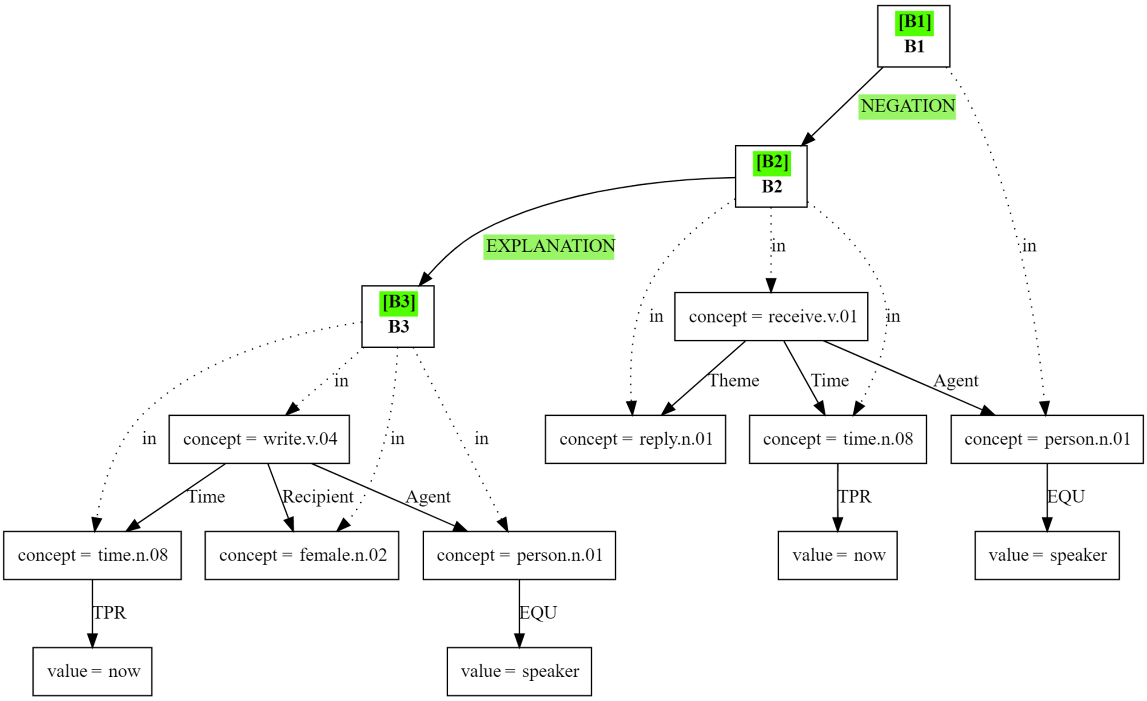


Figure 6: PMB annotation of the sentence "Since I didn't receive a reply, I wrote to her again." *(75/3043)*. The `NEGATION` and `EXPLANATION` labels should be reversed.

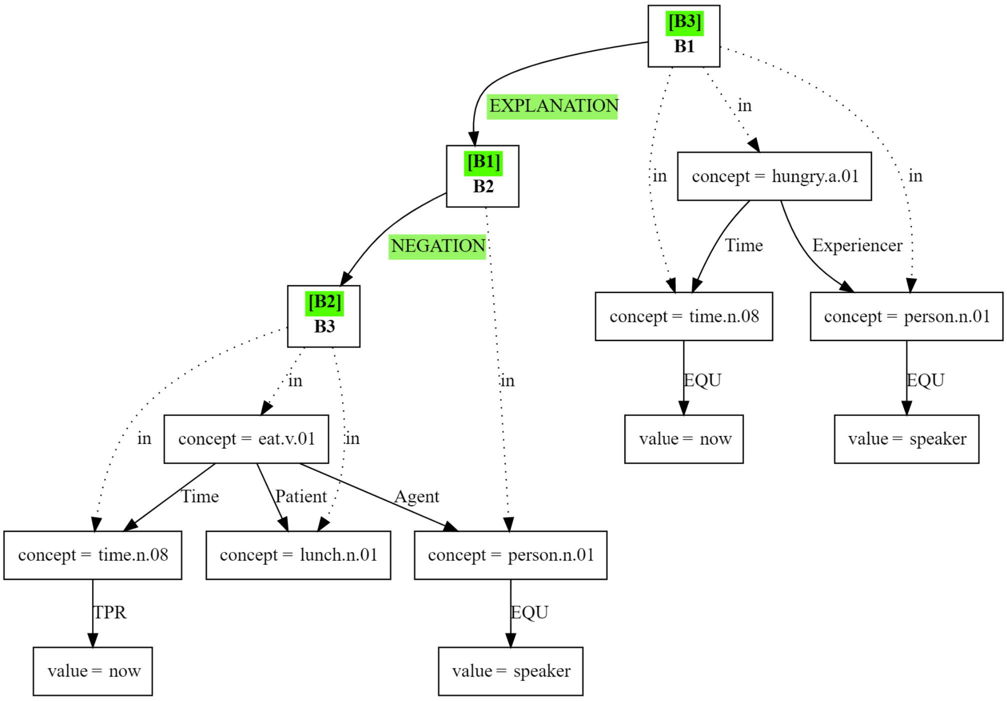


Figure 7: PMB annotation of the sentence "I am hungry because I did not eat lunch." *(86/1591)*.

Data-Driven Frame-Semantic Parsing with Tree Wrapping Grammar

Tatiana Bladier **Laura Kallmeyer** **Kilian Evang**
Heinrich Heine University Düsseldorf, Germany
first.last@hhu.de

Abstract

We describe the first experimental results for data-driven semantic parsing with Tree Rewriting Grammars (TRGs) and semantic frames. While several theoretical papers previously discussed approaches for modeling frame semantics in the context of TRGs, this is the first data-driven implementation of such a parser.[1] We experiment with Tree Wrapping Grammar (TWG), a grammar formalism closely related to Tree Adjoining Grammar (TAG), developed for formalizing the typologically inspired linguistic theory of Role and Reference Grammar (RRG). We use a transformer-based multi-task architecture to predict semantic supertags which are then decoded into RRG trees augmented with semantic feature structures. We present experiments for sentences in different genres for English data. We also discuss our compositional semantic analyses using TWG for several linguistic phenomena.

1 Introduction

While many user-facing applications of Natural Language Processing such as machine translation or sentiment analysis can these days be performed with state-of-the-art accuracy by syntax-agnostic machine learning models, grammar-based methods are still important. For one thing, they offer more transparency and insight into the decisions of a model, while in many cases having near-state-of-the-art performance (Xia et al., 2019; Kasai et al., 2019; Lindemann et al., 2019; Poelman et al., 2022). Secondly, they tend to be less data-hungry and therefore more readily adapted or transferred to low-resource languages. Symbolic methods for semantic parsing can also greatly contribute to grammar theory studies and to linguistic investigations of different languages.

In this paper, we are interested in developing a methodology for deep semantic parsing (i.e., producing semantic representations for entire sentences) which would also allow easy transfer to different languages, including low-resource ones. We start from the typologically oriented linguistic theory of Role and Reference Grammar (RRG). This theory uses a common inventory of labels and structures to describe languages from different language families (Van Valin and Foley, 1980; Van Valin, 2005). The formalization of RRG using Tree Wrapping Grammar (TWG; Kallmeyer et al., 2013) has paved the way for using this theory in computational linguistics and for developing NLP applications such as syntactic parsers (Bladier et al., 2022; Evang et al., 2022).

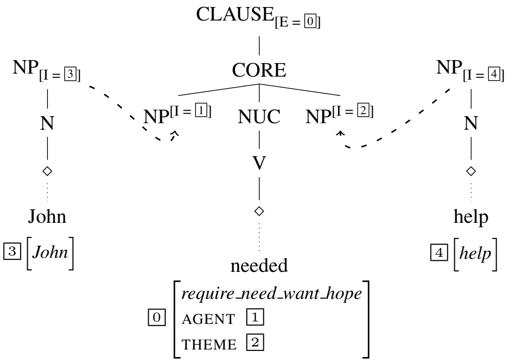


Figure 1: Frame-semantic derivation with TWG for *John needed help*

The TWG formalism is inspired by Tree-Adjoining Grammar (TAG; Joshi and Schabes, 1997) and allows for adequate modeling of long-distance dependencies. Since TWG is closely related to TAG, we can readily apply existing computational methods developed for TAG. In this work, we explore how well the methodology for compositional semantics with a tree-based syntax outlined in several theoretical papers on TAG (Kallmeyer and Osswald, 2012a,b; Zinova and

[1] The code for our semantic parser can be found on https://github.com/TaniaBladier/Frame-Semantic-Parser-with-Lexicalized-Grammars

Proceedings of the 15th International Conference on Computational Semantics pages 223–232
June 21–23, 2023. ©2023 Association for Computational Linguistics

Kallmeyer, 2012) is suitable for TWG and can be used for a large scale implementation.

A small-scale frame-semantic parser based on the Tree Adjoining Grammar was already implemented by Arps and Petitjean (2018). Our approach differs from theirs in that it is data-driven and aims for a broad-coverage semantic parser. Our method is based on transformers and contextual embeddings and we do not use a metagrammar in our application, but go for an approach based on supertagging. Our work also differs from Semantic Role Labeling (i.e., shallow semantic parsing) with TAG (Liu and Sarkar, 2009; Kasai et al., 2019) since we are interested in deep semantic representations of the sentences. Figure 1 shows how the semantic representations for the sentence *John needed help* can be produced compositionally with elementary trees in TWG paired with frames, and Figure 3 shows the frame representation for this sentence.

The objective of this paper is to implement a broad-coverage semantic parser based on Tree Rewriting Grammars. Since this is the first broad-coverage implementation of a deep semantic parser for either TAG or TWG, we are particularly interested in modeling linguistic phenomena which we came across during this data-driven implementation. We describe this in §2. We also want to investigate if our syntax-aware methodology allows us to achieve state-of-the-art results on semantic parsing. We describe the theoretical background of our work and introduce our approach to frame-based semantics with TWG in §3 and present experimental results in §4. We discuss future work in §5.

2 Semantic Parsing with TWG

2.1 Tree Wrapping Grammar

TWGs consist of elementary trees which can be combined using the operations of a) *substitution* (replacing a leaf node with a tree), b) *sister adjunction* (adding a new daughter to an internal node), and c) *tree-wrapping substitution* (adding a tree with a d(ominance)-edge by substituting the lower part of the d-edge for a leaf node and merging the upper node of the d-edge with the root of the target tree, see Fig. 2). The latter is used to capture long distance dependencies (LDDs), see the wh-movement in Fig. 2. Here, the left tree with the d-edge (depicted as a dashed edge) gets split; the lower part fills a substitution slot while the upper part merges with the root of the target tree. TWG is more powerful than TAG (Kallmeyer, 2016). The reason is that a) TWG allows for more than one wrapping substitution stretching across specific nodes in the derived tree and b) the two target nodes of a wrapping substitution (the substitution node and the root node) do not have to come from the same elementary tree, which makes wrapping non-local compared to adjunction in TAG.

TWG emerged as a result of the formalization of Role and Reference Grammar (RRG; Van Valin and LaPolla, 1997; Van Valin, 2005). RRG is a linguistic theory strongly inspired by typological concerns. RRG was used to describe languages with diverse syntactic structures such as Lakhota, Tagalog, and Dyirbal. RRG's syntactic structures are rather flat in order to be applicable to all types of different languages. According to RRG, sentence structure is organized in layers: nucleus (containing the predicate), core (containing the nucleus and the arguments of the predicate) and clause (the core and extracted arguments). Each layer can have modifiers (called periphery elements), and operators attach to the layer over which they take semantic scope.

2.2 Frame Semantics and TWG

We adapt the syntax-semantics interface for LTAG proposed by Kallmeyer and Osswald (2013) to semantic parsing with TWG. Kallmeyer and Osswald represent semantic frames as base-labelled, typed feature structures. The frames can be understood as a straightforward representation of the semantic and conceptual knowledge about a situation, while having good computational properties as their composition relies on the unification of attribute-value structures. The frames represent genuine semantic representations, and not logical expressions, whose meaning has to be derived during semantic composition[2].

The elementary trees in a lexicalized TWG are paired with frames via interface feature structures, as shown in Figure 1. Here, the root of the elementary tree for 'needed' is augmented with an interface feature structure whose E (event) attribute value is a frame of type $require\_need\_want\_hope$, which has two attributes: an agent and a theme.

[2] The advantage of the unification is that the order of semantic argument filling is not specified by successive lambda abstraction or the like. Instead, semantic argument slots can be filled in any order (in particular, independently of surface word order) via unifications triggered by syntactic composition). For a more detailed discussion see Kallmeyer and Romero (2004) and Kallmeyer and Osswald (2014)

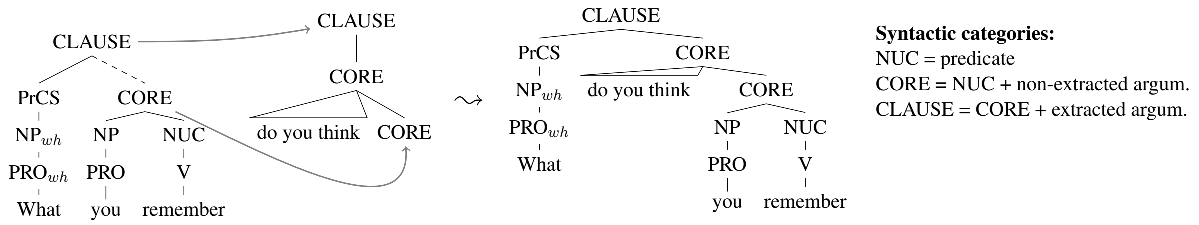


Figure 2: Tree-wrapping substitution for the sentence *"What do you think you remember"* with long-distance wh-movement.

$$\boxed{0}\begin{bmatrix} \textit{require\_need\_want\_hope} \\ \text{AGENT } \textit{John} \\ \text{THEME } \textit{help} \end{bmatrix}$$

Figure 3: Frame-semantic representation for *John needed help*.

The values of these attributes are shared with the feature structures paired with the NP substitution nodes for the subject and the object, where they are the values of the I (individual) attribute[3]. The roots of the elementary trees for 'John' and 'help' are augmented with feature structures for whose I attribute values are feature structures for whose types we use the respective lemmas (more detailed semantic representations of NPs are beyond the scope of this paper).

During parsing, as syntactic trees are combined (by adjunction, substitution or wrapping substitution), the semantic representations are also combined. The unification of interface feature structures triggers unification of feature values in the frames. In our example, as the substitution of the subject NP takes place (combining the elementary trees of 'needed' and 'John'), the respective values associated to the attribute I in the interface feature structures are unified. This results in the unification of the feature structures $\boxed{3}$ and $\boxed{1}$, which makes the frame for John become the agent of the event 'needed'. The same happens when the tree for 'help' is substituted at the object NP node of the 'needed' tree: $\boxed{4}$ and $\boxed{2}$ unify to let the frame for 'help' become the value of the theme attribute in the frame $\boxed{0}$.

To build our frame lexicon, we use the inventory of the lexical-semantic resource VerbAtlas (Di Fabio et al., 2019). VerbAtlas covers over 13 700 verbal WordNet (Fellbaum, 2000) senses, but organizes them into a relatively small number of frames (466) with only 25 cross-frame semantic roles, which makes it well suited for training neural language models. The frames in VerbAtlas are mapped to PropBank (Palmer et al., 2005) framesets and multilingual BabelNet (Navigli and Ponzetto, 2010) frames, and can potentially be linked to FrameNet (Baker et al., 1998; Baker, 2014) frames.

2.3 Complex linguistic cases

In the process of developing our data-driven semantic parser, we came across several complex linguistic constructions which were not previously described in papers dealing with the combination of Tree Rewriting formalisms and semantics. Depending on the syntactic complexity of the sentences, such constructions occur in about 20% of all sentences in our data, distributed unevenly among the subcorpora we used for the experiments. We describe some of our semantic modeling choices in this section[4].

Control constructions We introduce the variable *pivot* for cases in which an elementary tree does not have an explicit syntactic argument, but shares the argument with an elementary tree it combines with. Figs. 4 and 5 show an example. The *pivot* variable is only assigned to CORE nodes and is used to propagate the semantic representation of the controlled argument.

Constructions with a peripheral subordinate clause The representation of discourse relations is beyond the scope of this work, so for now we generate semantic representations for such clauses separately. Fig. 6 shows the elementary tree-frame pairs and Fig. 7 shows a representation for the sentence *The sheep follow him because they know his voice*.

Constructions with a non-peripheral subordinate clause If a subordinate clause is not a modi-

[3] The feature I is used as a variable in untyped frames referring to an argument (possibly syntactically complex) which fills the substitution slot.

[4] For the sake of space we only represent the relevant elementary trees in the figures of this section and skip some initial elementary trees that are substituted or adjoined into the larger trees.

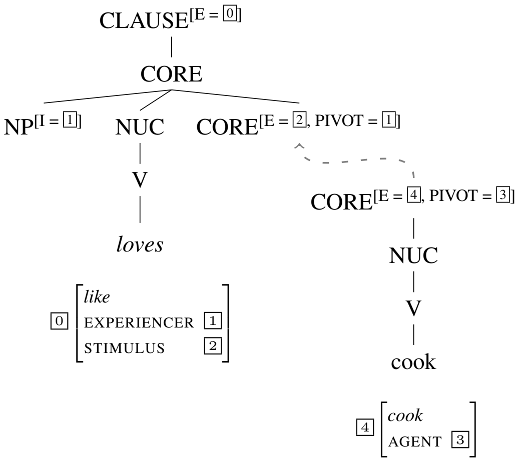


Figure 4: The pivot variable in semantic representation of the sentence *She loves to cook.*

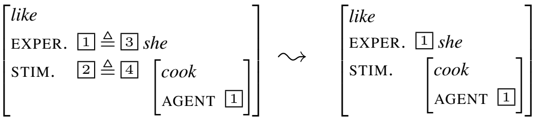


Figure 5: Label unifications and resulting frame for *she loves to cook.*

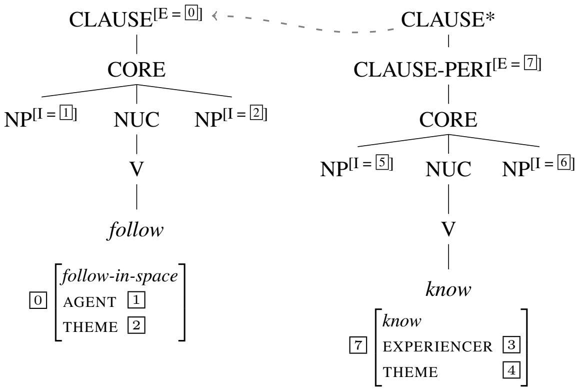


Figure 6: Tree-frame pairs for the sentence *The sheep follow him because they know his voice*

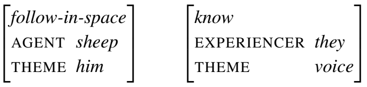


Figure 7: Semantic representations of a main clause and a peripheral subordinate clause in sentence *The sheep follow him, because they know his voice*

fier, but an argument of a main clause, the frame of the subordinate clause fills the corresponding argument slot of the parent frame (see the elementary trees and frame representation in Fig. 8, 9 for the sentence *What people say about themselves means nothing*).

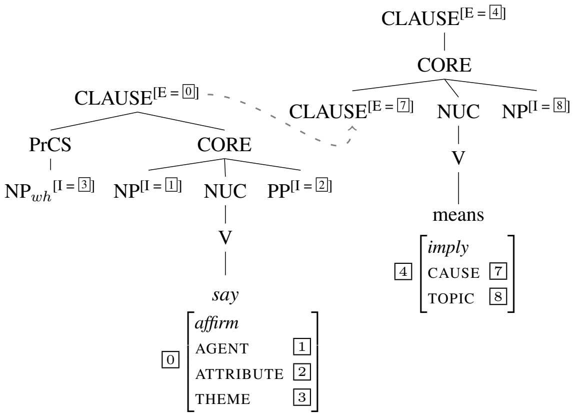


Figure 8: Tree-frame pairs for constructions with subordinate clauses

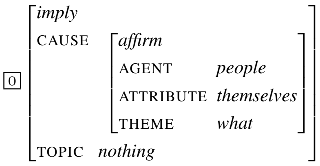


Figure 9: Constructions with subordinate clauses, here *What people say about themselves means nothing*

Treatment of prepositional phrases The treatment of prepositional phrases depends on whether the PP is an argument or an adjunct of the predicate. In (1-a) below, the PP fills a core role of the predicate *lowered*. However, the role filler *well* for this argument slot should itself be substituted first into the elementary tree of the preposition *into*. Thus, to propagate the filler of the *destination* role to the designated argument slot of *lowered*, we check during the substitution of the PP subtree and the subsequent frame unification that the argument role of the PP corresponds to the required argument role of the sentential predicate (see Fig. 10). If the prepositional phrase is an adjunct of the predicate (as, for example, in (1-b), where *with a check* modifies the predicate *pay*), the subframe of the prepositional phrase is added as an additional semantic role of the predicate after adjoining the PP subtree.

Since we focus on verbal predicates in this work, we do not explore an explicit frame representation of different prepositions, as outlined in Kallmeyer and Osswald (2013). Instead, we leave the representation of prepositions and other non-verbal predicates for future work.

(1) a. Tom lowered the bucket into the well.
b. I want to pay with a check.

Constructions with non-local dependencies Constructions with non-local dependencies (e.g.

long-distance wh-movement or extraposed relative clauses) can be handled via unification during wrapping substitution (see tree-frame pairs in Fig. 11 and the resulting representation in Fig. 12).

	Supertag	Frame	Arg. Link.
she	(NP (PRO ◇))	(entity)	(–)
loves	(CL (CO (NP) (NUC (V ◇)) (CORE)))	(like)	((1, 'Exp.'), (2, 'Stim.'))
to	(CO* (CLM ◇))		(–)
cook	(CO (NUC (V ◇)))	(cook)	((0, 'Agent'))

Table 1: Example of the training data, CL stands for Clause, CO means Core.

3 Method

3.1 Argument linking

As outlined in the previous section, our approach to semantic parsing requires two components which are used to compositionally produce a deep semantic representation of the sentences: TWG elementary trees and the corresponding semantic frames. We divide prediction of semantic frames into two subtasks: prediction of the correct frame and learning the argument linking within those frames.

The argument linking mechanism relies on the elementary tree of the predicate and predicts which substitution slot of the supertag carries which semantic role. For example, in Table 1 the argument linking for the predicate *likes* means that the first substitution slot of the corresponding supertag should get the role label "Experiencer" and the second slot gets the label "Stimulus", hence the numbers 1 and 2. In case an elementary tree has a semantic role with no local filler, as in control or raising constructions (see Figure 4) or in sentences with conjoining predicates, we mark the semantic role with the index 0, indicating that there is no substitution slot for this role (see, for example the frame *cook* in Table 1). For non-predicative frames we learn the frame with the dummy type ENTITY and resolve the type of the frame to the corresponding lemma after parsing.

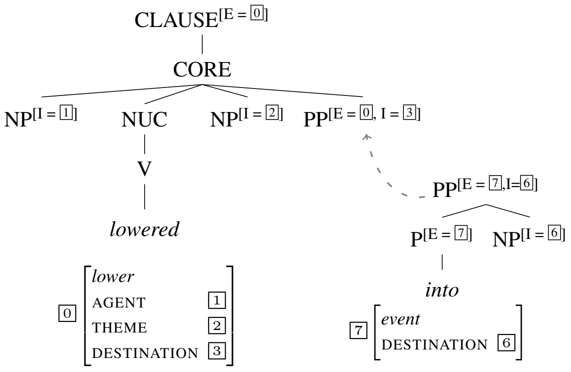


Figure 10: Propagating the role of the argument PP *into* to the main frame *lower* for the example (1-a)

3.2 Reducing the size of TWG grammars

Since the TWG grammars are usually large and contain several thousands distinct elementary trees, which is potentially hard for a neural model to learn, we reduce the size of the grammar by flattening the elementary trees and thus simplifying the syntactic structure of the trees from which we induce the TWG grammar. We collapse the internal structure of the trees, so that it preserves the relevant syntactic information about the lexical anchor and its argument structure. In particular, we delete the internal nodes of the tree which are not relevant for syntactic composition (i.e. the nodes are not involved in any tree combination operations) while leaving the root node and unlexicalized leaves untouched. We delete all SENTENCE nodes while keeping however the spine of CLAUSE, CORE and NUC since these are important targets for modifier and operator adjunctions. Figure 13 shows an example. After flattening the trees, we extract a TWG elementary trees using the automated grammar extraction approach of Bladier et al. (2020a). Since the syntactic trees in TWG grammars can have crossing branches, but the algorithm for TWG parsing (Bladier et al., 2020b), which we use to obtain syntactic representations for our data, does not support crossing branches, some nodes in trees have to be reattached before grammar extraction and re-attached to the correct nodes after parsing.

3.3 Multi-task transformer-based learning

We use the MaChAmp toolkit (van der Goot et al., 2021) to build a multi-task neural model for simultaneous learning of the elementary tree templates (i.e. supertags), frame selection, and argument linking, all cast as sequence labeling tasks. The MaChAmp multi-task models share a BERT-based encoder, but use task-specific decoders for the subtasks. Table 1 shows an example of the input for the multi-task neural model. We initially experimented with training a single-task model for each subtask and tried out different combinations of multi-task models. Since the results of a multi-task model turned out to be comparable with the single-task models (showing only around 0.1 percent of difference), we therefore carry out our ex-

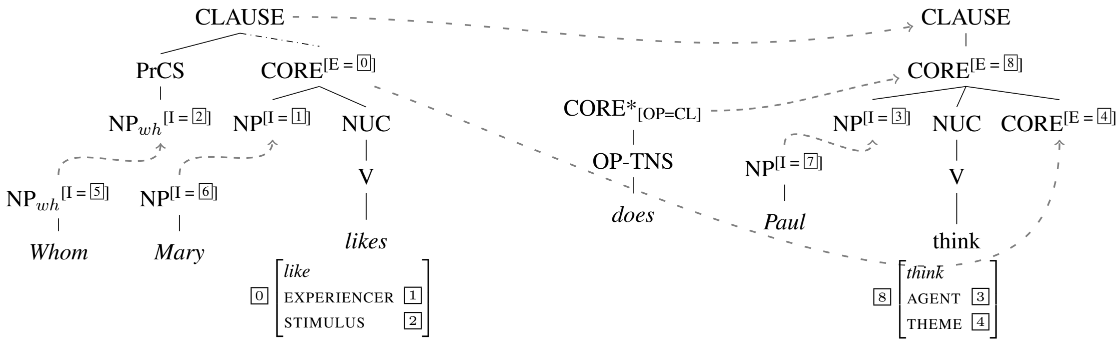


Figure 11: Wrapping substitution for wh-LDD in sentence *Whom does Paul think Mary likes?* The OP=CL notion means that the node will be attached to the CLAUSE node of the parent tree after the parsing step.

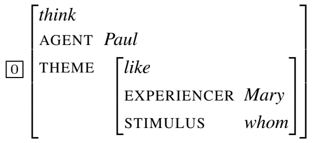


Figure 12: Semantic representation for an LDD construction in *Whom does Paul think Mary likes?*

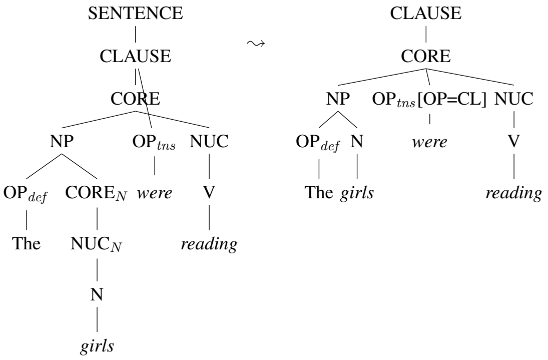


Figure 13: Example of a transformed tree before grammar extraction: the crossing branch from the original tree (on the left) is reattached and some of the internal nodes are removed. OP=CL indicates that the OP$_{tns}$ node was originally immediately below CLAUSE.

periments with the multi-task model. This model has the advantage of predicting all the components of our semantic parsing approach at once, resulting in lower training and prediction times. We tried to apply different weights on the loss function of each subtask to see if it affects the performance of the multi-task model, however the results did not change significantly. Apart from experimenting with different loss functions, we used the default values of the MaChAmp Bert model for training. The model is trained for 10 epochs, and we select the model with the highest F1-checkpoint for the evaluation.

4 Experiments and Discussion

4.1 Data

Since there is currently no manually annotated gold dataset for semantic parsing with TWG, we use alternative resources to train our model. We use the statistical neural TWG parser ParTAGe (Bladier et al., 2020b) developed for syntactic parsing with TWGs and train it on multilingual data from RRGparbank, the first large resource for TWG and Role and Reference Grammar (Bladier et al., 2022). The ParTAGe parser predicts the syntactic trees based on predicted n-best supertags for each sentence and also predicts the dependency heads based on the produced syntactic tree. The performance of this parser is different for sentences with different sentence length, but is sufficiently high for shorter sentences. We measured the ParTAGe performance on English sentences from the RRGparbank corpus (since the parser was originally trained on this data). We found that the performance of the parser on sentences with less then 7 tokens had the labeled F1 score of 93.52 for the produced syntactic trees, and the labeled F1 score of longer sentences was around 85.26.

We use the Parallel Meaning Bank v3.0.0 (PMB; Abzianidze et al., 2017) and the CoNLL-2012 English dataset based on OntoNotes 5.0 (Pradhan et al., 2012) for the frame-semantic parsing experiments. The PMB provides deep semantic representations of sentences following Discourse Representation Theory. It has rather short sentences (around 6.7 tokens on average) consisting of Web texts, newspaper articles and the Bible. The English part of the CoNLL-2012 corpus is a large resource which includes over 94 000 sentences from different genres, including journal articles, web data, broadcast news and phone conversations. We use the pre-defined train, development and test sets for both resources (see Table 2).

	PMB	OntoNotes
# sents (train, dev, test)	6654, 886, 902	75187, 9480, 9260
avg. sent length	6.94	16.71
# tokens	54205	201300
# lemmas	5463	10975
# dist. frames	350	436
# dist. frame/lemma pairs	949	2965
# frame occurrences	4783	34930
# role occurrences	13495	45496
# supertags	782	4158
# supertags occ. once	354	2204

Table 2: Statistics on the used data.

PMB and OntoNotes are not explicitly annotated with VerbAtlas frames, but PMB provides WordNet senses and VerbNet semantic roles, and OntoNotes is annotated with PropBank framesets and semantic roles. Since VerbAtlas provides manually created mappings to these resources, we used these mappings to create a sufficient amount of semantically annotated data. In order to obtain syntactic representations needed for our frame-semantic parser, we parse all sentences with the pretrained ParTAGe models available from Bladier et al. (2022).

4.2 Frame-semantic parsing experiments

Our frame-semantic parser predicts supertags needed to produce syntactic trees in parallel with the frame labels and corresponding semantic roles. We predict only heads of the semantic roles, since the full spans can be reconstructed deterministically from the predicted syntactic trees. We use the constituent trees produced by our parser to reconstruct the full spans of semantic roles[5].

VerbAtlas has 466 frames, 350 of which we observe in PMB and 436 in the OntoNotes data. The distribution of the frames is relatively even, without any frames occurring particularly more frequent then other frames. We do not consider frames associated with modal verbs. Since some of the frames occur only in test or development set and thus cannot be learned, we calculate the upper bound for the data to determine what would be the highest possible achievable score. The evaluations show a long tail of prediction errors without particular errors occurring more often then the others. Table 4 shows some of the most frequent mistakes.

The distribution of the supertags is uneven with a couple of most frequent ones occurring in the majority of the cases. We found 225 distinct predicative supertags in the PMB data, and 1358 in OntoNotes. Table 5 shows that the first three most common predicative supertags make up around two thirds of all predicates in PMB. A similar distribution is also present in the larger OntoNotes corpus, although the frequency of the most common supertags is less prominent.

The results of the frame-semantic parsing show that we achieve results comparable with the baseline Semantic Role Labeling (SRL) results on the OntoNotes and show a slight improvement on the PMB data (see Table 3[6]). The results on different genres in OntoNotes show a significant increase in performance on the Bible data and the worst results for the web texts. This result is due to the greater sentence length for the web data and a high amount of internet slang and deviations from standard English orthography and syntax.

4.3 Error analysis

Although VerbAtlas has rather coarse-grained frame lexicon, the number of frames (466) is still large and some frame pairs have only a subtle difference in its definition (e.g. the frame pairs GO-FORWARD and LEAVE_DEPART_RUN-AWAY or AFFIRM and SPEAK). Also there are some verbs, like for example *go*, which are polysemous and can be assigned different frames which appear more or less frequent in the annotated data. Since the majority of the frames appear only a couple of times in the training data, the model sometimes predicts the wrong frame which appears more frequently, as for example the frame LEAVE_DEPART_RUN-AWAY is wrongly predicted instead of CONTINUE in example (2).

(2) [...] but they're determined to keep going$_{\text{[leave\_depart\_run-away]}}$

Each frame in VerbAtlas comes with its own set of semantic roles. Although the number of the roles is small (26), the model has to learn the correct labels for each of the 466 frames. Since for most frames in VerbAtlas, the agentive and patientive role have the labels AGENT and THEME, the

[5] We reconstructed full spans of semantic roles only for OntoNotes, since the data from PMB are not annotated with full-span semantic roles.

[6] We use the following terms while describing our semantic parsing experiments: the term *trigger* stands for a lexical unit that can evoke a frame, the term *role* for frame element, and *role candidate* for the sequence of words that instantiates a role.

	PMB	OntoNotes					
		avg.	bn+bc	nw+mz	pt	tc	wb
frame trigger detection	93.75	92.92	92.35	92.14	96.41	94.79	91.56
frame label selection (w. *entity* and *event* labels)	89.75	89.57	88.56	88.65	95.81	92.5	86.15
frame label selection (only VA-labels)	83.9	89.06	87.93	87.78	97.11	92.06	85.48
*upper bound	99.81	99.46	99.59	99.38	99.71	99.65	98.88
role candidate detection	91.1	87.47**	86.54**	87.91**	91.45**	86.45**	86.25**
role label selection (head)	**86.15**	89.67**	88.36**	90.08**	93.16**	89.56**	88.15**
role label selection (full span)	–	**88.34****	87.61**	88.63**	92.11**	88.82**	86.43**
role label selection	85.8	**92.1**					
(baseline, head)	Bladier et al. (2021)	InVeRo-XL (Conia et al., 2021)					
role label selection	–	86.8					
(baseline, full span)		InVeRo-XL (Conia et al., 2021)					
avg. sent. length	5.99	14.73	14.36	20.09	11.02	8.04	16.71
# sents	902	9260	2968	2568	1051	1618	1055

Table 3: Frame-semantic parsing results. We use the frame inventory from VerbAtlas (VA; Di Fabio et al., 2019) in our semantic representations. The role label selection for full spans is not evaluated for the PMB experiment, since only semantic heads of role spans are annotated in gold PMB data. *Since some labels from the test set are not present in the training data, we measure the highest possible upper bound for the VA-label selection. **We measure the scores for OntoNotes only for pre-identified predicates to make the evaluations comparable with the reported baseline. bn+bc = broadcast, nw+mz = newswire, pt = bible, tc = telephone conversations, wb = web.

Gold frame	Predicted frame	%
GO-FORWARD	LEAVE_DEPART_RUN-AWAY	0.7
CONTINUE	LEAVE_DEPART_RUN-AWAY	0.48
INCITE_INDUCE	EXIST-WITH-FEATURE	0.42
KNOW	MEET	0.42
RESULT_CONSEQUENCE	ARRIVE	0.42

Table 4: Most frequent frame label prediction mistakes with the percentage from the overall frame label prediction errors, measured on OntoNotes data.

Supertag	% (PMB)	% (ON)
(CL (CO (NP) (NUC (V ⋄)) (NP)))	38.82	8.5
(CL (CO (NP) (NUC (V ⋄))))	14.37	6.64
(CL (CO (NP) (NUC (V ⋄)) (PP)))	10.62	3.3
(CL (CO (NP) (NUC (V ⋄)) (NP) (NP)))	7.6	0.1
(CL (CO (NP) (NUC (V ⋄)) (P) (NP)))	5.28	0.01

Table 5: Most common predicative supertags for PMB and OntoNotes (ON) data.

model frequently picks these two labels instead of some less frequent frame-specific role labels. For example in (3), the correct role set for the COME-AFTER_FOLLOW-IN-TIME frame is THEME and CO-THEME, but the model predicts the more common AGENT and THEME role labels.

(3) That$_{\text{[agent]}}$ follows$_{\text{[come-after\_follow-in-time]}}$ a decline$_{\text{[theme]}}$ in the prior six months [. . .]

As for the errors in prediction of argument linking, the most errors emerge when an infinitive modifies a noun or an adjective (see an example in (4)). The supertag for the verb in such constructions has the type of an auxiliary tree and thus lacks the agentive argument slot. In these cases, the semantic role corresponding to the PIVOT variable sometimes is not predicted (we described the PIVOT in greater detail in Section 2.3). For example, in (4) for the MANAGE frame, only the role THEME is predicted, but not the AGENT role for *strategy*.

(4) A time-honored strategy to control$_{\text{[manage]}}$ the masses$_{\text{[theme]}}$.

5 Conclusion and Future Work

In this paper, we presented the first broad-coverage frame-semantic parser with Tree Wrapping Grammar, a grammar formalism closely related to Tree Adjoining Grammar. To develop our parser, we adapted the theoretical approach of Kallmeyer and Osswald (2013) to semantic parsing with TAG and transferred it to TWG. We explored parsing strategies for several complex linguistic constructions. We developed our transformer-based language model based on the VerbAtlas frame lexicon, and experimented with English data in several genres. We could see that our semantic parser shows results close to the state-of-the-art semantic parsers.

In future work we want to explore the transferability of our approach to different languages, including low-resource ones. Our approach to semantic parsing starts from statistical syntactic parsing for TWG proposed by Waszczuk (2017); Bladier et al. (2020b). A recent work by Evang et al. (2022) presents a modification of this method for cross-lingual syntactic parsing based on word embeddings and English glosses. The underlying idea is to transfer supertag information from an English translation to the target sentence via word alignments. We plan to extend this method to semantics.

The frame lexicon VerbAtlas, which we use as a frame inventory for the semantic representations, lacks relations between frames. In order to enable semantic inference and logical reasoning with our parser, we currently investigate possibilities to develop a rule-based mapping from VerbAtlas frames to FrameNet frames, which would then yield also hierarchical relations between frames.

Acknowledgments

We thank three anonymous reviewers for their advice and useful comments. This work was carried out as a part of the research project TreeGraSP[7] funded by a Consolidator Grant of the European Research Council (ERC).

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

David Arps and Simon Petitjean. 2018. A Parser for LTAG and Frame Semantics. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Collin F. Baker. 2014. FrameNet: A knowledge base for natural language processing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 1–5, Baltimore, MD, USA. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Tatiana Bladier, Kilian Evang, Valeria Generalova, Zahra Ghane, Laura Kallmeyer, Robin Möllemann, Natalia Moors, Rainer Osswald, and Simon Petitjean. 2022. RRGparbank: A parallel role and reference grammar treebank. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4833–4841, Marseille, France. European Language Resources Association.

Tatiana Bladier, Laura Kallmeyer, Rainer Osswald, and Jakub Waszczuk. 2020a. Automatic extraction of tree-wrapping grammars for multiple languages. In *Proceedings of the 19th Workshop on Treebanks and Linguistic Theories*, pages 55–61.

Tatiana Bladier, Gosse Minnema, Rik van Noord, and Kilian Evang. 2021. Improving DRS Parsing with Separately Predicted Semantic Roles. In *Workshop on Computing Semantics with Types, Frames and Related Structures: Workshop at ESSLLI 2021*. ESSLLI.

Tatiana Bladier, Jakub Waszczuk, and Laura Kallmeyer. 2020b. Statistical parsing of tree wrapping grammars. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6759–6766.

Simone Conia, Riccardo Orlando, Fabrizio Brignone, Francesco Cecconi, and Roberto Navigli. 2021. Invero-xl: Making cross-lingual semantic role labeling accessible with intelligible verbs and roles. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–328.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. Verbatlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637.

Kilian Evang, Laura Kallmeyer, Jakub Waszczuk, Kilu von Prince, Tatiana Bladier, and Simon Petitjean. 2022. Improving low-resource RRG parsing with cross-lingual self-training. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4360–4371, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Christiane D. Fellbaum. 2000. WordNet: an electronic lexical database. *Language*, 76:706.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th*

[7] https://treegrasp.phil.hhu.de

Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 176–197, Online. Association for Computational Linguistics.

Aravind K Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of formal languages*, pages 69–123. Springer.

Laura Kallmeyer. 2016. On the mild context-sensitivity of k-Tree Wrapping Grammar. In *Formal Grammar: 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Italy, August 2016, Proceedings*, number 9804 in Lecture Notes in Computer Science, pages 77–93, Berlin. Springer.

Laura Kallmeyer and Rainer Osswald. 2012a. An analysis of directed motion expressions with lexicalized tree adjoining grammars and frame semantics. In *Logic, Language, Information and Computation. 19th International Workshop, WoLLIC 2012*, number LNCS 7456 in Lecture Notes in Computer Science, pages 34–55, Buenos Aires, Argentina. Springer. Proceedings.

Laura Kallmeyer and Rainer Osswald. 2012b. A frame-based semantics of the dative alternation in lexicalized tree adjoining grammars. *Empirical Issues in Syntax and Semantics*, 9:167–184.

Laura Kallmeyer and Rainer Osswald. 2013. Syntax-driven semantic frame composition in lexicalized tree adjoining grammars. *Journal of Language Modelling*, 1:267–330.

Laura Kallmeyer and Rainer Osswald. 2014. Syntax-driven semantic frame composition in lexicalized tree adjoining grammars. *Journal of Language Modelling*, 1(2):267–330.

Laura Kallmeyer, Rainer Osswald, and Robert D. Van Valin, Jr. 2013. Tree Wrapping for Role and Reference Grammar. In *Formal Grammar 2012/2013*, volume 8036 of *LNCS*, pages 175–190. Springer.

Laura Kallmeyer and Maribel Romero. 2004. LTAG semantics with semantic unification. In *Proceedings of TAG+7*, pages 155–162, Vancouver.

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir Radev, and Owen Rambow. 2019. Syntax-aware neural semantic role labeling with supertags. *arXiv preprint arXiv:1903.05260*.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks.

Yudong Liu and Anoop Sarkar. 2009. Exploration of the ltag-spinal formalism and treebank for semantic role labeling. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks*, pages 1–9. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022. Transparent semantic parsing with universal dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint conference on EMNLP and CoNLL-shared task*, pages 1–40.

Robert D. Van Valin, Jr. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge University Press.

Robert D. Van Valin, Jr. and William A. Foley. 1980. Role and reference grammar. In E. A. Moravcsik and J. R. Wirth, editors, *Current approaches to syntax*, volume 13 of *Syntax and semantics*, pages 329–352. Academic Press, New York.

Robert D. Van Valin, Jr. and Randy LaPolla. 1997. *Syntax: Structure, meaning and function*. Cambridge University Press.

Jakub Waszczuk. 2017. *Leveraging MWEs in practical TAG parsing: towards the best of the two worlds*. Ph.D. thesis, Université François Rabelais Tours.

Qingrong Xia, Zhenghua Li, Min Zhang, Meishan Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019. Syntax-aware neural semantic role labeling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7305–7313.

Yulia Zinova and Laura Kallmeyer. 2012. A frame-based semantics of locative alternation in LTAG. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+11)*, pages 28–36, Paris.

The argument–adjunct distinction in BERT: A FrameNet-based investigation

Dmitry Nikolaev **Sebastian Padó**
Institute for Natural Language Processing, University of Stuttgart
dnikolaev@fastmail.com pado@ims.uni-stuttgart.de

Abstract

The distinction between arguments and adjuncts is a fundamental assumption of several linguistic theories. In this study, we investigate to what extent this distinction is picked up by a Transformer-based language model. We use BERT as a case study, operationalizing arguments and adjuncts as core and non-core FrameNet frame elements, respectively, and tying them to activations of particular BERT neurons. We present evidence, from English and Korean, that BERT learns more dedicated representations for arguments than for adjuncts when fine-tuned on the FrameNet frame-identification task. We also show that this distinction is already present in a weaker form in the vanilla pre-trained model.

1 Introduction

The widely used Transformer-based contextualized language model BERT (Devlin et al., 2019) has been extensively studied regarding its capability to uncover linguistic patterns from raw text, with analyses focused mostly on syntax. Both constituency and dependency trees were either found encoded inside the model or were used to probe for syntactic rules such as agreement (Jawahar et al., 2019; Rogers et al., 2020).

In this paper, we shift the focus of BERT analysis to the syntax-semantics interface, considering the foundational distinction between arguments and adjuncts. According to Koenig et al. (2003), arguments and adjuncts differ in two crucial ways: arguments describe necessary participants in the event described by the verb and are therefore both *obligatory*, i.e. they have to be realized by default, and *specific*, i.e. they express idiosyncratic properties of the event or the event class. In contrast, neither is necessarily true for adjuncts. For example, in the sentence *Peter praised his colleague repeatedly*, the praising event is accompanied by two necessary, specific participants, namely a communicator, *Peter*, and an evaluee, *the colleague*; in contrast, the adverb *repeatedly*, which specifies the frequency, could be left out and applies to a very broad range of events. The argument–adjunct distinction has played a major role in linguistic theory (Chomsky 1981; Pollard and Sag 1994, but see Przepiórkowski 2016) and has implications for human language processing (Tutunjian and Boland, 2008) and semantic NLP (Zhang et al., 2020).

We empirically assess the status of the argument–adjunct disinction in BERT by making use of FrameNet (Baker et al., 1998) – an implementation of frame semantics (Fillmore, 1982), a theory of predicate-argument structure, which describes predicate meaning in terms of frames (prototypical situations) and frame elements (the situations' participants). FrameNet maintains a distinction between *core elements* and *non-core elements*, which maps onto the argument–adjunct distinction (see Section 2 for details).

We use a modification of the method of model analysis proposed by Rethmeier et al. (2020) for associating neurons inside neural-network models with features they are particularly attuned to. In our main analysis, we use FrameNet annotations to fine-tune BERT for a task – frame identification, – for which frame elements are informative, without exposing the frame-element labels to the model, and then correlate the learned model representations with the presence of these labels. We also repeat the correlational analysis on the vanilla (pre-trained) BERT model.

Our contribution is twofold: (a) we extend Rethmeier et al.'s methodology, which targeted LSTMs, to BERT and, instead of constructing a probability distribution of features a given neuron is attuned with, we extract tight neuron–feature combinations using correlation analysis, reminiscent of the larger neuroscience literature on input-specific neural activations (Dayan and Abbott, 2001); (b) we use this method for an analysis of the representation of arguments vs. adjuncts in Multilingual BERT

Proceedings of the 15th International Conference on Computational Semantics pages 233–239
June 21–23, 2023. ©2023 Association for Computational Linguistics

(mBERT) based on English and Korean data. We find that even though BERT representations are dominated by frequency effects, with common input patterns more robustly tracked by individual neurons, arguments and adjuncts differ in their activation patterns (arguments produce relatively more robust activations while adjuncts generally lack highly specialized neurons that track them) and that this distinction is already present, to a lesser extent, in a vanilla pre-trained model.[1]

2 Frame Semantics and FrameNet

Frame semantics (Fillmore, 1982) posits that a key element of the understanding of an utterance is knowledge about the situations that the predicates in it evoke. This knowledge is captured through *frames*, schemas that associate predicates (*frame-evoking elements* / FEEs) with situations, their inferences, and their relevant participants, which are realized in language as so-called *frame elements*. Frame-semantic resources were first developed for English (FrameNet; Baker et al., 1998) but have been extended to other languages (Baker et al., 2018).

The example given in the introduction, *Peter praised his colleague repeatedly*, evokes the JUDGMENT_COMMUNICATION frame where a COMMUNICATOR expresses an evaluation of an EVALUEE. These are two of the *core elements* (CEs) of this frame, which generally meet both of Koenig et al.'s criteria for argumenthood: they are obligatory (unless they are null-instantiated, cf. Fillmore 1986) and they are specific to frames (or groups of closely related frames, cf. Fillmore et al. 2004). In contrast, the JUDGMENT_COMMUNICATION frame contains a number of *non-core elements* (NCEs), which do not meet at least one of the two criteria and thus show adjunct behavior: they are either not specific (MANNER, FREQUENCY) or not obligatory (GROUNDS: the basis for the judgment; ROLE: the capacity of the evaluee). A similar situation obtains with many other frequently found frames, and we assume that the core vs. non-core distinction largely mirrors the argument/adjunct dichotomy.

Data For our experiments, we use FrameNet corpora in English and Korean. For English, we use the FrameNet 1.7 lexical unit annotations, which cover over 1.2k frames and 13k unique predicates. The Korean FrameNet was created around a set of about 4k sentences translated from English, which were then added to using crowd sourcing. It aims for full compatibility with the English FrameNet (Hahm et al., 2020). We select 50 most frequent frames in both languages for analysis; the full list is given in the Appendix. There are 34,373 sentences in the English train set and 3,819 sentences in the test set. We use the Korean dataset only as a test set in a zero-shot setting. It contains 4,591 sentences.

3 Experimental Setup

Fine-tuning BERT We start from a pre-trained BERT model and fine-tune it to assign a single frame to each sentence (Hermann et al., 2014) in line with the FrameNet annotation (cf. Section 2).

We experiment with two variants of the task. In the *FEE present* setting, the model is shown complete sentences, including the FEEs, but no frame-element annotation. This task aims at encouraging the model to connect FEEs with arguments, which are known to be relevant for frame identification (Yang and Mitchell, 2017). Adjuncts are expected to be less relevant (as they are unspecific) or less reliable (as they are optional). To select the frame, we feed the first subword of the first FEE token to a fully-connected 50-neuron layer (corresponding to the 50 frames) and obtain a prediction by applying the usual softmax.[2]

In the *FEE masked* setting, all FEE tokens are replaced with the [MASK] token, so that the model has to rely on the sentential context to identify the frame. Our hypothesis is that this version of the task incentivizes the model to more actively focus on extracting arguments. In this case, we feed the embedding of the first masked token into the frame classification head as above.

In both variants, the model is trained end-to-end using cross-entropy loss for twenty epochs with early stopping when the performance on the test set decreases. We use the pre-trained mBERT model provided by HuggingFace (Wolf et al., 2020). For English, we report results for the test set. For Korean, we adopt a zero-shot setting and, after checking that mBERT fine-tuned on English has some success in identifying Korean frames, analyze the activations that Korean sentences produce in it.

[1] The code used for the analyses in this paper is available at https://github.com/macleginn/argument-adjunct-framenet

[2] We opt for a simplistic classifier head to keep more information in the embeddings.

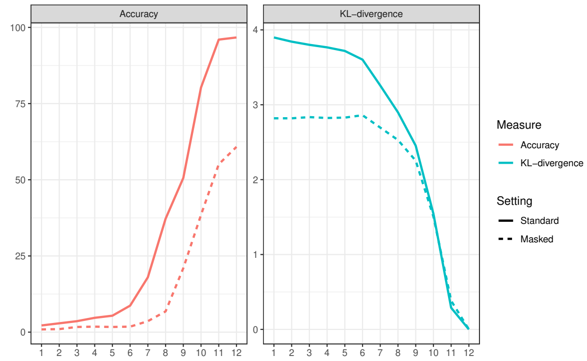


Figure 1: Left: Accuracy of predictions based on the output of different layers (development set). Right: the Kullback–Leibler divergence between the probability distribution of frame labels induced by intermediate layers and by the final layer.

Probing analysis Once the BERT model has been fine-tuned, we can analyze the activation patterns of different layers of the model (Rethmeier et al., 2020). On the data side, we cast input sentences as a binary matrix whose columns correspond to the presence or absence of each CE or NCE in these sentences, i.e. to their indicator functions. On the model side, we associate each input sentence with a $d = 768$-dimensional embedding of the first subword of the FEE token or the embedding of the first [MASK] token, depending on the setting, for a selected subset of BERT layers. We then carry out correlational analysis to identify, for each CE or NCE indicator function and for each layer of interest, the neuron whose activations are most strongly correlated with these functions.

To choose layers for the analysis, we evaluated English model predictions based on the representations in each layer. The results are shown in Figure 1. For both variants of the task, we find similar results: the outputs of the 11th layer are close to the final layer, and there is a swift increase in prediction accuracy from the 7–8th layer onward. On this basis, we probe the activation patterns of layer 11 (near-convergence) and layer 9 (start of competitive performance).

Analysis of neural activity was performed in a similar fashion by Durrani et al. (2020). They, however, extract activations in the context of specific tasks, such as POS tagging and syntactic chunking, instead of feeding sentences to a headless embedding model in an unsupervised setting.

Language	FEE present	FEE masked	MBL	RBL
EN	96	55	15	2
KO	40	21	12	2

Table 1: Frame ID accuracy in % on test set (layer 11). MBL: majority class baseline, RBL: random baseline

4 Results and Discussion

English Table 1 shows the test-set performance of layer 11 in the fine-tuned model.[3] As expected, the FEE-present setting is much easier than the FEE-masked one, where the model still substantially outperforms the baselines.

The results of the correlation analysis are presented in the scatterplots in Figure 2. Individual points show, for a frame element with a given frequency, how large the correlation with the most attuned neuron activation vector in the respective model is. The left plot shows core elements, the right plot non-core elements.

The plots show that frequency is the dominating factor: high-frequency frame elements tend to have (more or less) dedicated neurons tracking them, with correlations of 0.4 and above, while this is not true for low-frequency frame elements. This is to be expected given the maximum-likelihood training objective.

However, there still is a clear difference between CEs and NCEs: even the most frequent NCEs do not attain correlations above 0.3, and only a handful show correlations above 0.2, in both the standard and masked settings. In contrast, the correlations for CEs with frequencies above 100 are all higher than 0.2. This shows the model's low reliance on NCEs for frame identification.

Comparing the behaviors at layers 9 (red) and 11 (turquoise), we do not see major differences: in particular for NCEs, the plots are extremely similar. Comparing the two variants of the task (solid vs. dashed), we see that the masked-task model learns less dedicated representations for the CEs but spends some more effort on representing high-frequency NCEs – contrary to the expectation we formulated in Section 3. The global advantage of CEs over NCEs in all settings leads us to believe that the model simply relies on arguments in either case, and that in the masked setting the model just struggles more to identify where they are.

[3] Results for layer 12: 96.5/60.8 (EN), 41.3/23.9 (KO).

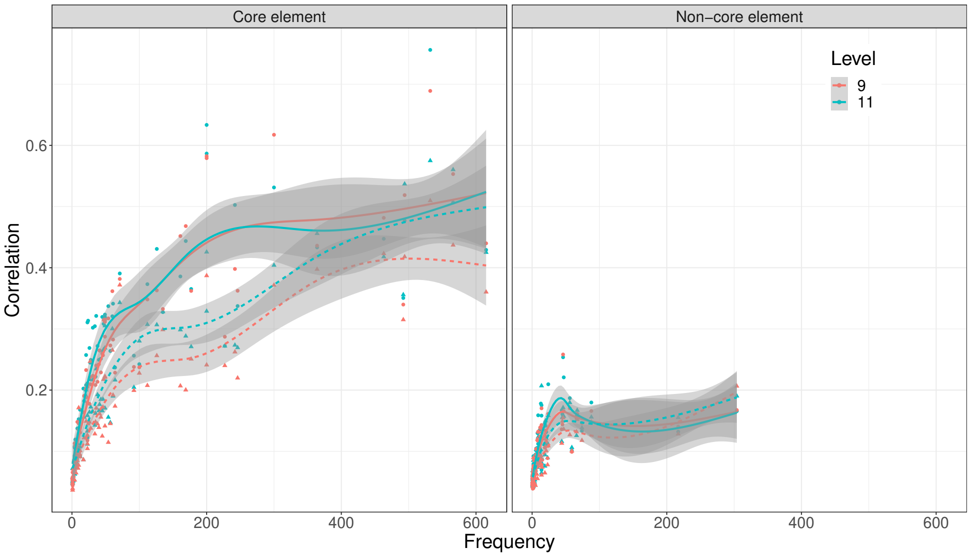


Figure 2: English fine-tuned setting: Averages and 95% confidence intervals for maximal correlations between BERT neurons and CEs (left) / NCEs (right), by frequency. Solid/dashed lines: FEE present/masked task. The curves show GAM-smoothed averages with 95% confidence intervals.

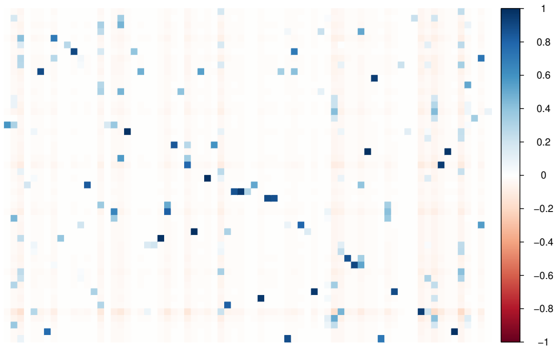


Figure 3: Correlations between frames (rows) and their core elements (columns).

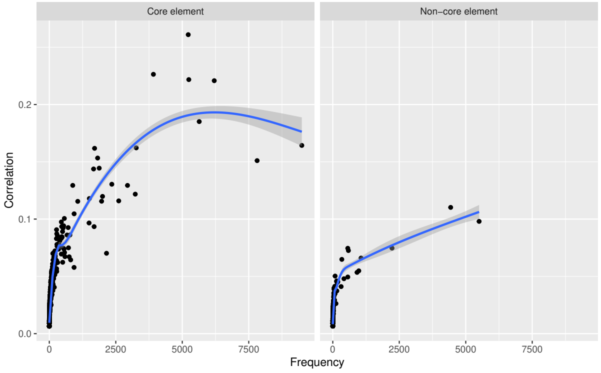


Figure 4: English vanilla BERT: GAM-smoothed averages and 95% confidence intervals for maximal correlations between neurons and CEs (left) / NCEs (right), by frequency. Solid/dashed lines: FEE present/masked task.

This interpretation is corroborated by an analysis of CE information content. Figure 3 shows a matrix of correlations between frames with non-masked FEEs (rows) and their CEs (columns). Some frames are in a nearly one-to-one correspondence with their CEs, but other CEs can be found with several frames. Arguably, when FEEs are present, they form a strong signal together with the CEs pointing towards particular frames. When FEEs are masked, however, frequent CEs – precisely those that are found with many different frames – become less informative, and the model shifts some of the weight towards NCEs.

Korean The accuracy results for the zero-shot application to Korean in Table 1 show similar tendencies to English, but with much lower accuracies. We attribute this to the simplistic linear classifier we use (cf. the observations on multilingual zero-shot transfer by Lauscher et al. 2020). However, the results of the correlation analysis shown in Figure 5 are strikingly similar to English: (a) top correlations of neural activations with CEs are much higher than those with NCEs; (b) strong frequency effects are evident; (c) the masked variant moves some focus from CEs to high-frequency NCEs. We take these observations as evidence that mBERT represents arguments and adjuncts in a remarkably similar way across languages as different as English and Korean, with the latter's rich morphology and SOV word order.

Without fine-tuning The above analysis uses a fine-tuned model. This begs the question of

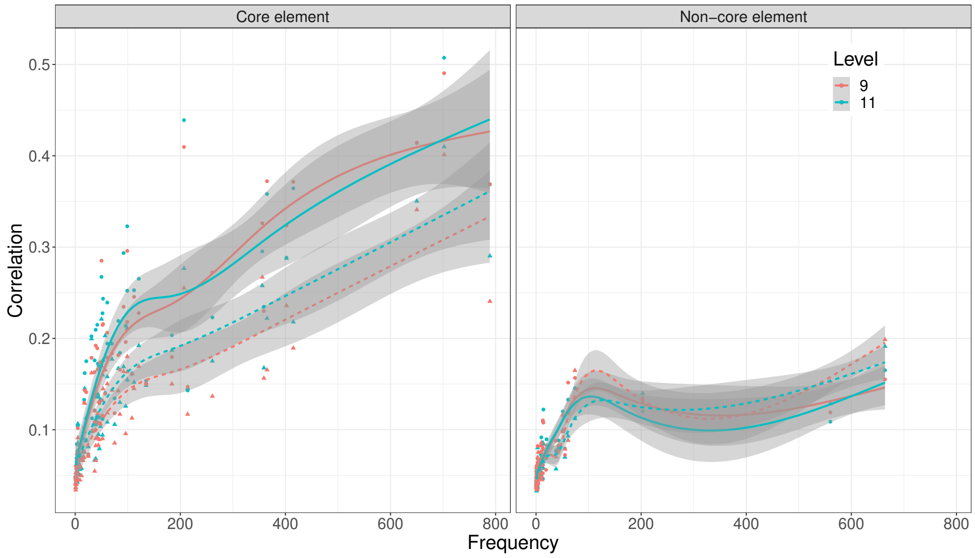


Figure 5: Korean zero-shot setup (model fine-tuned for English): GAM-smoothed averages and 95% confidence intervals for maximal correlations between BERT neurons and CEs (left) / NCEs (right), by frequency. Solid/dashed lines: FEE present/masked task.

whether the distinction between arguments and adjuncts is a side-effect of the fine-tuning task, as opposed to mBERT's acquiring it in an unsupervised way in pre-training (Tenney et al., 2019). To test this, we repeat the experiment using a vanilla English pre-trained model and the complete Berkeley FrameNet 1.7 release instead of the sentences with most-frequent frames. The results for layer 11, shown in in Figure 4, are remarkably similar in terms of the general pattern but with significantly weaker correlations: for CEs, correlations exceed 0.1 reliably for $N > 1000$, with maximum values approaching 0.3.[4] For NCEs, correlations are almost always < 0.1, reaching this value only for the most frequent NCEs, with $N \approx 5000$. This indicates that after pre-training BERT already has some notion of the distinction between arguments and adjuncts, but that this distinction becomes substantially more pronounced after fine-tuning on a task for which it is relevant.

5 Conclusion

Our study asked whether BERT can distinguish between arguments and adjuncts and operationalized these concepts via FrameNet's core vs. non-core frame-element distinction. For both English and Korean, our analysis of the presence of dedicated neurons that track individual frame elements found that this is the case, with frequency as a major covariate. The picture is clearer for a fine-tuned model, but the main patterns emerge already after pre-training.

On the neural-language-model side, our study confirms the ability of such models to recover 'deep' linguistic categories in an unsupervised manner. On the FrameNet side, our results have bearing on the status of borderline-core frame elements (Ruppenhofer et al., 2006), for which the behaviour of the model may serve as a heuristic. A promising avenue for future work would be to turn around our setup and to explore BERT representations in order to identify a set of properties that differentiate arguments and adjuncts from the model's point of view, *à la* Geva et al. (2021).

This work has focused on FrameNet. Other frameworks giving access to semantic-role information, such as the PropBank annotation scheme (Palmer et al., 2005), AMR (Banarescu et al., 2013), and UCCA (Abend and Rappoport, 2013), also may be fruitful for this type of analysis.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

[4] Two most-frequent frames, AGENT and THEME, are very general and unsurprisingly display weaker correlations. By comparison, the next three most-frequent frames, SPEAKER, GOAL, and TIME, are much richer semantically and have more dedicated representations.

Collin F. Baker, Michael Ellsworth, Miriam R. L. Petruck, and Swabha Swayamdipta. 2018. Frame semantics across languages: Towards a multilingual FrameNet. In *Proceedings of the 27th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 9–12, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Noam Chomsky. 1981. *Lectures on government and binding*. Forus.

Peter Dayan and Laurence F Abbott. 2001. *Theoretical neuroscience*. MIT Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Charles J. Fillmore. 1982. Frame semantics. In *Linguistics in the Morning Calm*, pages 111–137. Hanshin Publishing Co., Seoul, South Korea.

Charles J Fillmore. 1986. Pragmatically controlled zero anaphora. In *Proceedings of the Berkeley Linguistics Society*, volume 12, pages 95–107. Berkeley Linguistic Society, BLS.

Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato. 2004. FrameNet as a "net". In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Younggyun Hahm, Youngbin Noh, Ji Yoon Han, Tae Hwan Oh, Hyonsu Choe, Hansaem Kim, and Key-Sun Choi. 2020. Crowdsourcing in the development of a multilingual FrameNet: A case study of Korean FrameNet. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 236–244, Marseille, France. European Language Resources Association.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1458, Baltimore, Maryland. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jean-Pierre Koenig, Gail Mauner, and Breton Bienvenue. 2003. Arguments for adjuncts. *Cognition*, 89(2):67–103.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Carl Pollard and Ivan Sag. 1994. *Head-driven phrase-structure grammar*. Chicago University Press.

Adam Przepiórkowski. 2016. How not to distinguish arguments from adjuncts in LFG. In *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar*, pages 560–580.

Nils Rethmeier, Vageesh Kumar Saxena, and Isabelle Augenstein. 2020. TX-Ray: Quantifying and explaining model-knowledge transfer in (un-)supervised NLP. volume 124 of *Proceedings of Machine Learning Research*, pages 440–449, Virtual.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Josef Ruppenhofer, Michael Ellsworth, Miriam R L Petruck, Christopher R Johnson, and Jan Scheffczyk. 2006. FrameNet II: Extended Theory and Practice.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Damon Tutunjian and Julie E. Boland. 2008. Do we need a distinction between arguments and adjuncts? evidence from psycholinguistic studies of comprehension. *Language and Linguistics Compass*, 2(4):631–646.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Bishan Yang and Tom Mitchell. 2017. A joint sequential and relational model for frame-semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Copenhagen, Denmark. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9628–9635.

A Appendix

Most frequent frames used in the analysis

ARRIVING, ATTEMPT SUASION, AWARENESS, BECOMING AWARE, BODY MOVEMENT, BRINGING, CATEGORIZATION, CAUSE HARM, CAUSE MOTION, CHANGE POSITION ON A SCALE, CHANGE POSTURE, COGITATION, COMING TO BELIEVE, COMMITMENT, COMMUNICATION MANNER, COMMUNICATION NOISE, COMMUNICATION RESPONSE, CONTACTING, COTHEME, DEPARTING, DESIRING, EVIDENCE, EXPERIENCER FOCUS, EXPERIENCER OBJ, FILLING, FLUIDIC MOTION, GIVE IMPRESSION, IMPACT, INGESTION, JUDGMENT, JUDGMENT COMMUNICATION, JUDGMENT DIRECT ADDRESS, KILLING, LOCATION OF LIGHT, MANIPULATION, MOTION, MOTION NOISE, PERCEPTION ACTIVE, PERCEPTION EXPERIENCE, PLACING, REMOVING, REQUEST, RESIDENCE, REVEAL SECRET, SCRUTINY, SELF MOTION, STATEMENT, TELLING, TEXT CREATION, USING.

Collecting and Predicting Neurocognitive Norms for Mandarin Chinese

Le Qiu, Yu-Yin Hsu, Emmanuele Chersoni
Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University,
11 Yuk Choi Road, Hung Hom, Kowloon, Hong Kong (China)
lani.qiu@connect.polyu.hk, {yu-yin.hsu,emmanuele.chersoni}@polyu.edu.hk

Abstract

Language researchers have long assumed that concepts can be represented by sets of semantic features, and have traditionally encountered challenges in identifying a feature set that could be sufficiently general to describe the human conceptual experience in its entirety.

In the dataset of English norms presented by Binder et al. (2016), also known as **Binder norms**, they introduced a new set of *neurobiologically motivated* semantic features in which conceptual primitives were defined in terms of modalities of neural information processing. However, no comparable norms are currently available for other languages.

In our work, we built the Mandarin Chinese norm by translating the stimuli used in the original study and developed a comparable collection of human ratings for Mandarin Chinese. We also conducted some experiments on the automatic prediction of the Chinese Norms based on the word embeddings of the corresponding words to assess the feasibility of modeling experiential semantic features via corpus-based representations.

1 Introduction

A longstanding research trend in semantics assumes that the conceptual content of lexical items can be decomposed into semantic features identifying basic meaning components (Vigliocco and Vinson, 2007). Such features represent semantic primitives that can be present or absent in the semantic representation of a lexeme, such as *boy* in Example (1).

(1) *boy* [+MALE, -MATURE . . .]

However, this type of view has some critical limitations: First, discrete features are not suitable to address the gradient prototypicality of feature-to-concept associations (Murphy, 2002). Second, these feature sets tend to be manually selected, and are generally tailored to a few *in vitro* examples; thus, they are unable to account for large portions of the lexicon of natural languages (Chersoni et al., 2021).

On one hand, featural representations have the advantage of *human interpretability*, as they label the dimensions of word meanings explicitly, and provide explanatory factors for their semantic behavior; for example, the similarity between *beer* and *coffee* can be explained by assuming that they share the semantic feature of LIQUID. On the other hand, this type of features is highly subjective, and can only be collected through a time-consuming process of elicitation from human subjects (e.g. McRae et al. (2005); Vinson and Vigliocco (2008); Devereux et al. (2014); Buchanan et al. (2019)).

An alternative was proposed by Binder et al. (2016) using **brain-based semantics** based on *modalities of neural information processing*. After reviewing extensive evidence from studies of human physiology, the authors proposed a dataset of 535 words described in terms of 68 experiential features, each of which was associated with a specific neural processing in the neurobiological literature. The features were categorized according to 14 different domains of experience (Table 1).

The proposal by Binder et al. (2016) should naturally extend to other languages: If the features are genuinely neurobiologically motivated, it should also be possible to use them to describe the essential meaning components of languages other than English.[1] However, to the best of our knowledge, Binder-like norms are currently only available for the English language.[2]

[1] See also the recent work of Blasi et al. (2022) on the need for cognitive science studies to look beyond English, in order to support claims of universality.

[2] A partial exception is represented by the collection of ratings published by Wang et al. (2022); see Section 2.

Proceedings of the 15th International Conference on Computational Semantics pages 240–245
June 21–23, 2023. ©2023 Association for Computational Linguistics

Domain Type	Domain	Meaning components (features)
Sensory	Vision	VISION, BRIGHT, DARK, COLOUR, PATTERN, LARGE, SMALL, MOTION, BIOMOTION, FAST, SLOW, SHAPE, COMPLEXITY, FACE, BODY
Sensory	Somatic	TOUCH, HOT, COLD, SMOOTH, ROUGH, LIGHT, HEAVY, PAIN
Sensory	Audition	AUDITION, LOUD, LOW, HIGH, SOUND, MUSIC, SPEECH
Sensory	Gustation	TASTE
Sensory	Olfaction	SMELL
Motor	Motor	HEAD,UPPER LIMB, LOWER LIMB, PRACTICE
Spatial	Spatial	LANDMARK, PATH, SCENE, NEAR, TOWARD, AWAY
Number	Number	NUMBER
Event	Temporal	TIME, DURATION, LONG, SHORT
Event	Causal	CAUSED, CONSEQUENTIAL
Event	Social	SOCIAL
Cognition	Cognition	HUMAN, COMMUNICATION, SELF, COGNITION
Evaluation	Evaluation	BENEFIT, HARM, PLEASANT, UNPLEASANT
Emotion	Emotion	HAPPY, SAD, ANGRY, DISGUSTED, FEARFUL, SURPRISED
Drive	Drive	DRIVE, NEEDS
Attention	Attention	ATTENTION, AROUSAL

Table 1: List of the domains and meaning components (features) in Binder et al. (2016).

Therefore, in our work, we adopted the same design of Binder norms: We translated the words in the Binder dataset into Mandarin Chinese, and obtained ratings from human subjects for each of the 68 Binder features per word in order to obtain a comparable dataset. Moreover, we experimented with regression algorithms to assess the extent to which such norms could be predicted automatically based on the text-derived embeddings of the corresponding words.[3]

2 Related Work

Neurosemantic decoding research, initiated by the seminal work of Mitchell et al. (2008), has the aim of creating mappings between different concept representations, typically from a corpus-derived one (such as word embedding) to one derived from human data (such as fMRI scans and semantic norms). For example, previous studies used fMRI data to learn mapping from the traditional count-based distributional models (Devereux et al., 2010; Murphy et al., 2012), including both count- and prediction-based vectors (Bulat et al., 2017; Abnar et al., 2018), and topic models (Pereira et al., 2011, 2013); the same methodology has been used to map word-embedding models onto feature (Fagarasan et al., 2015; Bulat et al., 2016; Derby et al., 2019) and modality norms (Chersoni et al., 2020) to ground the vectors in perceptual data and to make them interpretable. Due to the grounding on perceptual experience, the Binder features for English have also been used for the same purpose (Utsumi, 2018; Turton et al., 2020; Chersoni et al., 2021). Notice that, differently from property norms (McRae et al., 2005; Devereux et al., 2014), the collection process is more constrained: the properties of concepts are not freely elicited from human participants; because the Binder features are a closed set, the participants were asked to only rate the relevance of a given feature for a given concept.

We are not currently aware of any other work that has introduced Binder-like norms for languages other than English. The recent work by Wang et al. (2022) introduced a fMRI dataset for Mandarin Chinese, together with a collection of Binder ratings for the target words. However, their targets differed from those in the original study by Binder et al. (2016) (a total of 672 words from the Synonymy Thesaurus of the Harbin Institute of Technology), and the representation was limited to 54 Binder features, as some of them were excluded due to high levels of correlation with at least one of the other features. With the aim of providing a comparable and more comprehensive resource to facilitate future experiments on the prediction of crosslingual norms, we opted to retain the original set of target words and features.

3 Data Collection

Binder et al. (2016) collected ratings for 68 cognitively-motivated features for 535 words in total.[4] 242 words were selected from the Knowledge Representation in Neural Systems project (Glasgow et al., 2016), including 141 nouns, 62 verbs,

[3] Dataset and code for the experiments will be available at the following URL: `https://github.com/Laniqiu/norming`.

[4] In their paper, they used the feature label *Temperature* for features Hot and Cold, *Texture* for Smooth and Rough, and *Weight* for Light and Heavy, resulting in 65 feature categories.

Type-POS	No. of items
Concrete Objects - Nouns	275
Living Things - Nouns	126
Other Natural Objects - Nouns	19
Artifacts - Nouns	130
Concrete Events - Nouns	60
Abstract Entities - Nouns	99
Concrete Actions - Verbs	52
Abstract Actions - Verbs	5
States - Verbs	5
Abstract Properties - Adjectives	13
Physical Properties - Adjectives	26

Table 2: Concept types, parts of speech (POS), and the number of items in the dataset by Binder et al. (2016).

Word	Vision	Bright	...	Cognition	Benefit
公寓*(gongyu)*	5.56	3.82	...	0.86	4.60
杏子*(xingzi)*	4.24	4.06	...	1.34	3.34

Table 3: Sample of Binder vectors for the words *gongyu (apartment)* and *xingzi (apricot)*.

and 39 adjectives, while another 293 words were added to include more abstract nouns. We adopted the original set of 535 target words and 68 features proposed by Binder et al. (2016), and the original survey queries that they proposed. We translated them into Mandarin Chinese using simplified characters. This survey was used to elicit the ratings for the salience of each attribute for each target word, with the same 0-6 Likert scale used in the original study (the higher the score, the higher the relevance of a feature when one has to think about the target concept, while 0 corresponds to "feature not applicable to this concept").

The target words and the survey queries were translated by two native speakers of Mandarin, who were Master's students of linguistics. For features and target words, we adopted their most common and core sense in English to translate into their corresponding Chinese. While some words in colloquial uses may have multiple senses, we selected more specific words which were equally frequent to the polysemous ones and to the sense expressed by the English counterparts. We were aware that the concept of "adjective" could sometimes not easily be recognized in Chinese, just as the function of words in the *-ed* form can be ambiguous in English as either adjectival or verbal past participle. When an adjective could be interpreted as other parts of speech categories (POS), we added an adjectival suffix -的 *de* to such adjectives to avoid such potential confusion. The final version of the survey queries and the target words were manually checked by one of the authors, who is also a native Mandarin speaker. The same POS of each word were maintained for the 535 words, and each word was associated with survey questions pertaining to the 68 cognitively motivated features. One target word in the survey, *banjo*, was replaced for a more culturally relevant musical instrument, 二胡 *erhu*, while the other words were the same as their English counterparts.

As is the case for the Binder norms, we adopted a continuous rating design to obtain the attributes for each word. We collected the data on a crowdsourcing platform that is commonly used in China (问卷星 Wenjuanxing), because the rating results might occur along a continuum and could be subjective due to the speakers' personal experiences and backgrounds, thus, a larger sample size was considered to be helpful in overcoming this issue. We obtained 8025 sets of ratings from the crowdsourcing survey; each of the 535 targets obtained 15 sets of rating results covering all 68 features. The demographics and the language backgrounds of the participants were checked before they participated in the survey. Each participant received RMB$20 after completing the survey and once their results had passed the survey's attention checks.

After completing the survey, we measured the Spearman correlation between English and Chinese ratings. We found out that the ratings were quite consistent across languages: on average, we obtained a correlation of 0.68 across words and a correlation of 0.59 across features.

4 Experiments

In order to learn to map between word-embedding spaces and our Chinese Binder features, we trained regression models using three different regressors, namely Ridge Regression, Random Forest and Multilayer Perceptron (MLP) [5], using the ratings of the 68 features in the dataset as the dependent variables and the dimensions of pretrained word-embedding models as the independent variables.

Considering that the task requires mapping between word types that are taken out of context, we decided to use static word-embedding mod-

[5] The regression models were implemented using Scikit-learn (Pedregosa et al., 2011) with standard hyperparameters. The only exception was the MLP, for which we selected the following parameters after a parameter search: hidden_layer_sizes=(50, 10), activation='identity', early_stopping=True, max_iter=1000 (the other parameters are the default ones).

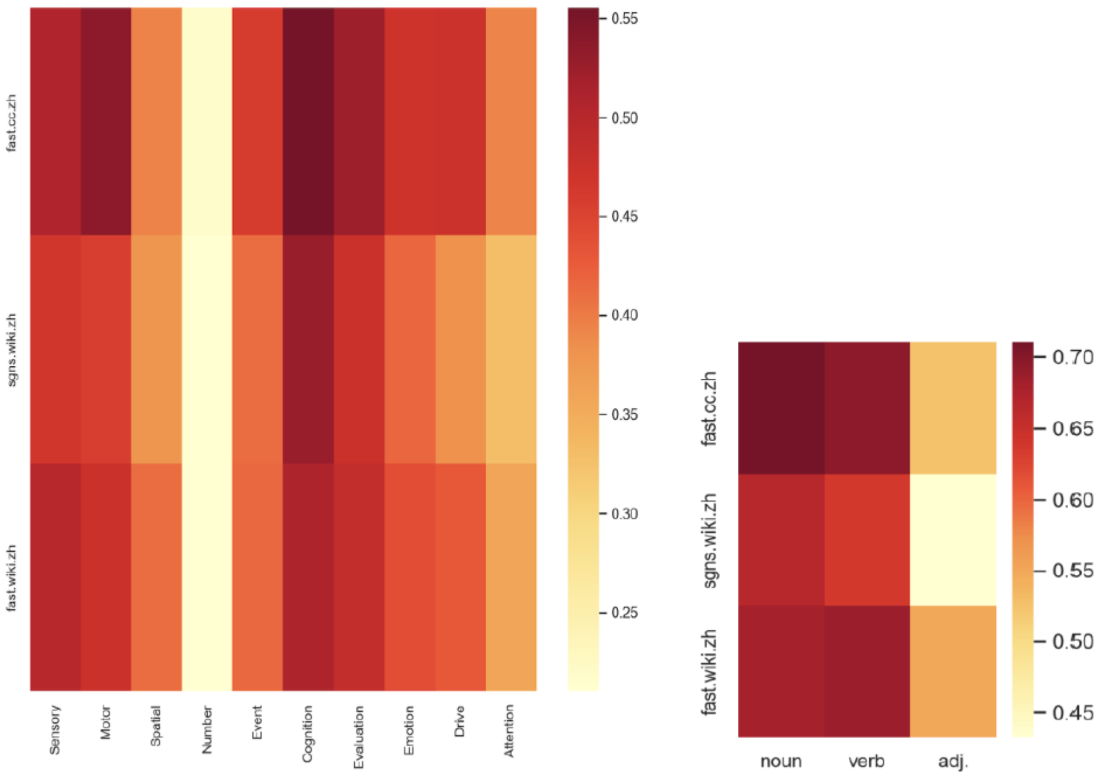


Figure 1: Feature correlation scores by domain type (left) and word correlation scores by POS (right).

els: we used four different types of embeddings: count-based sparse PPMI vectors (Church and Hanks, 1990; Bullinaria and Levy, 2007) that were trained on the Chinese Wikipedia (*ppmi.wiki.zh*; Qiu et al. (2018)), Skip-Gram vectors (Mikolov et al., 2013) that were trained on the Chinese Wikipedia (*sgns.wiki.zh*, Qiu et al. (2018)), and FastText vectors that were trained on the Chinese Common Crawl (*fast.cc.zh*) or on the Chinese Wikipedia (*fast.wiki.zh*) (Bojanowski et al., 2017)). All the embedding models had 300 dimensions as input features for the regressor, except for the sparse PPMI vectors, which had 350k dimensions. In addition, we initialized 300-dimensional random vectors for all the words in the dataset, and used them to train similar regression models as baselines (*Random*). In future, we also plan to test contextualized word embeddings (Devlin et al., 2019) in the task, although it is worth pointing out that their performance in out-of-context semantic tasks has recently been shown not to differ significantly from that of static models (Lenci et al., 2022).

Vectors	Model	Word	Feature
fast.cc.zh	Ridge	**0.70**	**0.49**
fast.cc.zh	RandomForest	0.66	0.36
fast.cc.zh	MLP	0.69	0.40
sgns.wiki.zh	Ridge	0.66	0.44
sgns.wiki.zh	RandomForest	0.63	0.33
sgns.wiki.zh	MLP	0.66	0.38
fast.wiki.zh	Ridge	0.68	0.47
fast.wiki.zh	RandomForest	0.64	0.35
fast.wiki.zh	MLP	0.69	0.44
ppmi.wiki.zh	Ridge	0.25	0.03
ppmi.wiki.zh	RandomForest	0.50	0.07
ppmi.wiki.zh	MLP	0.15	0.03
Random	Ridge	0.26	-0.01
Random	RandomForest	0.51	-0.02
Random	MLP	0.49	0.04

Table 4: Word and Feature Spearman correlation for all regression models (top scores are in **bold**).

Following Utsumi (2018), we adopted the **leave-one-out paradigm** for data splitting: For each of the n target words; we extracted one word out and trained a regression model on the other $n-1$ remaining words, and then we used the last word as the test set. The standard metric of the Spearman correlation was computed to compare the vectors of the Binder features predicted by the models and the gold vectors of human ratings (note that only one word was predicted for each run).

5 Results

The results in Table 4 reveal that embedding models based on FastText and Skip Gram had highly significant correlations with human scores, and that the FastText vectors trained on Common Crawl achieved higher scores than did any of the ones trained on Wikipedia. However, the sparse PPMI vectors had a much weaker performance, to the extent that the scores were close to the regressors initialized using the random vectors. Both the models with random and with PPMI vectors failed to achieve significant correlations at the feature level. Ridge Regression models were the most accurate, particularly for the correlations at the feature level. However, it should be said that the differences between the regressors trained with Skip-Gram and FastText are small and not significant, also due to the relatively small size of the samples.[6]

We also analyzed the features and the POS that were predicted better, in comparison to Chersoni et al. (2021)'s experiment using English data (see Figure 1). Our analyses revealed that, similarly to English, the predictions for the COGNITION domain were the best. This is not surprising, because this domain is important for characterizing abstract concepts, of which textual/linguistic information is probably the prevailing source for human concept learning (Vigliocco et al., 2009). Sensory and Motor features were also predicted at relatively high correlations level, suggesting that many aspects of experiential, first-hand information can still be retrieved from linguistic data (Riordan and Jones, 2011). Finally, domains related to Spatial, Temporal (NUMBER and EVENT) and Attention turned out to be most challenging ones, coherently with the findings of Chersoni et al. (2021)'s experiment.

It can also be seen that, while English nouns were predicted much better than other POS, similar correlations were observed for nouns and verbs in Chinese (adjectives were the most difficult in both languages).

6 Conclusions

In this paper, we introduced Binder-style norms for Mandarin Chinese, collected using a similar method to the original study, and ran regression experiments from embeddings to norms, showing that the latter can be predicted with moderate to high correlations with humans. Such an application is especially interesting because it allows to extend the norms to large portions of the lexicon.

In the future, we plan to experiment with regression models based on contextualized vectors and to run tests for zero-shot crosslingual norms predictions, which could pave the way for the automatic acquisition of norms in low-resource languages.

Acknowledgments

This work has been supported by the Start-up Fund for new recruits of the Hong Kong Polytechnic University (1-BE8G) and the RGC Direct Allocation Grant (A-PB1C).

References

Samira Abnar, Rasyan Ahmed, Max Mijnheer, and Willem Zuidema. 2018. Experiential, Distributional and Dependency-Based Word Embeddings Have Complementary Roles in Decoding Brain Activity. In *Proceedings of the LSA Workshop on Cognitive Modeling and Computational Linguistics*.

Jeffrey R Binder, Lisa L Conant, Colin J Humphries, Leonardo Fernandino, Stephen B Simons, Mario Aguilar, and Rutvik H Desai. 2016. Toward a Brain-Based Componential Semantic Representation. *Cognitive Neuropsychology*, 33(3-4):130–174.

Damián E Blasi, Joseph Henrich, Evangelia Adamou, David Kemmerer, and Asifa Majid. 2022. Over-reliance on English Hinders Cognitive Science. *Trends in Cognitive Sciences*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Erin M Buchanan, Kathrene D Valentine, and Nicholas P Maxwell. 2019. English Semantic Feature Production Norms: An Extended Database of 4436 Concepts. *Behavior Research Methods*, 51:1849–1863.

Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. Speaking, Seeing, Understanding: Correlating Semantic Models with Conceptual Representation in the Brain. In *Proceedings of EMNLP*.

Luana Bulat, Douwe Kiela, and Stephen Christopher Clark. 2016. Vision and Feature Norms: Improving Automatic Feature Norm Learning Through Cross-Modal Maps. In *Proceedings of NAACL-HLT*.

John A Bullinaria and Joseph P Levy. 2007. Extracting Semantic Representations from Word Co-Occurrence Statistics: A Computational Study. *Behavior Research Methods*, 39:510–526.

[6] p-values computed with Fisher's r-to-z transformation.

Emmanuele Chersoni, Enrico Santus, Chu-Ren Huang, and Alessandro Lenci. 2021. Decoding Word Embeddings with Brain-based Semantic Features. *Computational Linguistics*, 47(3):663–698.

Emmanuele Chersoni, Rong Xiang, Qin Lu, and Chu-Ren Huang. 2020. Automatic Learning of Modality Exclusivity Norms with Crosslingual Word Embeddings. In *Proceedings of *SEM*.

Kenneth Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1):22–29.

Steven Derby, Paul Miller, and Barry Devereux. 2019. Feature2Vec: Distributional Semantic Modelling of Human Property Knowledge. In *Proceedings of EMNLP*.

Barry Devereux, Colin Kelly, and Anna Korhonen. 2010. Using fMRI Activation to Conceptual Stimuli to Evaluate Methods for Extracting Conceptual Representations from Corpora. In *Proceedings of the NAACL Workshop on Computational Neurolinguistics*.

Barry J. Devereux, Lorraine K. Tyler, Jeroen Geertzen, and Billi Randall. 2014. The Centre for Speech, Language and the Brain (CSLB) Concept Property Norms. *Behavior Research Methods*, 46(4):1119–1127.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Luana Fagarasan, Eva Maria Vecchi, and Stephen Clark. 2015. From Distributional Semantics to Feature Norms: Grounding Semantic Models in Human Perceptual Data. In *Proceedings of IWCS*.

Kimberly Glasgow, Matthew Roos, Amy Haufler, Mark Chevillet, and Michael Wolmetz. 2016. Evaluating Semantic Models with Word-Sentence Relatedness. *arXiv preprint arXiv:1603.07253*.

Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. A Comparative Evaluation and Analysis of Three Generations of Distributional Semantic Models. *Language Resources and Evaluation*, 56(4):1269–1313.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. 2005. Semantic Feature Production Norms for a Large Set of Living and Nonliving Things. *Behavior Research Methods*, 37(4):547–559.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Tom M Mitchell, Svetlana V Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A Mason, and Marcel Adam Just. 2008. Predicting Human Brain Activity Associated with the Meanings of Nouns. *Science*, 320(5880):1191–1195.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Selecting Corpus-Semantic Models for Neurolinguistic Decoding. In *Proceedings of *SEM*.

Gregory Murphy. 2002. *The Big Book of Concepts*. MIT Press, Cambridge, MA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Francisco Pereira, Matthew Botvinick, and Greg Detre. 2013. Using Wikipedia to Learn Semantic Feature Representations of Concrete Concepts in Neuroimaging Experiments. *Artificial Intelligence*, 194:240–252.

Francisco Pereira, Greg Detre, and Matthew Botvinick. 2011. Generating Text from Functional Brain Images. *Frontiers in Human Neuroscience*, 5:72.

Yuanyuan Qiu, Hongzheng Li, Shen Li, Yingdi Jiang, Renfen Hu, and Lijiao Yang. 2018. Revisiting Correlations between Intrinsic and Extrinsic Evaluations of Word Embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 209–221. Springer.

Brian Riordan and Michael N. Jones. 2011. Redundancy in Perceptual and Linguistic Experience: Comparing Feature-based and Distributional Models of Semantic Representation. *Topics in Cognitive Science*, 3(2):303–345.

Jacob Turton, David Vinson, and Robert Smith. 2020. Extrapolating Binder Style Word Embeddings to New Words. In *Proceedings of the LREC Workshop on Linguistic and Neurocognitive Resources*.

Akira Utsumi. 2018. A Neurobiologically Motivated Analysis of Distributional Semantic Models. In *Proceedings of CogSci*.

Gabriella Vigliocco, Lotte Meteyard, Mark Andrews, and Stavroula Kousta. 2009. Toward a Theory of Semantic Representation. *Language and Cognition*, 1(2):219–247.

Gabriella Vigliocco and David P. Vinson. 2007. Semantic Representation. In Gareth Gaskell, editor, *The Oxford Handbook of Psycholinguistics*, pages 195–215. Oxford University Press, Oxford.

David P. Vinson and Gabriella Vigliocco. 2008. Semantic Feature Production Norms for a Large Set of Objects and Events. *Behavior Research Methods*, 40(1):183–190.

Shaonan Wang, Yunhao Zhang, Xiaohan Zhang, Jingyuan Sun, Nan Lin, Jiajun Zhang, and Chengqing Zong. 2022. An fMRI Dataset for Concept Representation with Semantic Feature Annotations. *Scientific Data*, 9(1):721.

Error Exploration for Automatic Abstract Meaning Representation Parsing

Maria Boritchev, Johannes Heinecke
Orange Innovation
2 avenue Pierre Marzin
22307 Lannion cedex, France
`{maria.boritchev,johannes.heinecke}@orange.com`

Abstract

Following the data-driven methods of evaluation and error analysis in meaning representation parsing presented in (Buljan et al., 2022), we performed an error exploration of an Abstract Meaning Representation (AMR) parser. Our aim is to perform a diagnosis of the types of errors found in the output of the tool in order to implement adaptation and correction strategies to accommodate these errors. This article presents the exploration, its results, the strategies we implemented, and the effect of these strategies on the performances of the tool. Though we did not observe a significative rise on average in the performances of the tool, we got much better results in some cases using our adaptation techniques.

1 Introduction

Semantic parsing of natural language is the task of extracting a formal meaning structure from a natural language sentence. Semantics of natural languages can be formalised in various ways, see for instance Bos (2011) and more recently Žabokrtský et al. (2020) for overviews; semantic parsing can be performed from any natural language into any of the semantic formalisms. One of these formalisms, Abstract Meaning Representations (AMR, Banarescu et al. (2013)) has been widely used in the context of deep semantic parsing of English and up to at least ten other languages, including French, German, Spanish, Italian, and Polish. There are two main types of approaches to multilingual AMRs: either the AMR graphs concepts are consistent with the target language (e.g. French concepts for French sentences), or the parsing results in an AMR graph with English concepts (Propbank-based). In this paper, we work in the scope of the latter approach. Machine semantic parsing of English has reached high-quality results, scoring over 83% Smatch score (Cai and Knight, 2013)[1] for the state of the art approaches (Yu and Gildea, 2022). In this context, we want to focus on the remaining 17%, and investigate both why the parser performs badly on these inputs and why the evaluation techniques would consider these parses as bad ones. We have limited our error explorations to languages we were familiar with; in particular, expert annotators familiar with Chinese should be involved in a follow-up study covering Chinese, for which a large amount of AMR annotation has been done. Machine AMR parsing works well, making the cases where it performs badly particularly interesting both linguistically and for deep learning studies. To make AMR parsing truly usable and reliable for real-life applications such as automatic summarization (Huang et al., 2022), question/answer generation (Deng et al., 2022), and neural machine translation (Li and Flanigan, 2022) we need to be able to trust it. We believe that this trust will come from a deep understanding of both our models and our data. The work presented in this article takes roots in explainability of artificial intelligence and computational linguistics. We conduct an error analysis and annotation exploration of the 50 worst examples from development corpora. We work in a multilingual context, on English (EN), French (FR), German (DE), Spanish (ES), Italian (IT), and Polish (PL). Our aim in this article is to share the error categories that we observed along with our attempts to remediate these errors, and the results of these attempts, in particular in terms of (non-significant) effects on the Smatch score. While our work constitutes a negative proof of concept, we still think it is an important contribution to share in the field to help to constitute a baseline for such adaptation techniques and encourage research and dialogues around them.

[1] A Python package is available at https://github.com/snowblink14/smatch/

Proceedings of the 15th International Conference on Computational Semantics pages 246–251
June 21–23, 2023. ©2023 Association for Computational Linguistics

2 AMR Parsing

AMR parsing was explicitly developed for English only. Its goal was to represent sentences through relations between predicates and their semantic arguments. These representations are now machine-generated and have been extended to at least ten other languages.

Abstract Meaning Representations In AMR, each sentence is represented with a rooted, directed, acyclic graph with labelled edges, where nodes are instances, concepts or literals, and edges are relations (figure 1). A single AMR graph can represent several natural language sentences as AMRs do not map words in a sentence to parts of the graph, but rather represent the semantic links that appear in the sentence or sequence of sentences. The goal of AMR representations is to abstract from syntactic constraints: sentences that have the same meaning but different formulations are represented with the same AMR. The representations are based on frames from PropBank (Kingsbury and Palmer, 2002), and the concepts and relations are either extracted from PropBank or English lemmas.

```
(h / hear-01              # "is a" relation (instantiation)
   :ARG0 (w / woman)                        # relation
   :ARG1 (c / cat
      :quant  2))                           # attribute
```

Figure 1: AMR graph for "the woman heard two cats".

Machine AMR Parsing The AMR parser we explore is based on AMRlib[2] for which the underlying language model T5 was changed for the multilingual MT5. We only used the T5/MT5 models since at the time we began our study they gave the best results (on English). AMRlib is based on a seq2seq model and outputs a "raw" AMR graph (without instance variables). The variables are inserted in a postprocessing step. If the raw graph contains too many errors (e.g. missing or additional quotes or parentheses), the postprocessing step loops through the raw graph until it has found a clean beginning. In this case, the final AMR graph lacks some instances and relations.

Data The training data for our parser is based on the English corpus of AMR 3.0 (LDC2020T02[3]). These corpora are mainly based on news reels. To obtain multilingual parsing, we trained our modified AMRlib using MT5 (instead of the monolingual T5) on data obtained by machine translation of English data to French, German, Spanish, Italian, and Polish. To reinforce the training, the parser was trained for each language on both corpora in English and in the target language. The AMR test corpus has been translated manually into German, Italian, Spanish, and Chinese (LDC2020T07). We evaluated on the first three of these and added the machine-translated versions for French and Polish since there is no manual translation of the sentences of the test corpora for these languages available. The results of our evaluation are listed in table 1.

IT	ES	DE	FR	PL
73.9	74.4	71.0	74.0	72.2

Table 1: Results for multilingual parsing evaluation.

3 Error Exploration

We performed an error analysis of the parser's outputs. We identified and implemented two strategies based on this analysis. Our results show improvements in the parsing results that are qualitatively interesting but quantitatively not significant enough.

Our analysis was done on the development corpus of AMR 3.0 (LDC2020T02), as we wanted to avoid introducing any bias in our study by using the test corpus. The sentences were machine translated into the given language, parsed using a model trained on the machine-translated training corpus, and then evaluated using the Smatch Python package against the gold development corpus. Then, the sentences were ranked by worst Smatch score, and the 50 first ones were annotated. Table 2 shows the Smatch scores for the first and 50th worst sentences, per language.

Smatch	EN	FR	DE	ES	IT	PL
worst	25.0	22.2	13.3	13.3	20.7	11.8
50th	60.9	49.6	46.8	48.9	49.5	47.4

Table 2: Worst and 50th worst Smatch per language.

3.1 Error Categories

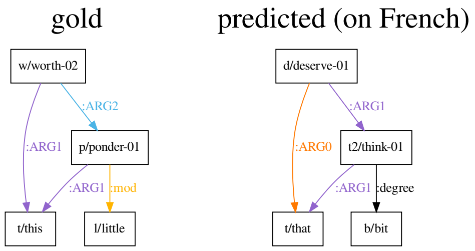
We identified seven categories of errors in the development data: (1) translation, (2) coordination, (3) input-based errors, (4) incomplete output, (5) reification, (6) errors in gold annotation, (7) other. These categories are listed in the order used for the exclusive annotation: if an error is annotated as a translation one, we did not try to annotate it further as belonging to another category as well.

[2] https://github.com/bjascob/amrlib

[3] https://catalog.ldc.upenn.edu/LDC2020T02

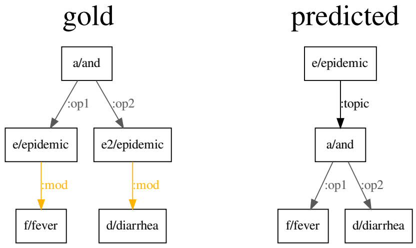
(1) Translation Translation-based errors have different origins: some of these come from wrong translations from English to the target language, which yields a bad parsing; others come from sentences for which the wordings in English and in the target language are structurally or lexically very different. Note that for DE, ES, and IT we used the official translations provided by LDC, so for these languages translation errors are of the second category. Some of the sentences contain technical terms which are badly or inexactly translated. The translation can also contain hallucinations because of the underlying seq2seq model, which adds parts that were not in the original sentence to the translated one. Lastly, the translation introduces synonymy in the AMR concepts that are used to build the graph. For instance, the English verb *break* was translated correctly into French *casser*, however, the AMR parser uses the concept *smash-01* instead of *break-01* found in the gold AMR graph.

Example: "This is worth pondering a little!" / "Cela mérite réflexion un peu !"



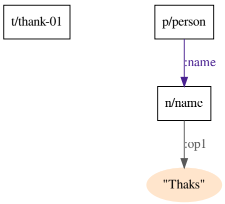
(2) Coordination This category corresponds to several subtypes of errors, including sentence coordination/multiple sentences, that can yield `multi-sentence` annotations, trigger `and`-concepts or not be annotated at all. First, we labelled here the errors that have to do with a bad parsing of conjunctions such as "and" or "but". Then, the ones that have to do more largely with sentence segmentation: sentences which were split incorrectly in two graphs linked with the `multi-sentence`-concept, or, on the contrary, two sentences which were merged using the `and`-concept.

Example: "There is an epidemic of fever and diarrhea."



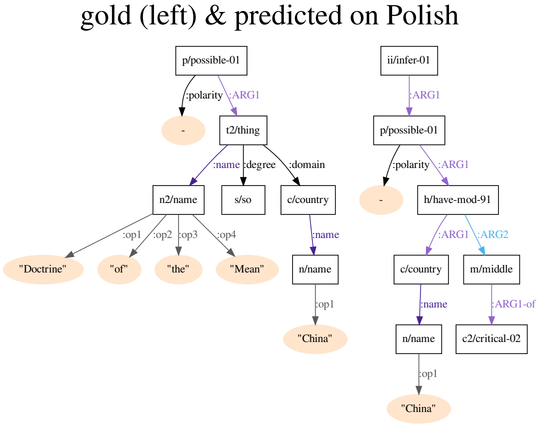
(3) Input There are errors in the input corpus, which can in turn yield errors in the output. In particular, some of the input sentences are too long for the model; when confronted with too long sentences, the model cuts off the sentence after the maximal input length has been reached, yielding incomplete AMR graphs. For Romance languages (FR, ES, IT), translated sentences are generally longer than the EN original ones. We also identified several cases in which the input sentence contains a misspelt word, or a word that is not in the model's vocabulary, or data in a format that is not identified by the model (ex: date), or named entities not recognized by the model as such.

Example: "Thaks."



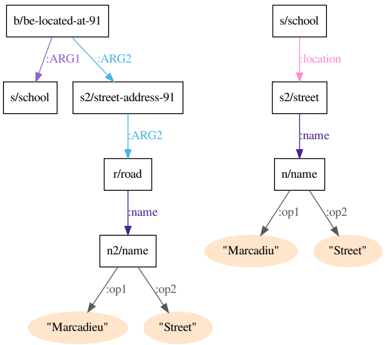
(4) Incomplete output Sometimes, we cannot identify any of the first three categories of errors, and the output AMR graph is still incomplete.

Example: "China can not be so 'Doctrine of the Mean' " / "Chiny nie mogą być więc 'Doktryną środka' "

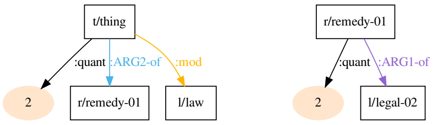


(5) Reification In the AMR 3.0 documentation, some relations (e.g. `:location`) can be reified into concepts (e.g. `be-located-at-91`), in order to be able to add a third argument. A reification without additional relations is considered semantically equivalent to the non-reified relation, thus in the gold annotations, both types of annotations are used. However, the standard evaluation script Smatch does not detect these equivalencies and produces a bad score.

Example: "the school is on marcadieu street"



(6) Gold In very few cases, the drop in the Smatch score comes from a mistake in the gold annotation and not from one in the parser's output.
Example: "Legally, there are two remedies."



(7) Other After establishing the 6 previous categories and conducting the annotation, we found other mistakes, which did not constitute a category on their own and could not be assigned to any of the previous categories. These errors have been annotated in this last category.
Example: "That was one hell of an over-reaction."/ "To była cholernie przesadna reakcja."

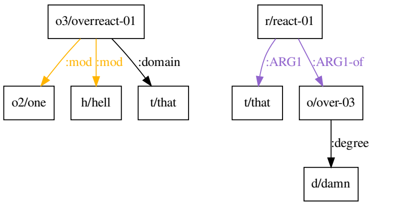


Table 3 shows the distribution of errors across these categories according to the annotation we performed on the 50 examples with the worst Smatch scores, for each language. Coordination is the most important error category for English and French; for the other languages, it is the second most important one after Translation. Then come the categories Input and Reification. Our annotation shows that the 2 other categories (not counting Other) are not significant enough with respect to the worst Smatch score examples.

3.2 Adaptation Strategies and Results

Errors in the input are difficult to correct, as we would risk overfitting and even worsening the situation when our parser would be confronted with new input mistakes outside the kind it would have been prepared to adapt to. Thus, we focused on coordination and reification phenomena for the development of our adaptation and correction strategies.

Reification To check whether the comparison between reified and non-reified relations impacts the evaluation, we wrote a script that reified every occurrence of reifiable relations in both the gold and system output of the development corpus and checked whether the Smatch score increases. However, the impact is minimal, instead of a Smatch score of 85.4, after reification we got 86.0 for EN.

Syntax-based Sentence Splitting Since we observed that long sentences are cut off when the number of tokens is bigger than the MT5 model can handle, we decided to test two sentence-splitting methods. We parsed all sentences of the development corpus with a dependency parser trained on Universal Dependency data[4]. In the first test, we focused on coordination by splitting sentences at the `parataxis` dependency relation (see black on white and white on black parts in figure 2). We then processed each partial sentence and merged the AMR graphs using the `multi-sentence` concept, e.g.:
original sentence: "The first stage splashed down in the Sea of Japan, the second stage crossed the main island of Japan."
partial sentences: (1) "The first stage splashed down in the Sea of Japan" (2) "the second stage crossed the main island of Japan."

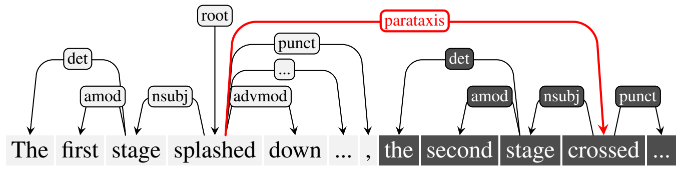


Figure 2: Dependency syntax tree for adjuncts (truncated).

For the second test, we extracted relative clauses (white on black in cf. figure 3) from the sentence (black on white in 3), replaced the relative pronoun ("who") by the head of the relative clause (to have a complete sentence). We then ran the AMR parser on each partial sentence and recombined the AMR graphs by merging the variables of the head of the relative clause (here "man") in both AMR graphs.
original sentence: "The man who saw the dog was afraid"
partial sentences: (1) "the man was afraid" (2) "the man saw the dog"

[4] https://universaldependencies.org

Language	Translation	Coord.	Input	Incomplete output	Reification	Gold error	Other
EN	n.a.	**21** (42%)	6 (12%)	0 (0%)	7 (14%)	3 (6%)	13 (26%)
FR	13 (26%)	**14** (28%)	1 (2%)	0 (0%)	10 (20%)	2 (4%)	10 (20%)
DE	**32** (64%)	7 (14%)	7 (14%)	3 (6%)	0 (0%)	0 (0%)	1 (2%)
ES	**25** (50%)	7 (14%)	7 (14%)	3 (6%)	0 (0%)	1 (2%)	7 (14%)
IT	**26** (52%)	12 (24%)	3 (6%)	5 (10%)	0 (0%)	0 (0%)	4 (8%)
PL	**22** (44%)	14 (28%)	1 (2%)	1 (2%)	5 (10%)	2 (4%)	5 (10%)

Table 3: Results of error annotations of the 50 first parses with the worst Smatch scores, per language.

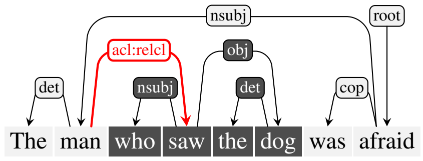


Figure 3: Dependency syntax tree for relative clause.

individual AMR graphs:

```
(v1 / fear-01              (v3 / see-01
  :ARG0 ( v2 / man))          :ARG0 ( v4 / man)
                              :ARG1 (v5 / dog))
```

joined graph (instances v2 and v4 merged into m):

```
(f / fear-01
    :ARG0 ( m / man
        :ARG0-of (s / see-01
            :ARG1 (d / dog))))
```

Even though for some complex sentences we got much better results with this splitting technique, for others this resulted in additional errors. On average the results in terms of Smatch score did not change.

4 Related Work

There are to our knowledge not many publications presenting systematic explorations of machine AMR parsing mistakes for the purpose of improving the explored tool. This observation might come from a publication bias, as a scientific community, we tend to publish positive results over negative ones. In Buljan et al. (2022), the authors present a discussion of methodological choices for diagnostic evaluation and error analysis in the context of four semantic parsers, two of which output AMR graphs. This article also explores one of the alternatives to Smatch, developed for several semantic representations of language (not only AMR), as part of the *meaning representation parsing* task. Damonte et al. (2017) presents another way of measuring the quality of automatic parses by using Smatch to compute more fine-grained metrics. Stemming from this work, Szubert et al. (2020) focuses on reentrancy phenomena in AMR graphs, categorizes their types, and shows results of experiments performed via an oracle correcting these errors, augmenting the overall parsing performance by 5%. Smatch is also being questioned in a multilingual context. In Wein and Schneider (2022), the authors argue for the necessity of a multilingual AMR evaluation metric and present a multilingual adaptation of S2match called XS2match. The work presented in our article is inspired by the previous work of the same authors (Wein and Schneider, 2021); in this work, they annotate translation divergences between a corpus of English and a corpus of Spanish data, grounding their annotation schema in AMR and labelling type and cause of divergences.

5 Discussion and Conclusion

As shown in table 3, the errors for the AMR graphs on languages other than English mostly concern the machine translation. Either the (English) input had typos (like “thaks” for “thanks”) or contained some named entities spelt in lowercase without any quotes which were translated literally into the target languages and not identifiable as named entities thereafter. The most frequent translation-related error is when a concept slightly differs from the concept in the gold. Even though we can consider these errors as minor, Smatch cannot identify close synonyms and classifies these differing concepts as plain errors.

The next steps for our research are twofold. On one hand, we will continue the diagnostic of our approach, in particular for languages other than English, by evaluating our parser using scores such as XS2match and exploring the errors that get the lower scores; conjointly, as translation issues were majoritary in our analyses, we will investigate how manual correction of translations can improve the parsing’s quality. On the other hand, we will investigate other approaches for our parser. Several categories of errors we diagnosed come from the seq2seq method and from the machine translation tools we use to produce the non-English corpora.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Johan Bos. 2011. A survey of computational semantics: Representation, inference and knowledge in wide-coverage text understanding. *Language and Linguistics Compass*, 5(6):336–366.

Maja Buljan, Joakim Nivre, Stephan Oepen, and Lilja Øvrelid. 2022. A tale of four parsers: methodological reflections on diagnostic evaluation and in-depth error analysis for meaning representation parsing. *Language Resources and Evaluation*, 56(4):1075–1102.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Zhenyun Deng, Yonghua Zhu, Yang Chen, Michael Witbrock, and Patricia Riddle. 2022. Interpretable amr-based question decomposition for multi-hop question answering. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, pages 4093–4099, Vienna, Austria.

Kuan-Hao Huang, Varun Iyer, Anoop Kumar, Sriram Venkatapathy, Kai-Wei Chang, and Aram Galstyan. 2022. Unsupervised syntactically controlled paraphrase generation with Abstract Meaning Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1547–1554, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From Tree-Bank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, Las Palmas, Canary Islands - Spain. European Language Resources Association.

Changmao Li and Jeffrey Flanigan. 2022. Improving neural machine translation with the Abstract Meaning Representation by combining graph and sequence transformers. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 12–21, Seattle, Washington. Association for Computational Linguistics.

Ida Szubert, Marco Damonte, Shay B Cohen, and Mark Steedman. 2020. The role of reentrancies in Abstract Meaning Representation parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2198–2207.

Shira Wein and Nathan Schneider. 2021. Classifying divergences in cross-lingual AMR pairs. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 56–65, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2022. Accounting for Language Effect in the Evaluation of Cross-lingual AMR Parsers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3824–3834, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Chen Yu and Daniel Gildea. 2022. Sequence-to-sequence AMR Parsing with Ancestor Information. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 571–577, Dublin, Ireland. Association for Computational Linguistics.

Zdeněk Žabokrtský, Daniel Zeman, and Magda Ševčíková. 2020. Sentence meaning representations across languages: what can we learn from existing frameworks? *Computational Linguistics*, 46(3):605–665.

Unsupervised Methods for Domain Specific Ambiguity Detection. The Case of German Physics Language

Vitor Lécio Lacerda Fontanella
Hochschule Hannover
Institute for Applied Data Science
Hannover, Germany
vitor-lecio.lacerda-fontanella
@hs-hannover.de

Christian Wartena
Hochschule Hannover
Institute for Applied Data Science
Hannover, Germany
christian.wartena
@hs-hannover.de

Abstract

Many terms used in physics have a different meaning or usage pattern in general language, constituting a learning barrier in physics teaching. The systematic identification of such terms is considered to be useful for science education as well as for terminology extraction. This article compares three methods based on vector semantics and a simple frequency-based baseline for automatically identifying terms used in general language with domain-specific use in physics. For evaluation, we use ambiguity scores from a survey among physicists and data about the number of term senses from Wiktionary. We show that the so-called Vector Initialization method obtains the best results.

1 Introduction

In science, it is common to refer to specific concepts using terms which are also used in everyday language but with a more specific or different meaning. At the same time, terms from science are assimilated into general language, often with a transformed meaning and use. Since these terms have a domain-specific use within science, they are a potential source of ambiguity, generating problems for successful communication and learning.

More specifically, in science education, it has been found that students' conceptions are often related to terms' general meanings non-congruent with the scientific ones (Itza-Ortiz et al., 2003; Clerk and Rutherford, 2000). In physics teaching, for example, words like *work*, *energy*, *momentum*, *impulse*, *power*, and *mass* have a narrower definition and a meaning that often differs entirely from the one used in everyday language (Itza-Ortiz et al., 2003; Song and Carheden, 2014). Song and Carheden (2014) argue that terms with multiple meanings are more difficult to learn, demanding further negotiation, expansion, and correct contextualization of their meanings. They also showed in a study with words from the chemistry teaching (e.g., *solution, polar,* and *compound*) that disassociating the scientific meaning from the one already acquired in everyday life is often hard. Moreover, Itza-Ortiz et al. (2003) show that students' ability to distinguish the different senses of a term correlates with test scores in the corresponding discipline.

By recognizing that terms with different meanings and uses in science and general language represent a learning barrier, their automatic identification within a discipline becomes a relevant task, supporting awareness of their use in teaching (Itza-Ortiz et al., 2003; Strömdahl, 2012; Liu et al., 2022), or even supporting specific teaching strategies for these cases (Vâlcea, 2019).

In Natural Language Processing (NLP), identifying semantic differences between domains (*Synchronic Lexical Semantic Change*) is similar to identifying lexical changes in time (*Diachronic Lexical Semantic Change*). In both cases, we can use properties of word embeddings to detect shifts in the relative positions in the embeddings space. The Synchronic Lexical Semantic Change has recently received attention in engineering requirements (Ferrari and Esuli, 2019; Jain et al., 2019; Mishra and Sharma, 2019) for the detection of potential sources of ambiguity. This task is also investigated in terminology extraction (Hätty et al., 2019), where statistical measures might not identify terms commonly used in specific and general contexts as part of a field's terminology.

Since word embeddings give a concise representation of a word's use (and, according to the distributional hypothesis, the meaning), many authors use them to study the domain-specific meaning of words. However, when computing word embeddings from two different corpora, we will end up with incomparable embedding spaces. The main differences in the proposed approaches deal with the solutions used to overcome this problem.

Proceedings of the 15th International Conference on Computational Semantics pages 252–257
June 21–23, 2023. ©2023 Association for Computational Linguistics

The present paper aims to compare three methods and a simple baseline for identifying general language words with a deviant meaning in physics. For the evaluation, we use two data sets, one obtained by collecting expert judgments on a small number of nouns and a larger data set derived from Wiktionary. To the best of our knowledge, this is the first evaluation of the automatic identification of general terms with a specific meaning in the science education domain.

2 Related Work

As mentioned above, we cannot immediately compare word embeddings derived from different corpora since the dimensions are randomly initialized, and the same dimensions in the two models will not correspond. The method **Vector Initialization** Kim et al. (2014) was the first to use neural network embeddings and solves this problem by initializing the embeddings' training from the previous corpus embeddings, and then comparing the position of the embedding before and after training. The hypothesis is that the embeddings' displacement after training reflect the word lexical change. This method was initially aimed at diachronic lexical change identification but can also be used for synchronic lexical change. Specific for the synchronic lexical change identification, Ferrari et al. (2017) and Mishra and Sharma (2019) use a variation of the Vector Initialization method to investigate lexical ambiguity between specific domains: they use marked target words before further training the embeddings. However, they need to select target words for the analysis based on their frequencies in both domain-specific corpora.

Other authors proposed to make the vector spaces comparable by defining a linear transformation between embedding spaces based on the solution of the **Orthogonal Procrustes** Problem (Hamilton et al., 2016; Jain et al., 2019; Schlechtweg et al., 2019). In the *Orthogonal Procrustes* analysis, a mapping matrix is determined using *Singular Value Decomposition* that rotates one of the vector spaces. The optimal alignment is the one that minimizes the distances between the embeddings of the same word in both vector spaces. Schlechtweg et al. (2019) added a pre-processing step to the procedure: the alignment of the mean center of the vector spaces before determining the mapping matrix.

The method proposed by Ferrari and Esuli (2019), named here as **Similar Words**, indirectly measures the similarity of embeddings from different spaces: they generate two lists of the most similar words using two vector spaces for a target word. Finally, they compute a rank correlation between the two lists to get an ambiguity score.

The methods above are based on static embeddings, like Word2Vec (Skip-Gram). This way, we obtain a general word representation in each context. Liu et al. (2022) use dynamic embeddings, using the average embedding of up to 1000 BERT embeddings in the corpus. They use a supervised regression model to identify domain-specific terms. Therefore we cannot compare their results to those from the unsupervised approaches discussed above. Martinc et al. (2020) also use averaged contextual embeddings. They use the same fine-tuned BERT embeddings for the general and domain-specific corpus but take the average for examples from each corpus separately. Thus the two averages obtained for each corpus are in the same embedding space and can be compared immediately.

Beyond qualitative evaluation, authors evaluate their method's results using manually created rankings (Ferrari and Esuli, 2019; Schlechtweg et al., 2019; Liu et al., 2022). Ferrari and Esuli (2019) only evaluated his method and ambiguity between specific domains, whereas Schlechtweg et al. (2019) systematically compared methods but, for synchronic lexical change, evaluated the methods only with the ranking of a few words used in general language and in the context of cooking. Liu et al. (2022) evaluated two methods (their own regression model and the unsupervised approach from Martinc et al. (2020)) on three different domains, generating lists of domain-specific terms. These lists were then evaluated manually, using precision as an evaluation measure; however, recall could not be assessed.

3 Experimental setup

In the following, we will present some (technical) information about our implementation of four methods for the identification of terms in general language with a specific sense in physics: **Vector Initialization**, **Orthogonal Procrustes**, **Similar Words** and **Relative Frequency**. The general idea behind the methods we compare was already described in section 2.

We generated three vector spaces using the Skip-Gram method from Gensim (Řehřek and Sojka,

Table 1: Overview of the corpora used.

corpus	Physics	deNews2020
Sentences	796 167	1 000 000
Tokens	15 506 365	17 624 256
Types	252 468	648 959

2011). The first (**model 1**) is obtained from a general language corpus. The second (**model 2**) follows the approach proposed by Kim et al. (2014): we train the model on the physics corpus, but initialize all vectors withe the embeddings from model 1. The third vector space (**model 3**) is obtained from the physics corpus alone. The embeddings have 200 dimensions, and the window size used in training was 5. In all cases we computed embeddings only for words occurring at least 10 times.

For the Vector Initialization method, we calculate the cosine value between the embeddings from model 1 and model 2. Words with the lowest cosine values will assumably have the most significant semantic displacement. For the Orthogonal Procrustes method, we align the vector spaces from models 1 and 3 after the pre-processing step proposed by Schlechtweg et al. (2019). After alignment, we also calculate the cosine values between the embeddings, expecting words with the same usage pattern in the two contexts to be more aligned after the procedure. Finally, for the **Similar Words** method, the ambiguity score will be determined by comparing the most similar words of the terms from models 1 and 3. To evaluate the methods, as a simple baseline, we also sort the words according to the relative frequency, assuming that all words frequently used in physics have a specific meaning in this domain.

4 Data

Identification of terms with domain-specific meaning requires two corpora, a *general language corpus* and a *domain-specific corpus*. The *general language corpus* should be, in principle, non-specific and large, representing, to some extent, everyday language. The domain-specific corpus (*physics corpus*) reflects the communicative context of our interest, namely physics teaching. For the present study, we use a German news corpus *denews2020* (Goldhahn et al., 2012). For the specific corpus, we use a corpus of German texts on physics, mostly high-level textbooks (Lacerda Fontanella et al., 2023). Table 1 gives some details on both corpora.

Table 2: Number of Words in Evaluation. The first column gives the number of words initially collected. The second column the number of the words from this initial collection that occur at least 10 times in both corpora.

	Total	denews ∩ Physics
Survey	48	48
Wiktio. Phy+	766	212
Wiktio. Phy-	135 660	9997

From the literature, we know a few terms that are considered problematic since they have a different meaning in physics than in general language. Such words are e.g., *Arbeit* (work, labor), *Energie* (energy), *Leistung* (power, performance), *Spannung* (tension), *Strom* (electricity, current), *Temperatur* (temperature), *Wärme* (heat, warmth) Strömdahl (2012); Rincke (2010).

However, for a more solid base for evaluation, we collected a set of 48 nouns, including the words mentioned above, along with more problematic and also unproblematic terms. In a survey, we asked participants to what extent, on a scale from 1 (same meaning) to 5 (totally different meaning), the meaning of a term differs in everyday use and the physical context. The ambiguity score for each term is the mean value of the answers in the survey. 14 subjects completed the survey. They were German native speakers with at least a master's degree in physics or physics teaching, including teachers, physicists, and science education researchers. We used survey data for evaluating the methods using the Pearson Correlation between the ambiguity score and the metric obtained for each word from the methods.

For a second experiment, we collected words from Wiktionary and counted how many senses for a word are marked as being specific for physics. E.g., a word like *Kraft* (force) appears with four senses in Wiktionary, one of them referring to physics. Since we compare senses that differ between physics and general language, we evaluated the ranking generated with each method, from potentially more to less ambiguous. We take from Wiktionary the binary information, no physics sense (0), and one or more sense in physics (1). Then, we calculate the area under the curve (AUC) to evaluate the ranking with this binary information. The total of words used (shown in Table 2) is much smaller than the number of words in the Wiktionary, given that the words must appear in both corpora

Table 3: Methods Evaluation.

Method	Survey (Correlation)	Wiktionary (AUC)
Vector Init.	**0.60**	**0.83**
Ortho. Proc.	0.52	0.71
Sim. Words	0.23	0.70
Rel. Freq.	0.58	0.68

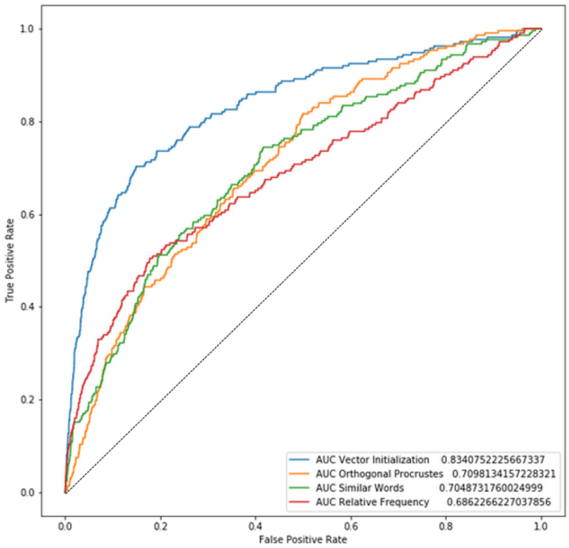


Figure 1: Area under the curve for the methods rankings and the data from Wiktionary.

and hold the minimum frequency requirement for computing an embedding.

5 Results

Table 3 shows the results of the quantitative evaluations of the methods. The method with the best correlation with the survey ambiguity score was the Vector Initialization with a moderate person correlation of 0.60, followed closely by Relative Frequency (0.58). The similar words method performs extremely bad on this task. For the ranking experiment, using much more words from Wiktionary, the Vector Initialization again is the best method, but now this method is clearly much better than the second best method. Relative frequency here is the worst method, though the differences between the other methods are quite small. The good results fron the relative frequency baseline are not surprising, since relative frequency between a specific and general corpus is considered to be an important criterion for terminology identification (Pazienza et al., 2005). However, Vector Initialization clearly outperforms the relative frequency.

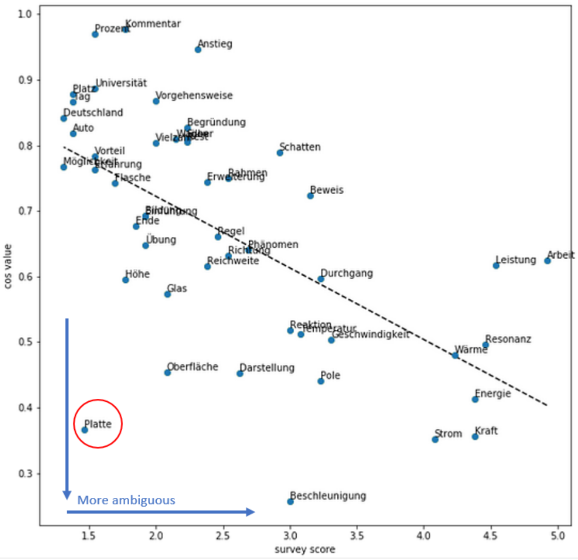


Figure 2: Correlation between **Vector Initialization** cosine values and survey score (Pearson=0.6). The term 'Platte' (board) is the most out the curve.

Finally we look at some qualitative results and examples from the experiments. Table 4 shows the first words in the ranking generated by each method. Here, the terms selected by the Vector Initialization methods make most sense, while especially the Orthogonal Procrustes and Similar Words method give a number of words that seem not to be related to physics at all.

Figure 2 displays the words of the survey with their averaged survey score and computed score. The participants did not perceive *Platte* (board) as ambiguous. Looking at some sentences with the term, we observe that this term is used in German very often referring to music albums. We believe the participants would hardly consider this sense while answering the survey.

Figure 3 shows the effect of the vector initialization method, displaying two terms on their original position and on their position after continued training on the TeCoPhy corpus. We see that the terms move in the direction of other typical physics terms.

6 Conclusion

Lexical ambiguity is a general challenge in communicative situations and an important issue in science education. Identifying domain specific ambiguity is needed to support the appropriate use of language in teaching and specific methodologies for terminology acquisition. In our research, the Vector Initialization method proved to be the most effective for identifying lexically ambiguous words.

Table 4: Twenty top lexical ambiguity candidates.

	Ortho Proc	Vec Init	Similar Words	Rel Freq
1	Heim	Kern	Zwilling	Ladung
2	Neo	Impuls	Ware	Flüssigkeit
3	Aussendung	Masse	Not	Masse
4	Kennzeichen	Winkel	Unterlage	Messung
5	Erhalt	Ladung	Amerikaner	Energie
6	Uniform	Beobachter	Paar	Wärme
7	Toleranz	Flüssigkeit	Lange	Definition
8	Ware	Einheit	Akt	Geschwindigkeit
9	Nerv	Körper	Verbreitung	Eigenschaft
10	Visum	Funktion	Verteilung	Winkel
11	Unterlage	Gas	Schnitt	Intensität
12	Grenzübergang	Feld	Kammer	Theorie
13	Bund	Volumen	Bestimmung	Körper
14	Rausch	Intensität	Zähler	Experiment
15	Hamilton	Ordnung	Produkt	Beschreibung
16	Spaltung	Feder	Ruf	Universum
17	Plus	Spannung	Siemens	Spektrum
18	Weiss	Strom	Signal	Gas
19	Profil	Summe	Brief	Spannung
20	Messe	Dimension	Fluss	Strömung

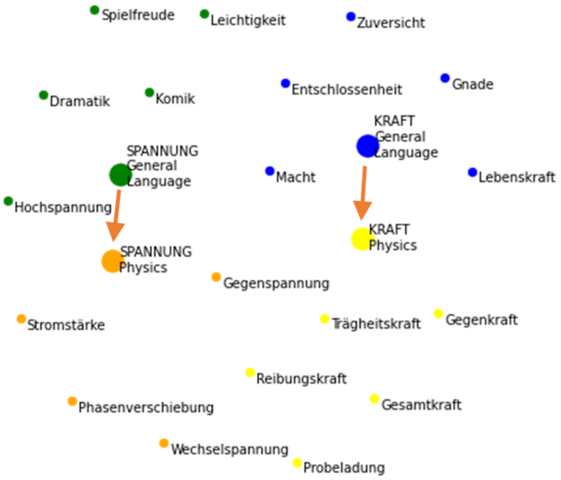


Figure 3: TSNE projection (Maaten and Hinton, 2008) of the most similar word embeddings for the terms Kraft (force) and Spannung (tension, stress, voltage) in model 1 and model 2.

This method achieved the highest Pearson correlation with the survey and AUC calculated with the Wiktionary data. However, in a different study by Schlechtweg et al. (2019), the Vector Initialization method performed poorly when ranking 22 target words. Such conflicting results may be due to the differences in the tasks involved, namely ranking target words versus automatically identifying lexical change.

Moreover, identifying lexical changes can aid in terminology extraction (Hätty et al., 2019), since it can uncover terms with a specialized meaning within a particular domain, despite being frequently used in general language. Such terms may not be found purely based on their frequency. Our research shows that the Vector Initialization method holds more promise than Orthogonal Procrustes as an additional technique for terminology extraction in science education.

A direction for future work is to bring the method to individual occurrences of a word: when finding an instance of an ambiguous word, we would like to be able to see whether the general or domain-specific meaning of the word is intended. This could finally help to see whether students use a word in the correct sense or whether they are misled by the everyday meaning of a specific term. For this purpose, we plan to explore methods based on contextual embeddings and evaluate their applicability to science education.

7 Acknowledgements

This research was funded by the Ministry of Science and Culture of Lower Saxony, Germany, within the PhD program "LernMINT: Data-assisted teaching in the STEAM subjects."

References

Douglas Clerk and Margaret Rutherford. 2000. Language as a confounding variable in the diagnosis of misconceptions. *International Journal of Science Education*, 22(7):703–717.

Alessio Ferrari, Beatrice Donati, and Stefania Gnesi. 2017. Detecting domain-specific ambiguities: An nlp approach based on wikipedia crawling and word embeddings. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 393–399.

Alessio Ferrari and Andrea Esuli. 2019. An nlp approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*, 26(3):559–598.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the 8th International Language Resources and Evaluation (LREC'12)*.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Anna Hätty, Dominik Schlechtweg, and Sabine Im Schulte Walde. 2019. Surel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the Eighth Joint Conference on*

*Lexical and Computational Semantics (*SEM 2019)*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Salomon F. Itza-Ortiz, N. Sanjay Rebello, Dean A. Zollman, and Manuel Rodriguez-Achach. 2003. The vocabulary of introductory physics and its implications for learning physics. *The Physics Teacher*, 41(6):330–336.

Vaibhav Jain, Ruchika Malhotra, Sanskar Jain, and Nishant Tanwar. 2019. Cross-domain ambiguity detection using linear transformation of word embedding spaces. *arXiv preprint arXiv:1910.12956*.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 61–65, Baltimore, MD, USA. Association for Computational Linguistics.

Vitor Lécio Lacerda Fontanella, Tom Bleckmann, Lukas Dieckhoff, Gunnar Friege, and Christian Wartena. 2023. TeCoPhy: A Text Corpus of German Physics Texts. In *Corpus Linguistics in the Digital Era: Genres, Registers and Domains (14th International Conference on Corpus Linguistics)*, pages 122–123, Oviedo, Spain. `https://cilc2023.wordpress.com/book-of-abstracts/`.

Yang Liu, Alan Medlar, and Dorota Głowacka. 2022. Lexical ambiguity detection in professional discourse. *Information Processing & Management*, 59(5):103000.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020. Leveraging contextual embeddings for detecting diachronic semantic shift. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4811–4819, Marseille, France. European Language Resources Association.

Siba Mishra and Arpit Sharma. 2019. On the use of word embeddings for identifying domain specific ambiguities in requirements. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 234–240. IEEE.

Maria Teresa Pazienza, Marco Pennacchiotti, and Fabio Massimo Zanzotto. 2005. Terminology extraction: An analysis of linguistic and statistical approaches. In Spiros Sirmakessis, editor, *Knowledge Mining*, volume 185 of *Studies in Fuzziness and Soft Computing*, pages 255–279. Springer-Verlag, Berlin/Heidelberg.

Radim Řehřek and Petr Sojka. 2011. Gensim – Statistical Semantics in Python.

Karsten Rincke. 2010. It's rather like learning a language: Development of talk and conceptual understanding in mechanics lessons. *International Journal of Science Education*, 33(2):229–258.

Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Im Schulte Walde. 2019. A wind of change: Detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Stroudsburg, PA, USA. Association for Computational Linguistics.

Youngjin Song and Shannon Carheden. 2014. Dual meaning vocabulary (dmv) words in learning chemistry. *Chem. Educ. Res. Pract.*, 15(2):128–141.

Helge R. Strömdahl. 2012. On discerning critical elements, relationships and shifts in attaining scientific terms: The challenge of polysemy/homonymy and reference. *Science & Education*, 21(1):55–85.

Cristina Vâlcea. 2019. Teaching technical polysemous words: Strategies and difficulties. In *ICERI2019 Proceedings*, ICERI Proceedings, pages 8388–8394. IATED.

“*Definition Modeling* : `To model definitions.`” Generating Definitions With Little to No Semantics

Vincent Segonne[*]
Université Grenoble Alpes
vincent.segonne
@univ-grenoble-alpes.fr

Timothee Mickus[*]
Helsinki University
timothee.mickus
@helsinki.fi

Abstract

Definition Modeling, the task of generating definitions, was first proposed as a means to evaluate the semantic quality of word embeddings—a coherent lexical semantic representations of a word in context should contain all the information necessary to generate its definition. The relative novelty of this task entails that we do not know which factors are actually relied upon by a Definition Modeling system. In this paper, we present evidence that the task may not involve as much semantics as one might expect: we show how an earlier model from the literature is both rather insensitive to semantic aspects such as explicit polysemy, as well as reliant on formal similarities between headwords and words occurring in its glosses, casting doubt on the validity of the task as a means to evaluate embeddings.

1 Introduction

Definition Modeling (Noraset et al., 2017, DefMod) is a recently introduced NLP task that focuses on generating a definition gloss given a term to be defined; most implementations rely on an example of usage as auxiliary input (Ni and Wang, 2017; Gadetsky et al., 2018; Mickus et al., 2019, a.o.). In the last few years, it has been the focus of more than a few research works: datasets have been proposed for languages ranging from Japanese (Huang et al., 2022) to Wolastoqey (Bear and Cook, 2021), and DefMod has even been the subject of a recent SemEval shared task (Mickus et al., 2022).

Practical applications for DefMod abound, from the generation of lexicographic data for low-resource languages (Bear and Cook, 2021), to computer-assisted language learning (Kong et al., 2022), creating learners’ dictionaries (Jiaxin et al., 2022), and from explaining slang (Ni and Wang, 2017) to clarifying scientific terminology (August et al., 2022). Yet, it was initially conceived by Noraset et al. (2017) as an evaluation task for word embeddings. If a word embedding is a coherent lexical semantic representation, then it ought to contain all the information necessary to produce a coherent gloss. Researchers have kept this semantic aspect firmly in mind: for instance, Bevilacqua et al. (2020) argue that DefMod provides a means to dispense word-sense disambiguation (WSD) applications from fixed, rigid sense inventories. More broadly, dictionaries in NLP are often used to capture some aspect of semantics.

This point bears closer inquiry. One may expect that writing definitions requires some knowledge of the meaning of the headword, but little has been done to confirm this expectation. Here, we focus on empirically verifying what impacts a model’s ability to generate valid definitions. As such, our interest lies mostly in examining what factors in the performance of a successful Definition Modeling system, rather than in the engineering aspects of DefMod implementations. We therefore re-purpose the fine-tuning protocol of Bevilacqua et al. (2020) to train a BART model (Lewis et al., 2020) to generate definitions, which we subsequently evaluate on infrequent words: As Bevilacqua et al. have extensively demonstrated the quality of their model on English data, it is suitable for our own endeavor.

Our findings suggest that it is possible to generate definition with little semantic knowledge: Our DefMod system, far from manipulating semantic information, mostly relies on identifying morphological exponents and tying them to lexicographic patterns. Semantic aspects of the headword—e.g., its polysemy or frequency—do not appear to weigh on model performances as captured through automatic metrics.

[*] Equal contribution.

Proceedings of the 15th International Conference on Computational Semantics pages 258–266
June 21–23, 2023. ©2023 Association for Computational Linguistics

2 Related Works

There is a broad domain of research that focuses on NLP solutions to lexicography problems and assessing how suitable they are (e.g., Kilgarriff et al., 2008; Frankenberg-Garcia, 2020; Frankenberg-Garcia et al., 2020; Hargraves, 2021). Conversely, many NLP works have used dictionaries to address semantic tasks, such as hypernym or synonym detection (Chodorow et al., 1985; Gaume et al., 2004) word-sense-disambiguation (Lesk, 1986; Muller et al., 2006; Segonne et al., 2019), compositional semantics (Zanzotto et al., 2010; Hill et al., 2016; Mickus et al., 2020), interpretability (Chang and Chen, 2019), representation learning (Bosc and Vincent, 2018; Tissier et al., 2017) or word retrieval (Siddique and Sufyan Beg, 2019, a.k.a. reverse dictionaries). We more narrowly concerned ourselves with definition modeling (Noraset et al., 2017), formulated as a sequence-to-sequence task (Ni and Wang, 2017; Gadetsky et al., 2018; Mickus et al., 2019). Our fine-tuning approach is borrowed from Bevilacqua et al. (2020); note that Huang et al. (2021) also employed a PLM (viz. T5, Raffel et al., 2020). We refer readers to Gardner et al. (2022) for a more thorough introduction.

3 Model & dataset

Datasets We retrieve data from DBnary (Sérasset, 2014),[1] an RDF-formatted dump of Wiktionary projects.[2] This source of data has previously been used to build DefMod datasets (Mickus et al., 2022), and is available in multiple languages—a desirable trait for future replication studies. More details are provided in Appendix B. For each term to be defined, we also tabulate its number of occurrences by tallying the number of string matches in a random subset of 5M documents from the deduplicated English Oscar corpus (Ortiz Suárez et al., 2019).

Headword frequency is worth focusing on, for at least two reasons. First, lexicographers are more likely to cover frequent words: dictionary-makers often espouse a data-driven approach to determine whether words should be included in general or specialized dictionaries (Hartmann, 1992; Frankenberg-Garcia et al., 2020);[3] Second, dictionary users should also be less familiar with rarer words—and likely require definitions. Hence, we set aside definitions where the headword has five or fewer occurrences in our Oscar subset for test purposes only, and further distinguish low-frequency headwords depending on whether they are attested in our Oscar sample. Remaining headwords are then split 80–10–10 between train, validation, and a second held out test set, so as to also measure models on identically distributed items. As such, we have three test sets, distinguished by the frequency of the headword in our Oscar sample: We note as $\# = 0$ the test set comprised of forms unattested in the sample; $\# \leq 5$ corresponds to headwords with five or fewer occurrences; $\# > 5$ matches with train set and validation set conditions.

Model The core of our methodology is borrowed from Bevilacqua et al. (2020): we fine-tune a generative pretrained language model, namely BART (Lewis et al., 2020), to produce an output gloss given an input example of usage, where the term to be defined is highlighted by means of special tokens `<define>` and `</define>`. We justify our adoption of their methodology by the fact that they report high results, through extensive NLG and WSD evaluation: as such, the approach they propose is representative of successful modern approaches to DefMod, and is suitable for a study such as ours. We refer the reader to their paper and Appendix A for details.

We expect DefMod systems to be sensitive to the variety of examples of usages and number of target glosses: more examples of usage should lead to higher performances, whereas not exposing the model to polysemy should be detrimental. This can be tested by down-sampling the training set, so as to select one gloss per headword (1G or ∀G) and/or one example of usage per gloss (1E or ∀E). This leads us to defining four related models: ∀G∀E, ∀G1E, 1G∀E, and 1G1E. [4]

4 Impact of frequency, polysemy and contextual diversity

Corresponding results in terms of BLEU, shown in Table 1, are in line with similar results on un-

[1] http://kaiko.getalp.org/about-dbnary/

[2] http://wiktionary.org/

[3] Lack of corpus evidence may also be reason enough for lexicographers to ignore rarer words (Hanks, 2009, 2012). Dictionaries often rely on usage data to select entries (e.g., https://www.merriam-webster.com/help/faq-words-into-dictionary)

[4] Using this notation, 1G∀E means that, for a given headword, we randomly selected one gloss with all its corresponding examples; for ∀G1E, all glosses were considered but with only one randomly selected example for each.

Config	**Split**			
	Val.	$\# > 5$	$\# \leq 5$	$\# = 0$
∀G∀E	9.07	9.13	11.15	10.85
∀G1E	9.06	9.10	11.11	10.94
1G∀E	8.29	8.32	10.69	10.53
1G1E	8.49	8.53	11.06	10.87

Table 1: Average BLEU performances on held-out sets. Averaged on 5 runs; std. dev. $< \pm 0.001$ always.

seen headwords e.g. in Bevilacqua et al. (2020).[5] They also highlight a strikingly consistent behavior across all four configurations: Mann-Whitney U tests stress that we do not observe lower performances for rarer words, as one would naively expect, except in few cases (∀G∀E, ∀G1E and 1G1E models, when comparing unattested and rare headwords) with relatively high p-values given the sample sizes ($p > 0.01$ always).

Another way to stress the lack of effect related to explicit polysemy or contextual diversity consists in correlating BLEU scores across models: Comparing the BLEU scores obtained by one model (say the ∀G∀E) to those of another model (e.g., the 1G1E model) indicates whether they behave differently or whether BLEU scores are distributed in roughly the same fashion. We systematically observe very high Pearson coefficients ($0.82 < r < 0.90$). In other words, definitions that are poorly handled in any model will in all likelihood be poorly handled in all other models, and definitions that are easy for any single model will be easy for all other models. We provide a breakdown per split and per model in Appendix C, Table 6.

5 Digging further: manual evaluation

To better understand model behavior, we sample 50 outputs of the ∀G∀E model, per BLEU quartile, for the validation split and our three test splits. We then annotate these 800 items as follows.

5.1 Annotation scheme

Sample items for all annotations are provided in Table 2.

Fluency (FL) measures if the output is free of grammar or commonsense mistakes. For instance, "`(intransitive) To go too far; to go too far.`" is rated with a FL of 1, and "`(architecture) A belfry`" is rated 5.

Factuality (FA) consists in ensuring that generated glosses contain only and all the facts relevant to the target senses. Hence the output "`Not stained.`" generated for the headword *unsatined* is annotated with a FA of 1, whereas the output "`A small flag.`" for the headword *flaglet* is rated with a FA of 5.

PoS-appropriateness (PA) A PoS-appropriate output defines its headwords using a phrase that match its part of speech—e.g., defining adjective with adjectival phrases and nouns with noun phrases. As such, the adjective headword *fried* yields the PoS-inappropriate "`(transitive) To cook (something) in a frying pan.`", while the production for the verb *unsubstantiate*, viz. "`(intransitive) To make unsubstantiated claims.`" has a PA of 1.

Pattern-based construction (PB) An output is said to display a pattern-based construction whenever it contains only words that are semantically tenuous or morphologically related to the headword. The headword *clacky* thus yield the PB output "`Resembling or characteristic of clacking.`", and the headword *fare* yields the non-PB production "`(intransitive) To do well or poorly.`"[6]

5.2 Results of the manual evaluation

When looking at all 800 annotations, we find that outputs tend to be fluent (average FL of 4.37) and overwhelmingly PoS-appropriate (95%). They frequently involve patterns (36.5% of PBs) which often involve a straight copy of the headword (10% of all productions). On the other hand, factuality is lacking (average FL of 2.69).

Inter-annotator agreement To quantify how consensual our annotations are, we randomly sample 200 items for dual annotation. Results in Table 3 highlight that, while the two annotators have different sensibilities as to the magnitude of the mistakes in FL and FA (as shown by the low κ), relative judgments on fluency and factuality are

[5] We observed similar patterns with most widely-used automatic NLG metrics, and focus on BLEU in the present article for brievity. Nonetheless, see e.g. Roy et al. (2021) for a discussion of the limitations of this metric.

[6] FA and FL are on a 5-point scale, PA and PB are binary.

POS	example	target	hypothesis	annotation	
verb	Thus was th' accomplish'd squire **endued** / With gifts and knowledge per'lous shrewd .	To invest (someone) with a given quality, property etc.; to endow.	(obsolete, transitive) To supply; to supply; to supply.	FL	1
noun	The wealth of those societies in which the capitalist **mode of production** prevails, presents itself as "an immense accumulation of commodities," its unit being a single commodity.	(Marxism) A combination of productive forces such as labour power and means of production, and social and technical relations of production such as property, power, laws and regulations, etc.	(economics) The economic system in which the production of goods and services is based on the production of commodities.	FL	3
noun	Often, though, a **suki** to the chest will cause the sword to become lodged between bone and cartilage making it very difficult to quickly remove.	(martial arts) An opening to the enemy; a weak spot that provides an advantage for one's opponent.	(historical) A blow made by a sword to the chest.	FL	5
verb	[...] the higher of them can never **abut on** a pyknon in the case envisaged here, where the tone is introduced to disjoin the tetrachords.	(transitive) To border on.	(music, transitive) To play (a note) at the same time.	FA	1
noun	"**Kurkuls** are our enemy," he shouted, "and we must exterminate them as a social class. [...]	(historical) A rich or supposedly rich peasant, targeted during Soviet collectivization, especially in the context of Ukraine or Ukrainians; kulak.	(rare) A kurkul.	FA	3
adj.	And its success or failure is likely to tell whether talents [...] make new fortunes from the **nonentertainment** companies that are looking to Hollywood.	Not of or pertaining to entertainment.	Not entertainment.	FA	5
adj.	an **arrant** knave, arrant nonsense	(chiefly, with a negative connotation, dated) Complete; downright; utter.	(obsolete, transitive) To make up; to invent; to invent.	PA	0
noun	[...] Another is to ban planned **obsolescence**, so manufacturers can't create products that are designed to fail .	(uncountable) The state of being obsolete—no longer in use; gone into disuse; disused or neglected.	The state or condition of being obsolescent.	PA	1
noun	A canister of flour from the kitchen had been thrown at the looking-glass and lay like trampled snow over the remains of a decent blue suit with the **lining** ripped out which lay on top of the ruin of a plastic wardrobe.	A covering for the inside surface of something.	The outer layer of a garment.	PB	0
adj.	an **obliquangular** triangle	(archaic, geometry) Formed of oblique angles.	(geometry) Of or pertaining to an oblique angle	PB	1

Table 2: Example of annotated items. Word being defined in **bold** in the example of usage.

Trait	Cohen κ	Spearman ρ	Pearson r
FL	0.405	0.633	0.693
FA	0.374	0.741	0.768
PA	1.000	1.000	1.000
PB	0.780	0.784	0.784

Table 3: Manual annotations, inter-annotator agreement. Pearson r were computed on z-normalized annotations.

consistent (as shown by ρ and r). Hence, we z-normalize FA and FL in the rest of this analysis.

Effects of patterns Mann-Whitney U-tests on FA and FL annotations show that non-pattern-based outputs are statistically rated with lower FL ($p < 3 \cdot 10^{-6}$, common language effect size $f = 42.3\%$) and lower FA ($p < 2 \cdot 10^{-9}$, $f = 37.7\%$) than pattern-based definitions, despite no significant difference in BLEU scores ($p = 0.262$). On the other hand, BLEU scores are correlated with FL and FA ratings (Spearman $\rho = 0.094$ and $\rho = 0.276$ respectively). In sum, the morphologically complex nature of a headword drives much of the behavior of our DefMod system. While BLEU captures some crucial aspects we expect to be assessed in DefMod, it is still impervious to this key factor.

To further confirm that patterns are indeed crucial to a DefMod system's performance, we train a model on data where headwords have been removed from examples of usages, keeping the surrounding control tokens. This in effect creates a 2-token sentinel for which the decoder must gener-

	Split		
Val.	$\# > 5$	$\# \leq 5$	$\# = 0$
5.60	5.72	5.11	4.85

Table 4: Performances with headword ablation

ate a gloss, and deprives the model of information about headword form. BLEU scores drastically drop with this ablated train set, as shown in Table 4. We also find unattested headwords yielding statistically lower BLEUs than rare headwords, which in turn yield lower BLEUs than the other two splits (Mann-Whitney U tests, $p < 10^{-7}$).

Frequency and polysemy We now return to polysemy and word frequency. We consider as an indicator of word polysemy the number of definitions for that headword present in our corpus, whereas we rely on our Oscar sample to derive frequency counts. Frequency and definition counts appear to be highly correlated (Spearman $\rho = 0.406$), and both also anti-correlate with PB ($\rho = -0.1143$ and $\rho = -0.111$ respectively), i.e., rare, monosemous words are defined by the model with patterns (that is, they are likely morphologically complex). We also observe an anticorrelation between FL and definition count (Spearman $\rho = -0.105$), which could be explained by the fact that patterns tend to yield more fluent outputs, as we just saw—however, as we do not observe a correlation between frequency and FL, the interaction between FL and polysemy (as measured by definition count) is likely not so straightforward.[7] Finally, BLEU scores do not correlate with word frequency nor definition counts, which strengthens our claim that this DefMod system makes limited use semantic information to generate glosses—if at all.

	FL	FA
BertScore (Zhang et al., 2020)	0.16	0.37
BLEU (Papineni et al., 2002)	0.09	0.28
chrF (Popović, 2015)	–	0.35
GLEU (Wu et al., 2016)	–	0.29
METEOR (Banerjee and Lavie, 2005)	–	0.31
ROUGE-L (Lin, 2004)	–	0.37
TER (Snover et al., 2006)	−0.10	−0.27

Table 5: Correlation of FA and FL with NLG metrics. Missing values correspond to insignificant coefficients.

Alternatives to BLEU These annotations leave one question unanswered: is BLEU an adequate means of measuring DefMod productions? In Table 5, we compare the Spearman correlation coefficient of various NLG metrics with our FA and FL annotations. Most NLG metrics do not correlate with fluency ratings: we posit this is due to the overwhelming majority of highly fluent productions in our sample. As for BLEU, it doesn't produce the highest (anti-)correlations—they are instead attested with BertScore for FL and ROUGE-L for FA. Lastly, Mann-Whitney U tests comparing metrics with respect to PB annotations indicate that most of these are not sensitive to the presence or absence of a pattern, with the exception of chrF ($f = 0.43$) and TER ($f = 0.42$). In all, our annotated sample suggests that most NLG metrics appear to display a behavior similar to BLEU: they capture factuality to some extent—but not the importance of patterns.

6 Conclusions

In this work, we have presented how an earlier Definition Modeling system was able to achieve reasonable performances and produce fluent outputs, although the factual validity leave much to be desired. This behavior is almost entirely due to morphologically complex headwords, for which the model is often able to derive reasonable glosses by decomposing the headword into a base and an exponent, and mapping the exponent to one of a limited set of lexicographic patterns. The model we studied seems more sensitive to formal traits than to explicit accounts of polysemy. There are numerous limitations to this work: we focused on one specific fine-tuning approach for one specific English PLM. Nonetheless, we have shown that models can achieve reasonable performances on DefMod without relying on semantics, casting doubt on the task's usefulness for word embedding evaluation, as initially suggested by Noraset et al. (2017)

In other words: using lexicographic data as inputs for an NLP model does not ensure that it will pick up on the semantic aspects contained therein.

Acknowledgments

We thank IWCS and $*$SEM reviewers for their insightful comments on this work.

[7] Neither do we observe no correlation with FA and PA.

This work is part of the FoTran project, funded by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation program (agreement № 771113). We also thank the CSC-IT Center for Science Ltd., for computational resources.

References

Tal August, Katharina Reinecke, and Noah A. Smith. 2022. Generating scientific definitions with controllable complexity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317, Dublin, Ireland. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Diego Bear and Paul Cook. 2021. Cross-lingual wolastoqey-English definition modelling. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 138–146, Held Online. INCOMA Ltd.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Tom Bosc and Pascal Vincent. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532, Brussels, Belgium. Association for Computational Linguistics.

Ting-Yun Chang and Yun-Nung Chen. 2019. What does this word mean? explaining contextualized embeddings with natural language definition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6064–6070, Hong Kong, China. Association for Computational Linguistics.

Martin S. Chodorow, Roy J. Byrd, and George E. Heidorn. 1985. Extracting semantic hierarchies from a large on-line dictionary. In *23rd Annual Meeting of the Association for Computational Linguistics*, pages 299–304, Chicago, Illinois, USA. Association for Computational Linguistics.

Ana Frankenberg-Garcia. 2020. Combining user needs, lexicographic data and digital writing environments. *Language Teaching*, 53(1):29–43.

Ana Frankenberg-Garcia, Geraint Paul Rees, and Robert Lew. 2020. Slipping Through the Cracks in e-Lexicography. *International Journal of Lexicography*, 34(2):206–234.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271, Melbourne, Australia. Association for Computational Linguistics.

Noah Gardner, Hafiz Khan, and Chih-Cheng Hung. 2022. Definition modeling: literature review and dataset analysis. *Applied Computing and Intelligence*, 2(1):83–98.

Bruno Gaume, Nabil Hathout, and Philippe Muller. 2004. Word sense disambiguation using a dictionary for sense similarity measure. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1194–1200, Geneva, Switzerland. COLING.

Patrick Hanks. 2009. The impact of corpora on dictionaries. In *Contemporary corpus linguistics*, chapter 13, pages 214–236. Continuum London.

Patrick Hanks. 2012. The Corpus Revolution in Lexicography. *International Journal of Lexicography*, 25(4):398–436.

Orin Hargraves. 2021. Lexicography in the post-dictionary world. *Dictionaries*, 42(2):119–129.

R. R. K. Hartmann. 1992. Lexicography, with particular reference to english learners' dictionaries. *Language Teaching*, 25(3):151–159.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Han Huang, Tomoyuki Kajiwara, and Yuki Arase. 2021. Definition modelling for appropriate specificity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2499–2509, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Han Huang, Tomoyuki Kajiwara, and Yuki Arase. 2022. JADE: Corpus for Japanese definition modelling. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6884–6888, Marseille, France. European Language Resources Association.

Yuan Jiaxin, Kong Cunliang, Xie Chenhui, Yang Liner, and Yang Erhong. 2022. COMPILING: A benchmark dataset for Chinese complexity controllable definition generation. In *Proceedings of the 21st Chinese National Conference on Computational Linguistics*, pages 921–931, Nanchang, China. Chinese Information Processing Society of China.

Adam Kilgarriff, Miloš Husák, Katy McAdam, Michael Rundell, and Pavel Rychlý. 2008. GDEX: Automatically finding good dictionary examples in a corpus. In *Proceedings of the 13th EURALEX International Congress*, pages 425–432, Barcelona, Spain. Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra.

Cunliang Kong, Yun Chen, Hengyuan Zhang, Liner Yang, and Erhong Yang. 2022. Multitasking framework for unsupervised simple definition generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5934–5943, Dublin, Ireland. Association for Computational Linguistics.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, page 24–26, New York, NY, USA. Association for Computing Machinery.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Timothee Mickus, Timothée Bernard, and Denis Paperno. 2020. What meaning-form correlation has to compose with: A study of MFC on artificial and natural language. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3737–3749, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timothee Mickus, Denis Paperno, and Matthieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland. Linköping University Electronic Press.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Philippe Muller, Nabil Hathout, and Bruno Gaume. 2006. Synonym extraction using a semantic distance on a dictionary. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 65–72, New York City. Association for Computational Linguistics.

Ke Ni and William Yang Wang. 2017. Learning to explain non-standard English words and phrases. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 413–417, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Thanapon Noraset, Chen Liang, Lawrence Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *AAAI*.

Pedro Javier Ortiz Suárez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut f"ur Deutsche Sprache.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Devjeet Roy, Sarah Fakhoury, and Venera Arnaoudova. 2021. Reassessing automatic evaluation metrics for code summarization tasks. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2021, page 1105–1116, New York, NY, USA. Association for Computing Machinery.

Vincent Segonne, Marie Candito, and Benoît Crabbé. 2019. Using Wiktionary as a resource for WSD : the case of French verbs. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 259–270, Gothenburg, Sweden. Association for Computational Linguistics.

Gilles Sérasset. 2014. DBnary: Wiktionary as a Lemon-Based Multilingual Lexical Resource in RDF. *Semantic Web Journal - Special issue on Multilingual Linked Open Data*, pages –. To appear.

Bushra Siddique and Mirza Mohd Sufyan Beg. 2019. A review of reverse dictionary: Finding words from concept description. In *Next Generation Computing Technologies on Computational Intelligence*, pages 128–139, Singapore. Springer Singapore.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2017. Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Copenhagen, Denmark. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1263–1271, Beijing, China. Coling 2010 Organizing Committee.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Hyperparameters

Models are implemented in fairseq (Ott et al., 2019). We used the `bart_large` model and followed the instructions on the github repository for finetuning BART on the summary task.[8] We used the same parameters except for the learning rate, which after some experiments, was set to $5 \cdot 10^{-6}$. For every configuration (∀G∀E, ∀G1E,1G∀E, 1G1E) we kept the model with the best loss on the validation dataset.

B Data preprocessing

In the present work, we retrieve definition glosses (i) associated with an example of usage and (ii) where the term to be defined is tagged as a noun, adjective, verb, adverb or proper noun. Like Bevilacqua et al., we also consider MWEs as potential terms to define.

To highlight a headword within an example of usage, the approach of Bevilacqua et al. (2020) consists in surrounding them with learned task-specific control tokens. We therefore parse example of usages using SpaCy[9] to retrieve the first sequence of tokens whose lemmas match with the lemmas of the term to be defined.

The BART model we fine-tune on DefMod has been pretrained on OpenWebText, which contains some pages retrieved from Wiktionary. We preemptively remove these pages from all dataset splits, so as to ensure there is no overlap between pre-train, train and test data.

Frequencies are tabulated on a case-folded, whitespace-normalized subset of the Oscar corpus. In practice, we extract the number of hard string matches of each headword preprended and appended with word boundaries.

C BLEU scores correlations

In Table 6, we display how similar are the behaviors on different models across splits. Each sub-table corresponds to a different split, and pits all combinations of models. For instance, the last cell in the second row of sub-Table 6c indicates that to the Pearson correlation between the ∀G1E and the 1G1E on the $\# \leq 5$ test split is above 88.4%. The crucial fact that emerges from these tables is the distribution of BLEU is very similar across all models we tested—which entails that explicit polysemy or contextual diversity do not weight on performances, as measured through BLEU scores.

	∀G1E	**1G∀E**	**1G1E**
∀G∀E	0.89	0.87	0.85
∀G1E		0.84	0.87
1G∀E			0.88

(a) Validation split

	∀G1E	**1G∀E**	**1G1E**
∀G∀E	0.89	0.86	0.85
∀G1E		0.83	0.87
1G∀E			0.87

(b) Test $\# > 5$ split

	∀G1E	**1G∀E**	**1G1E**
∀G∀E	0.88	0.89	0.86
∀G1E		0.85	0.88
1G∀E			0.88

(c) Test $\# \leq 5$ split

	∀G1E	**1G∀E**	**1G1E**
∀G∀E	0.88	0.88	0.85
∀G1E		0.86	0.88
1G∀E			0.88

(d) Test $\# = 0$ split

Table 6: BLEU scores correlations (Pearson r)

[8] https://github.com/facebookresearch/fairseq/blob/main/examples/bart/README.summarization.md

[9] https://spacy.io/

SMARAGD: Learning SMatch for Accurate and Rapid Approximate Graph Distance

Juri Opitz   Philipp Meier   Anette Frank
Dept. of Computational Linguistics
Heidelberg University
69120 Heidelberg
{opitz,meier,frank}@cl.uni-heidelberg.de

Abstract

The similarity of graph structures, such as Meaning Representations (MRs), is often assessed via structural matching algorithms, such as SMATCH (Cai and Knight, 2013). However, SMATCH involves a combinatorial problem that suffers from NP-completeness, making large-scale applications, e.g., graph clustering or search, infeasible. To alleviate this issue, we learn SMARAGD: Semantic Match for Accurate and Rapid Approximate Graph Distance. We show the potential of neural networks to approximate SMATCH scores, i) in linear time using a machine translation framework to predict alignments, or ii) in constant time using a Siamese CNN to directly predict SMATCH scores. We show that the approximation error can be substantially reduced through data augmentation and graph anonymization.

1 Introduction

Semantic graphs such as Meaning Representation (AMR) are directed, rooted and acyclic, and labeled. For instance, in AMR (Banarescu et al., 2013) labels indicate the events and entities of a sentence, and structures capture semantic roles and other key semantics such as coreference.

Often, pairs of MRs need to be studied, using MR metrics. Classically, MRs are compared to assess Inter Annotator Agreement in SemBanking or for the purpose of parser evaluation, typically using the *structural* SMATCH metric (Cai and Knight, 2013; Opitz, 2023). Going beyond these applications, researchers have leveraged SMATCH-based MR metrics for NLG evaluation (Opitz and Frank, 2021; Manning and Schneider, 2021), for re-inforcing AMR parsers (Naseem et al., 2019), as a basis for a COVID-19 semantics-based search engine (Bonial et al., 2020), comparison of cross-lingual AMR (Uhrig et al., 2021; Wein et al., 2022), and fine-grained argument similarity assessment (Opitz et al., 2021b). Many of these extended scenarios greatly profit from a *quick similarity computation*. Also, additional future applications can be anticipated that require fast metric inference: e.g., corpus linguists who want to find instantiations of abstract semantic patterns in a large corpus.

But graph metrics typically suffer from a high time complexity: Computation of SMATCH is NP-hard (Nagarajan and Sviridenko, 2009), and it can take more than a minute to compare some 1,000 AMR pairs (Song and Gildea, 2019). To understand that this can become problematic in many setups, consider a hypothetical user who desires exploring a (small) AMR-parsed corpus with only $n = 1,000$ instances via clustering. The (symmetric) SMATCH needs to be executed over $(n^2 - n)/2 = 499,500$ pairs, resulting in a total time of more than 6 hours.

This high time complexity is a well-known bottleneck and negatively impacts AMR evaluation time (Song and Gildea, 2019), as well as parsing efficency of approaches involving re-inforcement learning (Naseem et al., 2019) or graph ensembling (Hoang et al., 2021), where the SMATCH metric is executed with high frequency. Furthermore, given recent interest into larger meaning representations that cover multiple sentences, such as multi-sentence AMR (O'Gorman et al., 2018), dialogue AMR (Bonial et al., 2021) or discourse representation structures (Kamp, 1981; van Noord et al., 2018), we anticipate that this problem will become more pressing in the future.

Testing ways to mitigate these issues, we propose a method that learns to match semantic graphs from a teacher SMATCH, and show that this can reduce AMR clustering time from hours to seconds, with only little expected loss in accuracy.

Our contributions are:

1. We explore three different neural approaches to synthesize the combinatorial graph metric

Proceedings of the 15th International Conference on Computational Semantics pages 267–274
June 21–23, 2023. ©2023 Association for Computational Linguistics

SMATCH from scratch.

2. We show that we can approximate SMATCH up to a small error, by leveraging novel data augmentation tricks.

Our code is available at: `https://github.com/PhMeier/Smaragd/`.

2 Related work

Other metrics for MR similarity Recently, researchers have proposed AMR metrics beyond SMATCH. We can distinguish two lines of work: i) metrics aiming at extreme efficiency by skipping the alignment and extracting graph parts via breadth-first traversal (Song and Gildea, 2019; Anchiêta et al., 2019). ii) Weisfeiler-Leman graph metrics that aim to reflect human similarity ratings (Opitz et al., 2021a). Opitz et al. (2020) make an argument for the importance of graph alignment.

Algorithm synthesis Neural networks have been studied for solving other problems efficiently. Examples range from sorting numbers (Graves et al., 2014; Neelakantan et al., 2016) to solving elaborated tasks such as symbolic integration (Lample and Charton, 2019), the famous traveling salesman problem (Gambardella and Dorigo, 1995; Budinich, 1996; Bello et al., 2016; Zhang et al., 2021), and computer programs (Balog et al., 2016; Nye et al., 2020; Chen et al., 2021). The 'long-range arena' benchmark (Tay et al., 2021) includes algorithm synthesizing tasks, such as 'listOps' (learning to calculate), or Xpath (tracing a squiggly line), which prove challenging even for SOTA architectures. Since structural graph matching with SMATCH constitutes a very hard combinatorial problem, investigating efficient neural approximations seems an interesting challenge in general – beyond the use-case of rapid graph distance calculation.

3 Learning NP-hard graph alignment

The SMATCH metric measures the structural overlap of two graphs. We i) compute an alignment between variable nodes of graphs and ii) assess triple matches based on the provided alignment. Formally, we start with two graphs a and b with variable nodes $X = (x_1, ...x_n)$ and $Y = (y_1...y_m)$. The goal is then to find an optimal *alignment*

$$map^\star : X \rightarrow Y, \qquad (1)$$

searching for a map that maximizes the number of *triple matches* for the two graphs. For instance, assume two AMR triples `(x, ARG0, y)` $\in \mathcal{G}$ and `(u, ARG0, v)` $\in \mathcal{G}'$. If $x = u$ and $y = v$, we count *one* triple match. Finally:

$$SMATCH = \max_{map} score(a, b, map) \qquad (2)$$

Researchers typically use a harmonic mean based overlap $score = F1 = 2PR/(P + R)$, where $P = |triples(a) \cap triples(b)|/|triples(a)$ and $R = |triples(a) \cap triples(b)|/|triples(b|$.

3.1 Setup

Experimental data creation We create the data for our experiments as follows: 1. We parse 59,255 sentences of the LDC2020T02 AMR dataset with a parser (Lyu and Titov, 2018) to obtain graphs that can be aligned to reference graphs; 2. For every parallel graph pair (a, b), we use SMATCH (ORACLE) to compute an F1 score s and the alignment $map^\star$, yielding an extended data tuple $(a, b, s, map^\star)$ We shuffle the data and split it into training, development and test set (56255-1500-1500).

Objective and approach The task is to reproduce the teacher ORACLE as precisely as possible. We design and test three different approaches. The first is indirect, in that it predicts the alignment, from which we compute the score. The second directly predicts the scores. The third approach enhances the second, to make it even more efficient.

3.2 Synthesis option I: Alignment learning

Here, we aim to learn the alignment itself (Eq. 1) with an NMT model, as illustrated in Figure 1. For the input, we linearize the two AMRs and concatenate the linearized token sequences with a special `<SEP>` token. The output consists of a sequence $x_j{:}y_k$ ... $x_i{:}y_m$ ... where in every pair $u{:}v$, u is a variable node from the first AMR mapped to a node v from the second AMR. The SMATCH score is then calculated based on the predicted alignment.

To predict the node alignments/mapping of variables, we use a transformer based encoder-decoder NMT model. Details about the network structure and hyperparameters are stated in Appendix A.1.

3.3 Synthesis option II: SMATCH prediction

In this setup, we aim to predict SMATCH F1 scores for pairs of AMRs directly, in a single step. This means that we directly learn Eq. 2 with a neural network and our target is the ORACLE F1 score.

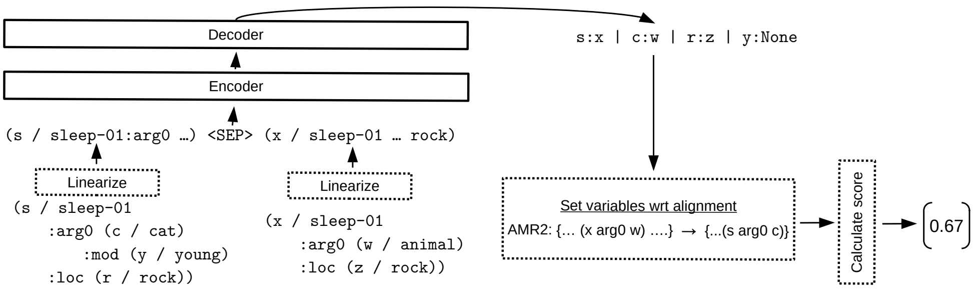


Figure 1: Seq2seq SMATCH alignment-learner.

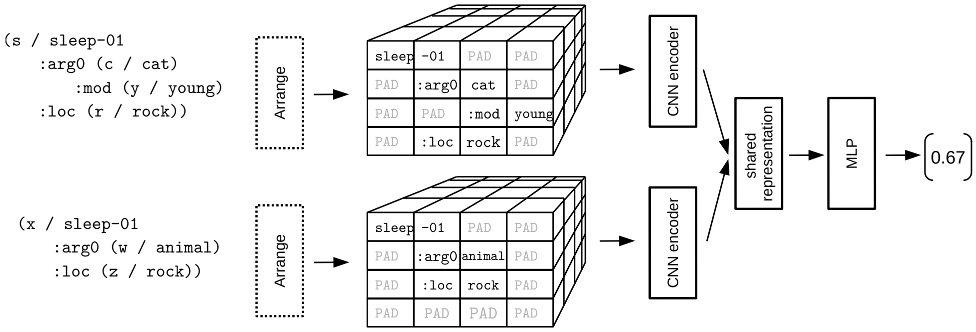


Figure 2: Implicit CNN-based SMATCH graph metric predictor.

To learn this mapping, we adapt the convolutional neural network (CNN) of Opitz (2020), as shown in Figure 2. The model was originally intended to assess AMR accuracy (Opitz and Frank, 2019), i.e., measuring AMR parse quality without a reference. Taking inspiration from human annotators, who exploit a spatial 'Penman' arrangement of AMR graphs for better understanding, it models directed-acyclic and rooted graphs as 2d structures, employing a CNN for processing, which is highly efficient. To feed a pair of AMRs, we remove the dependency graph encoder of the model and replace it with the AMR graph encoder. Moreover, we increase the depth of the network by adding one more MLP layer after convolutional encoding. A basic mean squared error is employed as loss function. More details about hyperparameters are stated in Appendix A.2.

3.4 Synthesis option III: AMR Vector learning

Inspired by Reimers and Gurevych (2019), we aim to make the CNN even more efficient, by alleviating the need for pair-wise model inferences. Instead of computing a shared representation of two CNN-encoded graphs, we process each representation with an MLP (w/ shared parameters), to obtain two vectors $NN(a)$ and $NN(b)$. These vectors are then tuned with signal from ORACLE(s):

$$\mathcal{L} = \sum_{(a,b,s)} \Big([1 - |NN(a) - NN(b)|] - s \Big)^2, \quad (3)$$

where || is returns a vector distance $\in [0, 1]$. This approach enables extremely fast search and clustering: the required (clustering-)model inferences are $O(n)$ instead of $O(n^2)$, since the similarity is achieved with simple linear vector algebra.

3.5 Data compression and extension tricks

Vocabulary reduction trick The SMATCH metric measures the structural overlap of two graphs. This means that we can greatly reduce our vocabulary, by assigning each graph pair a *local vocabulary* (see Figure 3, 'anonymize').

First, we gather all nodes from two graphs a and b, computing a joint vocabulary over the concept nodes. We then relabel the concepts with integers starting from 1. E.g., consider AMR a: *(r / run-01 :ARG0 (d / duck))*, and AMR b: *(x / run-01 :ARG0 (y / duck) :mod (z / fast))*. The gold alignment is $map^\star = \{(r,x),(d,y),(\emptyset,z)\}$. Now, we set the shared concepts and relations to the same index *run=run=1* and *duck=duck=2* and *:ARG0=:ARG0=3* and distribute the rest of the indices *r=4, d=5, x=6, y=7, z=8, fast=9, :mod=10*. This yields equivalent AMRs a' = *(4 / 1 :3 (5 / 2))*

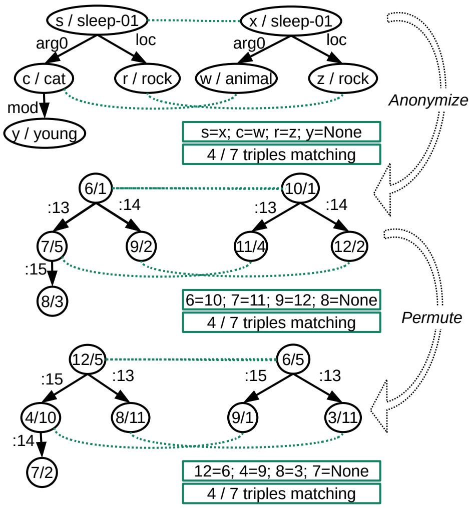


Figure 3: AMR graph anonymization and permutation.

	data trick	Eq. 2	Pea's ρ	time$^{(secs)}$
ORACLE	na	77.5	100	28680
rand. baseline	na	13.5	22.2	0.4
align. synthesis		39.0	52.8	1089
align. synthesis	voc	64.5	80.0	1089
align. synthesis	voc+aug	76.4	**98.4**	1089
score synthesis		na	<u>87.5</u>	140
score synthesis	voc	na	*82.0*	140
score synthesis	voc+aug	na	96.8	140
vector synthesis		na	84.7	0.7
vector synthesis	voc	na	75.6	0.7
vector synthesis	voc+aug	na	94.2	0.7

Table 1: Results of experiments. time: Approximate time for computing a pair-wise distance matrix on 1k AMRs on a TI 1080 GPU.

and b' = *(6 / 1 :3 (7 / 2) :10 (8 / 9))*. The target alignment then equals $map^{\star} = \{(4,6),(5,7),(\emptyset,8)\}$. This strategy greatly reduces the vocabulary size, in our case from 40k tokens to less than 700.

Auxiliary data creation trick We also find that we can cheaply create auxiliary gold data. We re-assign different indices to AMR tokens, and correspondingly modify the ORACLE alignment (Figure 3, 'permute'). In our experiments, we permute the existing token-index vocabularies 10 times, resulting in a ten-fold increase of the training data. We expect that, with this strategy, the model will better learn properties of permutation invariance, which in turn will help it synthesize the algorithm.

3.6 Evaluation

Output post-processing For the score synthesis (Option II) and vector synthesis (Option III), no further post-processing is required, since we directly obtain the estimated SMATCH scores as output. In the explicitly synthesized alignment algorithm, however, we get map, which is the predicted alignment from the sequence-to-sequence model. In this case, we simply feed map as an argument into Eq. 2, to obtain the scores.

Evaluation We compare the predicted scores $\hat{y}$ against the gold scores y with Pearson's ρ. However, for the model that predicts the explicit alignment (Option I), we can compute another interesting and meaningful metric. For this, we first calculate the average SMATCH score over AMR pairs given the gold alignment $map^{\star}$, and then we calculate the average SMATCH score over AMR pairs given the predicted alignment $\widehat{map}$ using Eq. 2. Note, that the SMATCH score based on the gold alignment constitutes an upper bound (max). Therefore, the SMATCH score based on the predicted alignment shows us how close we are to this upper bound. Our baseline consists of scores that are computed from a random alignment (*random*).

Results (Table 1) Our best model is the NMT approach using both data augmentation tricks. Obtaining 98.4 ρ, it very closely approximates the ORACLE, while being about 30 times faster than ORACLE and 76.2 points better then the random baseline. Perhaps the best tradeoff between speed and approximation performance is gained by the simple CNN score synthesis (96.8 ρ, 200x faster than ORACLE), also using both data tricks. The vector synthesis falls a bit shorter in performance (94.2 ρ), but it is extremely fast and achieves a 40,000x speed-up compared to ORACLE and about 1500x compared to the NMT approach.[1]

Consistently, the data extension (*aug*) is very useful. However, the vocabulary reduction (*voc*) is only useful for the NMT model (+27.2 points), whereas the scores are lowered for the CNN-based models (-5.5 for score synthesis, -9.1, *vector synthesis*). We conjecture that the CNNs learn SMATCH more indirectly by exploiting token similarities in the global vocabulary, and therefore struggle more to build a generalizable algorithm, in contrast to the bigger NMT transformer that learns to assess tokens fully from their given graph context.

[1] Note also that all models in Table 1 are significantly better (p<0.001) than the random baseline (one-sided test w/ z-transform).

4 Conclusion

We tested methods for learning to solve the hard structural graph matching problem that is key to many applications where we compare meaning representations. To this aim, we explored different neural architectures, and data augmentation strategies that help models to generalize. Our best models increase metric calculation speed by a large factor while incurring only small losses in accuracy that can be tolerated in many use cases. Our work paves the way to emergent use-cases of meaning representation that involve pair-wise analysis: e.g., semantic clustering or semantic pattern-based search for corpus linguistic studies.

Limitations

An issue of the tested methods concerns the alignment of larger graphs with many variables. On one hand, when the alignment candidate space increases, the runtime of SMATCH increases exponentially, while our considered approaches remain fast. However, in such a scenario, the neural models are bound to trade in some accuracy. Table 4 (Appendix A.3) assesses the effect size for differently sized alignment candidate spaces: while the model overall copes with different search space sizes, the accuracy loss is more considerable for large problems. We conclude that the fast and accurate alignment of *larger* AMR graphs remains a challenging and unsolved problem. However, note that such a bottleneck even exists for the algorithmic metrics, which either use a hill-climber that suffers from worsening sub-optimality or require a costly ILP procedure that may be infeasible for larger graphs (see Opitz (2023) for discussion and analysis). In this regard, we believe that our proposed data extension trick in combination with long-sequence transformers (Beltagy et al., 2020; Rae et al., 2020; Choromanski et al., 2021) may provide valuable means to address this limitation, or provide useful tradeoffs.

Other limitations are: i) the models that were trained without our proposed anonymization protocol were tested on graphs that contain English concepts, and therefore depend on an English vocabulary. ii) For loading the models, our tested methods require more RAM memory than SMATCH, which can be calculated on a low-budget computer.

References

Rafael Torres Anchiêta, Marco Antonio Sobrevilla Cabezudo, and Thiago Alexandre Salgueiro Pardo. 2019. Sema: an extended semantic evaluation for amr. In *(To appear) Proceedings of the 20th Computational Linguistics and Intelligent Text Processing*. Springer International Publishg.

Matej Balog, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, and Daniel Tarlow. 2016. Deepcoder: Learning to write programs. *CoRR*, abs/1611.01989.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. 2016. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Claire Bonial, Mitchell Abrams, David Traum, and Clare Voss. 2021. Builder, we have done it: Evaluating & extending dialogue-amr nlu pipeline for two collaborative domains. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 173–183, Groningen, The Netherlands (online). Association for Computational Linguistics.

Claire Bonial, Stephanie M. Lukin, David Doughty, Steven Hill, and Clare Voss. 2020. InfoForager: Leveraging semantic search with AMR for COVID-19 research. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 67–77, Barcelona Spain (online). Association for Computational Linguistics.

Marco Budinich. 1996. A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing. *Neural Computation*, 8(2):416–424.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Xinyun Chen, Dawn Song, and Yuandong Tian. 2021. Latent execution for neural program synthesis beyond domain-specific languages. In *Advances in Neural Information Processing Systems*, volume 34, pages 22196–22208. Curran Associates, Inc.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.

Luca M Gambardella and Marco Dorigo. 1995. Ant-q: A reinforcement learning approach to the traveling salesman problem. In *Machine learning proceedings 1995*, pages 252–260. Elsevier.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Thanh Lam Hoang, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam Nguyen, Dzung Phan, Vanessa Lopez, and Ramon Fernandez Astudillo. 2021. Ensembling graph predictions for amr parsing. In *Advances in Neural Information Processing Systems*, volume 34, pages 8495–8505. Curran Associates, Inc.

Hans Kamp. 1981. A theory of truth and semantic representation. *Formal semantics-the essential readings*, pages 189–222.

Guillaume Lample and François Charton. 2019. Deep learning for symbolic mathematics. *arXiv preprint arXiv:1912.01412*.

Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 397–407, Melbourne, Australia. Association for Computational Linguistics.

Emma Manning and Nathan Schneider. 2021. Referenceless parsing-based evaluation of AMR-to-English generation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 114–122, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Viswanath Nagarajan and Maxim Sviridenko. 2009. On the maximum quadratic assignment problem. *Mathematics of Operations Research*, 34(4):859–868.

Tahira Naseem, Abhishek Shah, Hui Wan, Radu Florian, Salim Roukos, and Miguel Ballesteros. 2019. Rewarding Smatch: Transition-based AMR parsing with reinforcement learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4586–4592, Florence, Italy. Association for Computational Linguistics.

Arvind Neelakantan, Quoc V. Le, and Ilya Sutskever. 2016. Neural programmer: Inducing latent programs with gradient descent. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018. Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Maxwell Nye, Armando Solar-Lezama, Josh Tenenbaum, and Brenden M Lake. 2020. Learning compositional rules via neural program synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 10832–10842. Curran Associates, Inc.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Juri Opitz. 2020. AMR quality rating with a lightweight CNN. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 235–247, Suzhou, China. Association for Computational Linguistics.

Juri Opitz. 2023. SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1595–1607, Dubrovnik, Croatia. Association for Computational Linguistics.

Juri Opitz, Angel Daza, and Anette Frank. 2021a. Weisfeiler-Leman in the Bamboo: Novel AMR Graph Metrics and a Benchmark for AMR Graph Similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.

Juri Opitz and Anette Frank. 2019. Automatic accuracy prediction for AMR parsing. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 212–223, Minneapolis, Minnesota. Association for Computational Linguistics.

Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, Online. Association for Computational Linguistics.

Juri Opitz, Philipp Heinisch, Philipp Wiesenbach, Philipp Cimiano, and Anette Frank. 2021b. Explainable unsupervised argument similarity rating with Abstract Meaning Representation and conclusion generation. In *Proceedings of the 8th Workshop on Argument Mining*, pages 24–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. Amr similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Linfeng Song and Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. Translate, then parse! a strong baseline for cross-lingual AMR parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.

Shira Wein, Wai Ching Leung, Yifu Mu, and Nathan Schneider. 2022. Effect of source language on AMR structure. In *Proceedings of The 16th Linguistic Annotation Workshop (LAW)*, Marseille, France. European Language Resources Association (ELRA).

Zizhen Zhang, Hong Liu, MengChu Zhou, and Jiahai Wang. 2021. Solving dynamic traveling salesman problems with deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*.

parameter	value
embedding size	512
encoder	4 transformer layers w/ 4 heads
decoder	4 transformer layers w/ 4 heads
feed forw. dim	2048
loss	cross-entropy
weight init	xavier
optimizer	adam
learning rate	0.0002
batch size	8192 (tokens)

Table 2: Overview of NMT hyper-parameters.

parameter	value
emb. dimension	100
‘pixels’	60x15
CNN encoder	concatenate(256 3x3 convs, 3x3 max pool 128 5x5 convs, 5x5 max pool)
MLP	relu layer followed by lin. regressor
weight init	xavier
optimizer	adam
learning rate	0.001
batch size	64

Table 3: Overview of CNN hyper-parameters.

A Appendix

A.1 Sequence-to-sequence network parameters

Hyper-parameters for the NMT approach are displayed in Table 2. The best model is determined on the development data by calculating BLEU against the reference alignments.

A.2 CNN network parameters

Hyper-parameters for the CNN approach are displayed in Table 2. The best model is determined on the development data by calculating Pearson’s ρ correlation of predicted scores and gold scores.

A.3 Analysis of performance on different problem sizes

See Table 4.

		Δ vs. ORACLE		
data type	data size	Eq. 2	Pea's ρ	better
full	1500	-1.1	-1.6	-
$<$ 5 vars	505	-0.6	-1.2	yes
$<$ 10 vars	1041	-0.7	-1.2	yes
$<$ 15 vars	1206	-0.9	-1.2	yes
$<$ 20 vars	1353	-0.9	-1.2	yes
$<$ 25 vars	1449	-0.9	-1.3	yes
$>$ 5 vars	940	-1.5	-2.2	no
$>$ 10 vars	476	-2.1	-3.6	no
$>$ 15 vars	318	-2.3	-5.4	no
$>$ 20 vars	183	-3.0	-10.1	no
$>$ 25 vars	83	-4.7	-19.3	no
$>$ 30 vars	37	-8.0	-25.5	no
$>$ 35 vars	20	-12.5	-41.1	no
single snt AMRs	1421	-1.0	-1.5	yes
multi snt AMRs	79	-2.7	-9.6	no

Table 4: Experiments on different test subsets that represent different problem complexities predicted with our best model (*align. synthesis+voc+aug*). $<>$ x vars means that one of two graphs contains $<>$ x variables. *better*: is the drop in accuracy of the model vs. ORACLE smaller compared with the model tested on all data?

AMR4NLI: Interpretable and robust NLI measures from semantic graphs

Juri Opitz[✿] **Shira Wein**[❀] **Julius Steen**[✿] **Anette Frank**[✿] **Nathan Schneider**[❀]

[✿]Heidelberg University [❀]Georgetown University
opitz.sci@gmail.com {steen,frank}@cl.uni-heidelberg.de
{sw1158,nathan.schneider}@georgetown.edu

Abstract

The task of natural language inference (NLI) asks whether a given premise (expressed in NL) entails a given NL hypothesis. NLI benchmarks contain human ratings of entailment, but the meaning relationships driving these ratings are not formalized. Can the underlying sentence pair relationships be made more explicit in an interpretable yet robust fashion? We compare semantic structures to represent premise and hypothesis, including *sets of contextualized embeddings* and *semantic graphs* (Abstract Meaning Representations), and measure whether the hypothesis is a semantic substructure of the premise, utilizing interpretable metrics. Our evaluation on three English benchmarks finds value in both contextualized embeddings and semantic graphs; moreover, they provide complementary signals, and can be leveraged together in a hybrid model.

1 Introduction

Natural language inference (NLI) and textual entailment (TE) assess whether a hypothesis ($\mathcal{H}$) is entailed by a premise ($\mathcal{P}$). Systems have various interesting applications, e.g., the validation of automatically generated text (Holtzman et al., 2018; Honovich et al., 2022). Recent systems make use of neural networks to encode $\mathcal{H}$ and $\mathcal{P}$ into a vector and thereupon make a prediction (Jiang and de Marneffe, 2019). While this can provide strong results when such systems are trained on large-scale training data, the overall decision process is not transparent and may rely more on spurious cues than on informed decisions (Poliak et al., 2018).

We aim to develop more transparent alternatives for NLI prediction, and therefore compare representations and metrics to predict entailment. Figure 1 gives an intuition of how 5 different sentences overlap in meaning. Representing each sentence with a semantic structure, we assume that, by and large, the semantic elements of an entailed sentence should be contained within the premise.

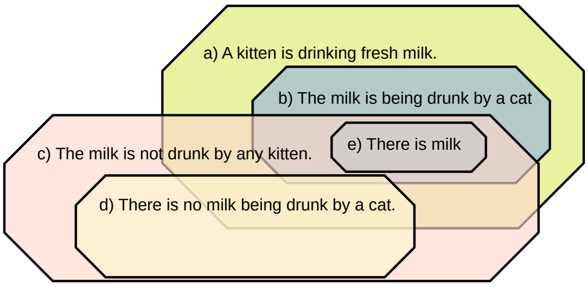

a) A kitten is drinking fresh milk.
b) The milk is being drunk by a cat
c) The milk is not drunk by any kitten.
e) There is milk
d) There is no milk being drunk by a cat.

Figure 1: Semantic (sub-)structure analysis shows that 4 of 25 candidate relations are true entailment relations: b) is entailed by a). d) is entailed by c). e) is entailed by a), b), and c).

These considerations trigger three interesting research questions that we will investigate in this paper: RQ1. *How to characterize a semantic structure?* RQ2. *How to determine/measure what is a substructure?* RQ3. *Is there a suitable and interpretable structure and measure that help to make NLI judgments more robust, or more accurate?*

To assess RQ1, we test three options: token sets, sets of contextualized embeddings, or graph-based meaning representations (MRs). As a meaning representation, we select Abstract Meaning Representation (AMR; Banarescu et al., 2013), using automatic AMR parses of the NLI sentences. To assess RQ2, we test different types of metrics that are designed or adapted to measure entailment on the selected structures, inspired from research on, e.g., MT evaluation and MR similarity. One of our key goals is to investigate whether it is possible to accurately capture relevant semantic substructure relationships via meaning representations. Finally, we show that we can positively answer all aspects of RQ3: First, besides their enhanced interpretability, unsupervised semantic graph metrics are more robust and generalize better than fine-

Proceedings of the 15th International Conference on Computational Semantics pages 275–283
June 21–23, 2023. ©2023 Association for Computational Linguistics

tuned BERT. Second, importantly, we show that they are high-precision NLI predictors, a property that we exploit to achieve strong NLI results with a simple decomposable hybrid model built from a fine-tuned BERT on the one hand, and a semantic graph score on the other. Code and data are available at `https://github.com/flipz357/AMR4NLI`.

2 Related work

Textual entailment Automatic approaches for this task date back to, at least, Dagan et al. (2006), who introduced a shared task for entailment classification. Since then, we can distinguish many different kinds of systems for addressing the task (Androutsopoulos and Malakasiotis, 2010), for instance, based on logics (Bos and Markert, 2005) or string- and tree-similarity (Zhang and Patrick, 2005), or graph matches of semantic frames and syntax (Burchardt and Frank, 2006) that aim in a similar direction as us. Recent releases of large-scale training corpora, such as SNLI (Bowman et al., 2015), or MNLI (Williams et al., 2018) can be exploited for supervised training of strong classifiers, e.g., by fine-tuning a BERT language model (Devlin et al., 2019). However, trained systems tend to suffer from the 'Clever Hans' effect and fall prey to spurious cues (Niven and Kao, 2019; Jin et al., 2020), such as position (Ko et al., 2020) or even gender (Sharma et al., 2021). This can lead to undesired and peculiar NLI system behavior. Poliak et al. (2018) show that supervised NLI systems can make many correct predictions solely based on $\mathcal{P}$, without even seeing $\mathcal{H}$. In our work, we want to test more transparent ways of rating entailment.

Metrics and meaning representations In part due to the reduced dependence on spurious cues, unsupervised/zero-shot metrics are found in evaluation of MT (e.g., BERTscore (Zhang et al., 2020), BLEURT (Sellam et al., 2020)), and NLG faithfulness checks (Honovich et al., 2022). Through the lens of abstract meaning representation (Banarescu et al., 2013), systems perform explainable sentence similarity (Opitz et al., 2021b; Opitz and Frank, 2022b), NLG evaluation (Opitz and Frank, 2021; Manning and Schneider, 2021), cross-lingual AMR analysis (Wein and Schneider, 2021, 2022; Wein et al., 2022), and search (Bonial et al., 2020; Müller and Kuwertz, 2022; Opitz et al., 2022). Leung et al. (2022) discuss different use-cases of embedding-based and MR-based metrics.

3 Method

3.1 Underlying research hypotheses

RH1: Semantic substructure analysis with asymmetric metrics can predict entailment We aim to study the entailment problem through analysis of semantic structure of $\mathcal{P}$ and $\mathcal{H}$. To perform such analysis, we need a metric that can measure the degree to which $\mathcal{H}$-structure is contained in the $\mathcal{P}$-structure. Therefore, we hypothesize that an *asymmetric metric* is preferable. Note that asymmetric metrics of complex objects like sets or graphs tend to be under-studied in NLP.[1]

RH2: Meaning representations are suitable semantic structures Semantic structures for $\mathcal{P}/\mathcal{H}$ should (ideally) hold facts that make them true. In this work we explore three options to build such structures for $\mathcal{H}/\mathcal{P}$: i) the set of text tokens, ii) the set of (contextual) embeddings obtained from them, and iii) graph-structured MRs. It is the latter that we hope will represent the facts best: A token set holds 'facts' in their surface form, which can be lossy in morphologically rich languages or with paraphrases. Contextual embedding sets, on the other hand, are powerful meaning representations, but hardly offer interpretability. An MR-structure is semantically more explicit, and is defined to represent a sentence's meaning through its parts.

3.2 Implementation

Preliminaries Let us define a

$$metric_T^{\mathcal{D}} : \mathcal{D} \times \mathcal{D} \rightarrow [0, 1] \qquad (1)$$

where 1 implies true entailment. With the parameter $\mathcal{D}$ we denote the metric domain (i.e., text with $metric^{text}$ or MR with $metric^{graph}$). The type parameter T specifies whether the metric is symmetric ($metric_{sym}$), or asymmetric ($metric_{asym}$).

3.3 Text metrics: $metric^{text}$

Token metrics Given a set of tokens from $\mathcal{H}$ and from $\mathcal{P}$, our asymmetric $metric^{text}_{asym}$ calculates a

[1] Indeed, most metrics used in NLP are *naturally symmetric* (e.g., cosine distance). Others fuse two asymmetric metrics into, e.g., an F1 score from precision and recall (Popović, 2015; Zhang et al., 2020). Alternatively, they are inherently asymmetric but enforce symmetry via balancing with an inversely correlated metric, e.g., BLEU (Papineni et al., 2002) focuses on precision but tries to factor in recall via a 'brevity penalty'. Even in related cases, where using an asymmetric metric seems intuitive, we find that sometimes symmetric metrics being used instead, e.g., Ribeiro et al. (2022) design a baseline for assessing faithfulness of automatically generated summaries with a symmetric F1 score using an AMR metric.

unigram *precision*-score:

$$\text{TokP} = |\mathcal{H}|^{-1} \cdot |toks(\mathcal{H}) \cap toks(\mathcal{P})|, \quad (2)$$

which is known to be a simple but strong predictor baseline for NLI-related tasks such as faithfulness evaluation in generation (Lavie et al., 2004; Banerjee and Lavie, 2005; Fadaee et al., 2018) (the most closely related 'BLEU-1' is used in many papers to assess system outputs). By switching $\mathcal{H}$ and $\mathcal{P}$ in Eq. 2, we calculate TokR, and based on these a symmetric $metric_{sym}^{text}$ TokS via harmonic mean.

BERTscore (Zhang et al., 2020) is a contextual embedding metric that calculates a greedy match between BERT embeddings of two texts, in our case: hypothesis $E^{\mathcal{H}} := embeds(\mathcal{H})$ and premise $E^{\mathcal{P}} := embeds(\mathcal{P})$. For our asymmetric $metric_{asym}^{text}$, we calculate a precision-based score:

$$\text{BertScoP} = |E^{\mathcal{H}}|^{-1} \sum_{e \in E^{\mathcal{H}}} \max_{e' \in E^{\mathcal{P}}} e^T e'. \quad (3)$$

Symmetric $metric_{sym}^{text}$ BertS is calculated as harmonic mean of BertScoP and BertScoR, the latter being obtained by switching $\mathcal{H}$ and $\mathcal{P}$ in Eq. 3.

3.4 MR Graph metrics: $metric^{graph}$

We study the following (a)symmetric MR metrics.

GTok Emulating TokP and TokS, we introduce GTokS and GTokP via Eq. 2 applied to two bags of graphs' node- and edge-labels.

Structural matching with Smatch (Cai and Knight, 2013) aligns triples of two graphs for best matching score, and returns precision (SmatchP) and a symmetric F1 score (SmatchS). We use the optimal ILP implementation of Opitz (2023).

Contextualized matching with WWLK aims at a joint and contextualized assessment of node semantics and node semantics informed by neighborhood structures. Therefore, Opitz et al. (2021a) first iteratively contextualize a vector representation for each node by averaging the embeddings of all nodes in their immediate neighborhood (the iteration count is indicated by K, which we set to 1). The normalized Euclidean distance of the concatenation of these refined vectors defines a cost matrix C, where C_{ij} is the distance of nodes $i \in \mathcal{P}$, $j \in \mathcal{H}$. The AMR similarity score is derived by solving a transportation problem: $WWLK = 1 - \min_F \sum_i \sum_j F_{ij} C_{ij}$ where F_{ij} is the flow between nodes i, j. Opitz et al. constrain $\sum_j F_{*j} = 1/|\mathcal{P}|$ and $\sum_i F_{i*} = 1/|\mathcal{H}|$. We call this symmetric setting **WWLKS**. We additionally propose an asymmetric sub-graph matching score **WWLKP** where we let $\sum_j F_{*j} \leq 1$ instead.

The most reduced version, which deletes all structural information from the graphs, is achieved by setting $k = 0$, which we denote as N(ode)Mover(P|S) score, analogously to the popular word mover's score (Kusner et al., 2015).

3.5 Hybrid model

Our decomposable hybrid model takes the prediction of a text metric, and the prediction of a graph metric, and returns an aggregate score. Such a metric can provide an interesting balance between a score grounded in a linguistic interpretation, and a score obtained from strong language models. If the two scores are both useful *and* complementary, we may even hope for a rise in overall results. To test such a scenario we will combine the best performing $metric_{graph}$ with the best performing $metric_{text}$ via a simple sum ($\alpha = 0.5$):

$$\alpha \cdot metric^{graph} + (1 - \alpha) metric^{text}. \quad (4)$$

4 Evaluation setup

Data sets We employ five standard sentence-level data sets: i) **SICK (test)** by Marelli et al. (2014) and **SNLI (dev & test)** by Bowman et al. (2015), as well as iii) **MNLI (matched & mismatched)** by Williams et al. (2018). Mismatched (henceforth referred to as MNLI-mi) can be understood as a supposedly more challenging data set since it contains entailment problems from a different domain than the training data, allowing a more robust generalization assessment of trained models. By contrast, in MNLI-ma(tched) the domain of the testing data matches that of the training data. For each data set, we map the three NLI labels to a binary TE classification setting, by merging *contradiction* and *neutral* to the *non-entailed* class.[2]

Evaluation metric We expect predictions to correlate with the probability of entailment, i.e.,

$$metric_T^{\mathcal{D}}(x, y) \uparrow \implies P(x \text{ entails } y) \uparrow,$$

[2] Same as in Uhrig et al. (2021), we use the T5-based off-the-shelf parser from amrlib for projecting AMR structures.

where ↑ means 'higher is better'. The NLI 'gold probability' labels are approximated as binary human majority labels. To circumvent a threshold search and obtain a meaningful evaluation score for comparing our metrics, we follow the advice of Honovich et al. (2022), who evaluate metrics for zero-shot faithfulness evaluation of automatic summarization systems, using mainly the Area Under Curve (AUC) metric. The AUC score is the probability that given randomly drawn instances ($\mathcal{P}, \mathcal{H}$, entailed) and ($\mathcal{P}', \mathcal{H}'$, non-entailed) the entailed instance receive a higher score. To rank metrics, we calculate two averages: AVG^{all} averages the scores over all data sets, while AVG^{nli} excludes SICK.[3]

Trained (upper-bound) We use a BERT trained on 500k SNLI examples.[4] It predicts an entailment probability from a vector representation generated by a transformer model.

5 Results

5.1 Main insights

Main insights can be inferred from Table 1. On all data sets, and overall on average, **asymmetric metrics substantially outperform symmetric metrics**. Sometimes they improve results by up to ten AUC points over their symmetric counterparts (e.g., NMoverS vs. NMoverP, +9.2). Comparing token sets, embedding sets and graphs, we find that both embedding set and graph prove advantageous: NMoverP achieves slightly better results than BertScoP, which has been *pre-trained* on large data. *Fine-tuned* BERT outperforms the tested unsupervised metrics when test data is in-domain (see SNLI results), but falls short at generalization. However, our **simple hybrid model can inform the output with sub-graph overlap and yields a strong boost outperforming all unsupervised and even trained metrics by a large margin (+4.5 points)**.

5.2 Analysis

Advantage of AMR and AMR metrics: high precision For each metric, we retrieve the p% most probable predictions, and calculate their accuracy. Results, averaged over all data sets, are displayed in Table 2. In high % levels, MR metrics outperform BertScoP by almost 20 points (e.g., BertScoP vs. WWLKP: +17.6 points), and even the fine-tuned BERT is strongly outperformed. Therefore, we can attribute the surprisingly strong performance of the graph metrics (and the hybrid model) to its potential for delivering high scores in which we can trust – if it determines that the semantic graph of $\mathcal{H}$ is (largely) a subgraph of $\mathcal{P}$, true entailment is most likely (in Appendix A, we show two examples).

Advantage of untrained (AMR) metrics: better robustness We check the robustness of our diverse NLI metrics on a controlled substructure of 3,261 SNLI testing examples by Gururangan et al. (2018), who removed examples that show spurious biases and/or annotation artifacts. Results in Table 3 show a catastrophic performance drop by trained BERT (−12.0 points), while untrained metrics such as TokP and WWLKP remain unaffected (+0.4 points) and WWLKP now even outperforms the SNLI-trained BERT model. Lastly, we see that the hybrid model can (partially) mitigate the drop introduced by its trained component (–7.3 points).

Discussion: graph metrics struggle with recall, and other limitations The MR metrics struggle with recall since they have problems to cope with MRs that strongly differ structurally, but not (much) semantically, which is a known issue (Opitz et al., 2021a). An example from our data is the following: In *The man rages*, *man* is the *arg0* of rage, while in the entailed sentence *A person is angry*, *person* is the *arg1* of *angry*, yielding large structural dissimilarity of MR graphs (SmatchP=0.0). In future work we aim to explore and improve this issue, such that we are able to identify that the experiencer of *angry* is strongly related to the *agent* of *rage*.

Potentially unrelated to the recall problem, other issues may hamper AMR usage for NLI, e.g., inconsistent copula modeling (Venant and Lareau, 2023), or parsing errors: even though parsers tend to provide high-quality output structures, they can still suffer from significant flaws (Opitz and Frank, 2022a), and thus their improvement may positively affect AMR4NLI performance.

Weights in hybrid model Recall that we can use α in Eq. 4 to weigh two metrics. We inspect different α in Figure 2 for fusing trainBERT (text) and WWLKP (graph, $\alpha \geq 0.5$: graph metric is weighted higher). While a balance ($\alpha \approx 0.5$) overall seems effective, SNLI profits if the text metric has more influence, and MNLI profits if the graph metric dominates. Finally, again we see more stable

[3] SICK contains entailment labels but not the direction of entailment and thus we do not include it in AVG^{nli}.

[4] https://huggingface.co/textattack/bert-base-uncased-snli

$\mathcal{D}$(omain)	*metric*	SICK	SNLI-dev	SNLI-test	MNLI-ma	MNLI-mi	AVG^{all}	AVG^{nli}
text	TokS	72.1	64.2	64.6	66.7	68.7	67.2	66.0
	TokP	74.7	70.0	70.6	68.2	70.3	70.8	69.8
	BertScoS	79.8	66.7	66.2	68.4	71.6	70.5	68.2
	BertScoP	**82.0**	74.5	74.0	**74.5**	**77.5**	76.5	75.1
AMR graph	GTokS	78.2	63.2	62.6	66.4	68.5	67.8	65.2
	GTokP	81.0	75.1	74.7	71.1	72.6	74.9	73.4
	NMoverS	77.7	65.8	64.9	66.7	68.5	68.7	66.5
	NMoverP	79.4	77.9	77.2	72.9	74.8	76.5	75.7
	SmatchS	76.3	63.3	62.3	65.7	67.6	67.0	64.7
	SmatchP	79.2	72.3	71.6	70.0	71.9	73.0	71.4
	WWLKS	77.2	66.4	65.6	65.7	67.5	68.5	66.3
	WWLKP	79.3	**78.0**	**77.3**	71.9	73.8	76.1	75.3
text	trainBERT	81.0	88.8	88.2	71.5	72.0	80.3	80.1
hybrid	trainBERT + WWLKP	**85.9**	**91.0**	**90.4**	**77.9**	**78.9**	**84.8**	**84.5**

Table 1: Overall AUC results on five data sets. The last two rows involve a trained component.

		AVG Accuracy scores									
$\mathcal{D}$(omain)	*metric*	1%	2%	3%	4%	5%	7%	10%	15%	AVG^{all}	AVG^{nli}
text	TokP	88.4	87.1	81.0	74.4	72.8	71.4	68.3	64.2	76.0	77.3
	BertScoP	74.5	74.0	73.3	73.9	73.9	73.0	72.0	69.4	73.0	73.8
AMR graph	GTokP	86.5	86.5	87.1	88.0	87.7	86.1	80.4	73.6	84.5	88.4
	NMoverP	85.3	84.5	85.0	85.2	86.2	84.7	82.4	74.2	83.4	89.6
	SmatchP	90.0	89.1	88.4	85.2	81.9	77.9	74.2	68.3	81.9	83.8
	WWLKP	**97.3**	**96.8**	**96.1**	**95.0**	**93.8**	88.4	82.4	74.8	90.6	90.7
text	trainBERT	84.5	84.0	82.9	81.5	80.6	79.0	76.8	73.2	80.3	81.9
hybrid	trainBERT + WWLKP	96.7	95.7	94.3	93.4	92.5	**90.2**	**86.7**	**82.2**	**91.5**	**92.9**

Table 2: Precision assessment. We select p% of a metric's highest predictions and check the ratio of true entailment.

training	no		yes	no	no/yes
domain	text	text embedding		AMR	hybrid
metric	TokP	BScoP	BERT	WWLKP	+BERT
AUC	71.0	71.4	76.2	77.7	83.1
AUC Δ	+0.4	-3.6	-12.0	+0.4	-7.3

Table 3: Evaluation on 3,261 *hard* SNLI-test examples. AUC Δ: observed change in performance (cf. Table 1).

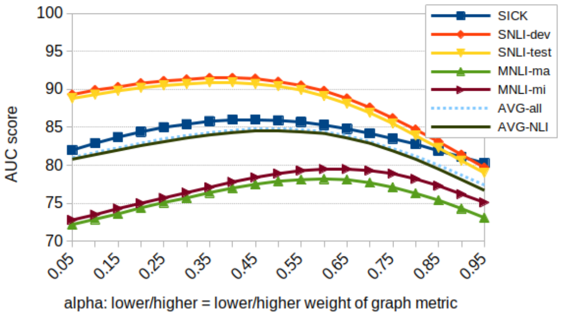


Figure 2: Balancing the hybrid text-graph metric.

performance of graph metrics overall (converging AUC with high α vs. diverging AUC with low α).

6 Conclusion

We find that metrics defined on advanced semantic representations are useful predictors of entailment. This is especially true for metrics performing asymmetric measurements on graph-structured meaning representations and sets of contextualized embeddings. Interestingly, meaning representation-based metrics offer advantages over strong embedding-based metrics beyond just interpretability: while showing similar performance as BERTscore, they are more robust than fine-tuned BERT *and* offer high-precision predictions. With this, we show that linguistic and neural representations can complement each other in a hybrid model, leading to substantial improvement over both untrained and trained neural approaches.

Acknowledgments

We thank anonymous reviewers for their feedback. This work is partially supported by a Clare Boothe Luce Scholarship.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

Claire Bonial, Stephanie M. Lukin, David Doughty, Steven Hill, and Clare Voss. 2020. InfoForager: Leveraging semantic search with AMR for COVID-19 research. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 67–77, Barcelona Spain (online). Association for Computational Linguistics.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Aljoscha Burchardt and Anette Frank. 2006. Approaching textual entailment with lfg and framenet frames. In *Proc. of the Second PASCAL RTE Challenge Workshop.[-]*.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2018. Examining the Tip of the Iceberg: A Data Set for Idiom Translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Nanjiang Jiang and Marie-Catherine de Marneffe. 2019. Evaluating bert for natural language inference: A case study on the commitmentbank. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6086–6091.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 1109–1121, Online. Association for Computational Linguistics.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Alon Lavie, Kenji Sagae, and Shyamsundar Jayaraman. 2004. The significance of recall in automatic metrics for mt evaluation. In *Machine Translation: From Real Users to Research: 6th Conference of the Association for Machine Translation in the Americas, AMTA 2004, Washington, DC, USA, September 28-October 2, 2004. Proceedings 6*, pages 134–143. Springer.

Wai Ching Leung, Shira Wein, and Nathan Schneider. 2022. Semantic similarity as a window into vector- and graph-based metrics. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 106–115, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Emma Manning and Nathan Schneider. 2021. Referenceless parsing-based evaluation of AMR-to-English generation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 114–122, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Almuth Müller and Achim Kuwertz. 2022. Evaluation of a semantic search approach based on amr for information retrieval in image exploitation. In *2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Juri Opitz. 2023. SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1595–1607, Dubrovnik, Croatia. Association for Computational Linguistics.

Juri Opitz, Angel Daza, and Anette Frank. 2021a. Weisfeiler-Leman in the Bamboo: Novel AMR Graph Metrics and a Benchmark for AMR Graph Similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.

Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, Online. Association for Computational Linguistics.

Juri Opitz and Anette Frank. 2022a. Better Smatch = better parser? AMR evaluation is not so simple anymore. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pages 32–43, Online. Association for Computational Linguistics.

Juri Opitz and Anette Frank. 2022b. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 625–638, Online only. Association for Computational Linguistics.

Juri Opitz, Philipp Heinisch, Philipp Wiesenbach, Philipp Cimiano, and Anette Frank. 2021b. Explainable unsupervised argument similarity rating with Abstract Meaning Representation and conclusion generation. In *Proceedings of the 8th Workshop on Argument Mining*, pages 24–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Juri Opitz, Philipp Meier, and Anette Frank. 2022. Smaragd: Synthesized smatch for accurate and rapid amr graph distance. *arXiv preprint arXiv:2203.13226*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Shanya Sharma, Manan Dey, and Koustuv Sinha. 2021. Evaluating gender bias in natural language inference. *arXiv preprint arXiv:2105.05541*.

Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. Translate, then parse! a strong baseline for cross-lingual AMR parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.

Antoine Venant and François Lareau. 2023. Predicates and entities in Abstract Meaning Representation. In *Proceedings of the Seventh International Conference on Dependency Linguistics (Depling, GURT/SyntaxFest 2023)*, pages 32–41, Washington, D.C. Association for Computational Linguistics.

Shira Wein, Wai Ching Leung, Yifu Mu, and Nathan Schneider. 2022. Effect of source language on AMR structure. In *Proceedings of The 16th Linguistic Annotation Workshop (LAW)*, Marseille, France. European Language Resources Association (ELRA).

Shira Wein and Nathan Schneider. 2021. Classifying divergences in cross-lingual AMR pairs. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 56–65, Punta Cana, Dominican Republic. Association for Computational Linguistics.

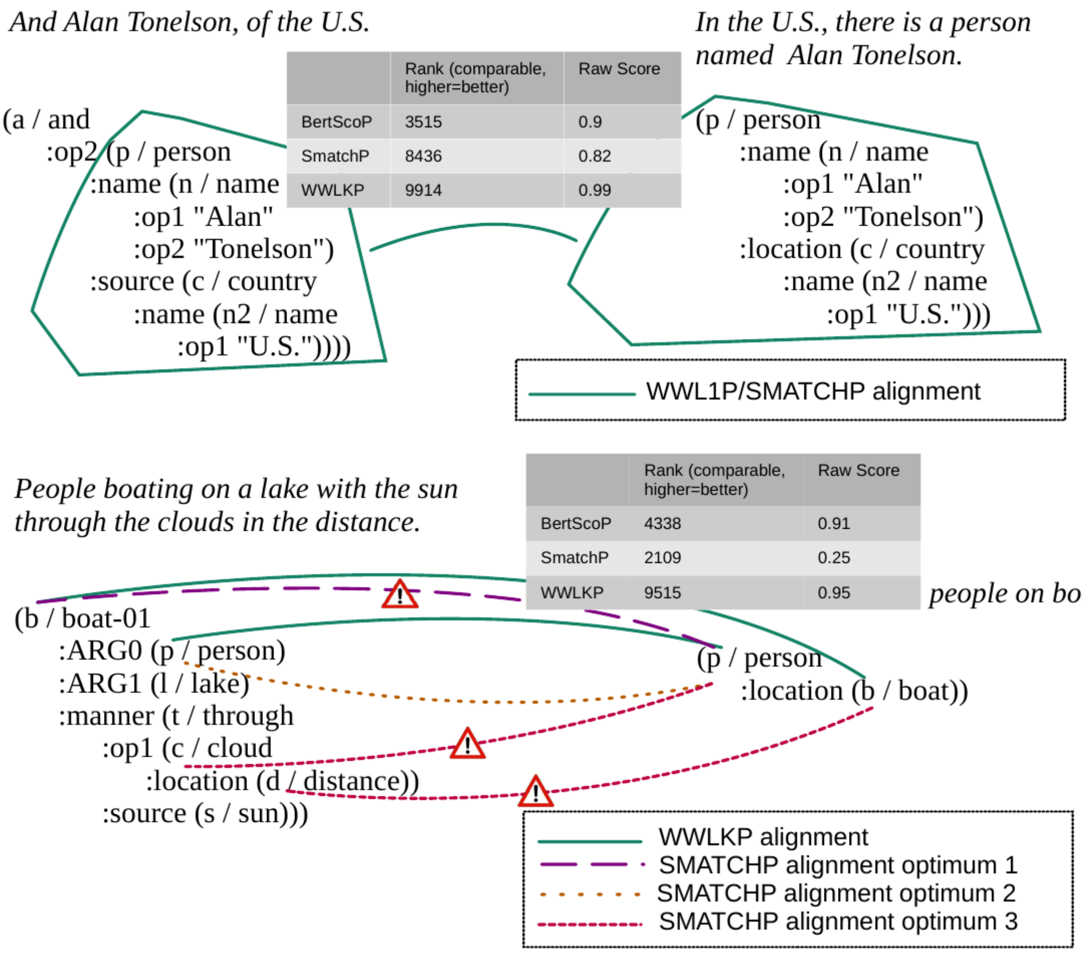
Shira Wein and Nathan Schneider. 2022. Accounting for language effect in the evaluation of cross-lingual AMR parsers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3824–3834, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Yitao Zhang and Jon Patrick. 2005. Paraphrase identification by text canonicalization. In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 160–166.

A Appendix



And Alan Tonelson, of the U.S.

In the U.S., there is a person named Alan Tonelson.

	Rank (comparable, higher=better)	Raw Score
BertScoP	3515	0.9
SmatchP	8436	0.82
WWLKP	9914	0.99

```
(a / and
    :op2 (p / person
        :name (n / name
            :op1 "Alan"
            :op2 "Tonelson")
        :source (c / country
            :name (n2 / name
                :op1 "U.S."))))
```

```
(p / person
    :name (n / name
        :op1 "Alan"
        :op2 "Tonelson")
    :location (c / country
        :name (n2 / name
            :op1 "U.S.")))
```

People boating on a lake with the sun through the clouds in the distance.

	Rank (comparable, higher=better)	Raw Score
BertScoP	4338	0.91
SmatchP	2109	0.25
WWLKP	9515	0.95

people on boat

```
(b / boat-01
    :ARG0 (p / person)
    :ARG1 (l / lake)
    :manner (t / through
        :op1 (c / cloud
            :location (d / distance))
        :source (s / sun)))
```

```
(p / person
    :location (b / boat))
```

WWLKP alignment
SMATCHP alignment optimum 1
SMATCHP alignment optimum 2
SMATCHP alignment optimum 3

Figure 3: Two example ratings assessing true entailment: The first shows how MR can define a useful semantic set, the second shows that sometimes embedding-based graph metrics, such as WWLKP, are needed to assess the subgraph properly (in this example, SmatchP provides semantically meaningless alignments and a score that is too low.)

Use Defines Possibilities: Reasoning about Object Function to Interpret and Execute Robot Instructions

Mollie Shichman[1], **Claire Bonial**[2], **Austin Blodgett**[2], **Taylor Hudson**[3], **Francis Ferraro**[4], **Rachel Rudinger**[1]

[1] University of Maryland, College Park, [2] Army Research Lab
[3] Oak Ridge Associated Universities, [4] University of Maryland, Baltimore County
mshich@umd.edu, claire.n.bonial.civ@army.mil, ferraro@umbc.edu, rudinger@umd.edu

Abstract

Language models have shown great promise in common-sense related tasks. However, it remains unseen how they would perform in the context of physically situated human-robot interactions, particularly in disaster-relief scenarios. In this paper, we develop a language model evaluation dataset with more than 800 cloze sentences, written to probe for the function of over 200 objects. The sentences are divided into two tasks: an "easy" task where the language model has to choose between vocabulary with different functions (Task 1), and a "challenge" where it has to choose between vocabulary with the same function, yet only one vocabulary item is appropriate given real world constraints on functionality (Task 2). DistilBERT performs with about 80% accuracy for both tasks. To investigate how annotator variability affected those results, we developed a follow-on experiment where we compared our original results with wrong answers chosen based on embedding vector distances. Those results showed increased precision across documents but a 15% decrease in accuracy. We conclude that language models do have a strong knowledge basis for object reasoning, but will require creative fine-tuning strategies in order to be successfully deployed.

1 Introduction

When it comes to using robots in disaster-relief scenarios such as search-and-rescue, it is essential that a robot can interpret and execute an instruction based on its current understanding of the objects detected in its environment. For example, in order to *Enter the building,* the robot should know to search for entrance points, such as doors and windows. Similarly, to *Scan the second floor,* the robot must be able to find appropriate ways to get to the second floor, such as stairs. Finally, to *Use the outlet to check for power,* the robot must know how outlets are used. Essentially, the robots need to know an object's function(s) in order to complete envisioned interactions.

Envisioned interactions are a multi-modal approach to responding to natural language instructions. For this paper, we assume that various sensors and computational systems, such as LIDAR, motion, or camera sensors, have taken care of identifying the objects in a scene. This information is passed to a language based world model, which deduces which, if any, of the objects perceived are relevant to the instruction based on the objects capabilities. This information would then be passed on to a lower-level policy-planning tool. An envisioned interaction that this research supports is depicted in Figure 1.

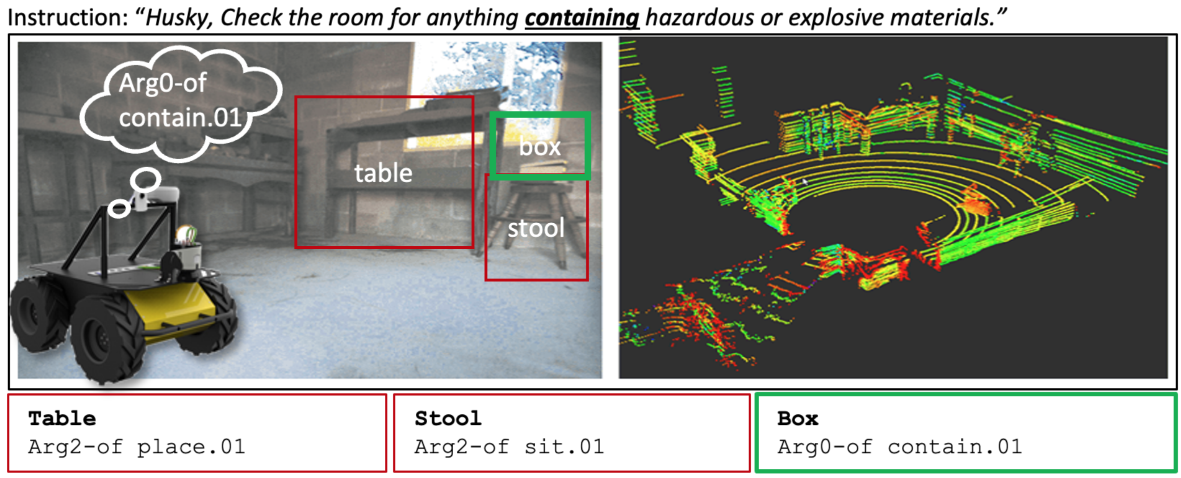
To understand the possibilities for executing a natural language instruction within the current environment, the robot requires *apriori*, commonsense knowledge of the objects in the environment. In particular, knowledge of object function is critical for interpreting natural language instructions in physically situated disaster-relief tasks. Given that such tasks are dynamic and dangerous, a robot should be able to accept unconstrained natural language (as opposed to placing a cognitive burden on the rescue worker to use a robot's controlled language). We hypothesize that a large language model (LM) would be uniquely equipped to handle this challenging task of supporting commonsense reasoning about an object's function for situated natural language understanding (NLU), due to the LM's latent world knowledge (Petroni et al., 2019).

The contributions of this paper include:

1. The development of a dataset of objects, found to be relevant to disaster-relief scenarios, with their functions established in terms of PropBank rolesets (Section 2);

2. The creation of an LM evaluation set of sentences that probe the model for its knowledge

Proceedings of the 15th International Conference on Computational Semantics pages 284–292
June 21–23, 2023. ©2023 Association for Computational Linguistics



Instruction: *"Husky, Check the room for anything **containing** hazardous or explosive materials."*

Figure 1: Envisioned interaction in which understanding and executing the instruction are supported by reasoning about objects in the environment detected via visual sensors (left) and LIDAR sensors (right). Given an instruction to look for a container of materials, the functions of detected objects with labels "table," "box," and "stool," can be compared against the containment function, represented by the PropBank role and roleset, "Arg0-of contain.01." Here, only "box" has the appropriate function, prompting further exploration of contents of the box.

of those object functions in both an "easy" task (Task 1) and a "challenge" (Task 2) (Section 3.1), and the augmentation of Task 1 for a follow-on evaluation (Section 3.2);

3. DistilBERT (Section 4) evaluation results (Section 5) with suggestions for future improvements informed by related work (Sections 6, 7).

We will make our object function dataset and cloze-sentence evaluation set available upon request.

2 Object Function Background and Dataset

PropBank (Palmer et al., 2005) is a semantic role labeling framework that provides a lexicon of event "rolesets," where each corresponds to a particular sense of a verb, eventive noun, or relational adjective. Each sense is described in terms of its set of participant roles, captured as argument numbers "ARG" 0-5, or as "ARG-M" modifier or adjunct arguments. In addition to the lexicon, PropBank provides a large corpus of annotated data where each relation is marked up with its sense in the lexicon and the arguments are marked for their semantic role with respect to that sense roleset. This lexicon is also used in the annotated corpus of Abstract Meaning Representation (AMR) (Banarescu et al., 2013). The standard roles are ARG0, which corresponds to Dowty's prototypical Agent, and ARG1, which corresponds to the prototypical Patient (Dowty, 1991). The corresponding semantic roles of the other, higher-numbered ARGs 2-5 are verb specific. ARG-Ms, which can theoretically modify or accompany any verb, include roles such as INSTRUMENT and PATH.

By leveraging the PropBank lexicon and corpus to establish that ladders and stairs fulfill the same role semantically (as the ARG1 for *climb.01*), we are able to derive a set of objects that have the same functionality (ways to climb between floors of a building).[1]Essentially, using Propbank is a pre-existing method of establishing commonalities between objects' functions. For example, Propbank allows us to group barrels, boxes, crates, and cabinets together because they all are ARG0 of the Propbank sense *contain.01*

While alternative resources that encode object functionality do exist, such as the Suggested Upper Merged Ontology (SUMO) (Niles and Pease, 2003), which includes axioms and object definitions indicating function, we found that PropBank provided a data-driven approach for us to develop a ground truth of each object's functionality as well as an elegant way of encoding and representing that function, for example as ARG1 of climb-01. This semantic representation of function thus fits with broader NLU that leverages the PropBank and Abstract Meaning Representation (AMR) (Banarescu et al., 2013) for a distillation of unconstrained natural language instructions into action primitives and their parameters, executable by a

[1] See climb.01 roleset: `https://propbank.github.io/v3.4.0/frames/climb.html#climb.01`

robot. Object function is therefore encoded in the same way as the natural language instructions that might reference the object or desired functionality. For example in Figure 1, the instruction would be parsed into AMR, abstracting the target object which would be a thing that is an ARG0-of contain.01. Then, the objects currently detected, localized, and labeled in the environment using the robot's sensors would be evaluated for which object had the matching function of ARG0-of contain.01. We created a vocabulary of about 280 objects mentioned in a human-robot dialogue corpus (Marge et al., 2017). These dialogues were previously collected via wizard-of-oz experimental interactions between people and remotely located robots in a search and exploration task, which is similar to our target domain of robotic exploration for disaster relief. Informed by existing PropBank and AMR corpora, an annotator then decided the best role for each vocabulary item given an appropriate PropBank sense. For instance, the word *crate* was assigned the PropBank ARG0 role of the *contain.01* sense—indicating it is the container holding some kind of contents. After one annotator made initial judgements, two other annotators familiar with PropBank and AMR reviewed the annotation to validate or offer alternative labels for vocabulary whose PropBank annotations were more difficult to surmise.

3 LM Evaluation Dataset

With the objects labelled with the appropriate Propbank sense-role pairing to signify their functionalities, we needed to develop a method of zero-shot testing a language model. For this methodology, it was important that we develop an understanding both of the language model's capabilities and how a small group of expert human annotators could be skewing the results beyond their particular writing styles. This led to two rounds of evaluation data generation: one with a manually developed answer set, and one with an answer set based on distances within LM vector space.

3.1 Manually Developed Sentence and Answer Set

We wanted to analyse both the LM's ability to differentiate between objects with different functions (Task 1) and between objects with the same function (Task 2), so we designed two different tasks. For both, we generated cloze sentences that express the need for a particular functionality or affordance, where the correct answer is one of our object vocabulary items that offers that functionality (according to the function annotations described in Section 2). The LM's task was to pick the correct word from a short list of possible answers. Providing a short list of answers was both inspired by the Winograd Schema Challenge (Levesque et al., 2012) and because a robot would be faced with a set of recognized and labeled objects in its environment to choose from in a given disaster-relief scenario.

For Task 1, annotators wrote sentences such that all words with the same function can reasonably fill in the blank. For instance, in the sentence *Go check if there's anything suspicious inside that BLANK*, the blank can be filled by any word denoting an object whose function is a container, be it a barrel or a cabinet. In Task 1, two wrong options were also presented; these did not share the function of the right answer and were arbitrarily chosen by the annotator from the rest of the vocabulary list. One sentence was written for each function. If more than one vocabulary term had the same function, the same sentence would be used multiple times, but the correct answer would be changed so that each word with the same function was represented in the evaluation set. This was done to see if a LM was consistent in correctly choosing objects with the same function. The sentences were written fairly explicitly so that only the word's intended function could be reasonably inferred by a human reader, as we had words that could serve multiple functions, like *stairs*, which could fall under *ascend.01* PATH or *descend.01* PATH.[2] A sample of Task 1 sentences from one author/annotator is given below. The LM must choose which of the answer choices is the most likely filler of the masked position.[3]

(1) I need to see from higher up, so I'm going up the [MASK].

Choices ladder, cushion, tomato
Correct ladder

(2) I need to see from higher up, so I'm going up

[2] While a qualitative analysis of the results did not show any evidence of polysemous words causing errors, we are uncertain how many vocabulary items are polysemous and what that effect may have on our results.

[3] From an implementation perspective, we used the following format: SENTENCE with [MASK] ||| ANSWER CHOICES ||| CORRECT ANSWER.

the [MASK].

Choices stairway, cushion, tomato
Correct stairway

(3) The [MASK] will keep the horse from running out of the pen.

Choices mop, barrier, bucket
Correct barrier

(4) The roof collapsed when the flimsy [MASK] failed to support its weight.

Choices curtain, lamp, column
Correct column

Note that the answer vocabulary is based upon objects mentioned in the human-robot collaborative exploration corpus, and therefore relevant to robotic exploration tasks, even if the sentences are not instructions per se. By not limiting the annotators to writing instructions only, we allowed for more use-cases given the object's function. For example, here are three sentences given the function of *contain.01 ARG0*.

1. I was getting ready to move, so I put all of my belongings into a [MASK].
2. Go check if there's anything suspicious inside that [MASK].
3. I need to hold my collection of cups for safekeeping, so I'm going to use a [MASK].

Each sentence works for any objects that can contain, but they each highlight a unique aspect of containing that would be important for a robot to recognize.

For Task 2, we narrowed our focus in order to study how LMs can leverage commonsense knowledge to differentiate between items with the same function. For our initial evaluation, we chose two functions from our dataset that contained the most unique objects within them: facilitating transport (objects listed with this function include *car, boat, bike*) and containment (objects listed include *jug, luggage, cup*). Within each function, we wrote sentences that would be true for one object with the same functionality but not another. As an example, the LM could choose between *ladder* and *stairs* to fill in the blank for *I need to get to the second floor, so I'm going to move the BLANK to that window*. Both serve the function of climbing, but they are not interchangeable because ladders are portable and stairs are not. We generated all possible pairings of objects within our chosen functions and randomly selected the pairings for sentence generation. More details about the sentence data can be found in Table 1 and a sample from one annotator for the transportation function is given here:

(4) I'm trying to get my legs in shape, so I take my [MASK] to school each day.

Choices bicycle, boat
Correct bicycle

(5) My husband's going green so he takes his [MASK] everywhere he needs to go.

Choices bicycle, car
Correct bicylce

(6) Today you really need air conditioning, so you decide to take the [MASK] to get to the office.

Choices bicycle, car
Correct car

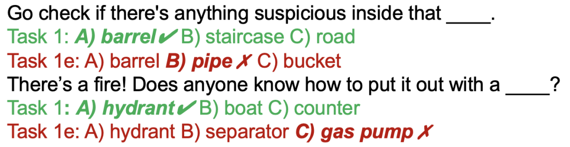
(7) She couldn't afford any gas, so she had to ride her [MASK] to the next village over.

Choices bicycle, motorcycle
Correct bicycle

Note that the real-world knowledge required to determine the correct answer for Task 2 we hypothesized to be fairly nuanced—a connection between biking and *getting legs in shape*, or *going green*, or *NOT being able to afford gas*, for example.

3.2 Answer Sets from Embedded Vectors

After an initial analysis of the results of Task 1, we noted that the performance across "documents," where each document is the set of evaluation sentences written by a single annotator, varied substantially (as we will describe in greater detail in Section 5). This prompted us to consider where this variation was coming from. Each document was intended to evaluate the LM's knowledge of the functionality of the same set of objects, so this was variance outside of what could be concluded to be related to commonsense knowledge of object functionality. We only had three annotators, which has been shown to introduce bias (Geva et al., 2019). As each annotator both authored sentences and selected the sentence's wrong answers, we hypothesized that both factors likely add bias to our results. As a first step to reduce inadvertent variance stemming from wrong answer choices, we elected to experiment with different methods of choosing wrong answers for Task 1 to see how the wrong answers affected the results of our experiments. Specifically, we decided to compare the results with the manually chosen wrong answers for Task 1 with a more technical procedure in which we selected the wrong answers based on the cosine distance between vectors taken from the LM's result of encoding each individual vocabulary term. While the embedded vectors of individual words differ from the embedded vector the word takes within an encoded BERT sentence, we decided this was a reasonable approximation that also took into account the limited onboard computing power a robot would have for our task.



Go check if there's anything suspicious inside that ____.
Task 1: ***A) barrel*** ✓ B) staircase C) road
Task 1e: A) barrel ***B) pipe*** ✗ C) bucket
There's a fire! Does anyone know how to put it out with a ____?
Task 1: ***A) hydrant*** ✓ B) boat C) counter
Task 1e: A) hydrant B) separator ***C) gas pump*** ✗

Figure 2: Two examples of the effect different answer choices for task 1 vs task 1e. The answers chosen for closeness by vector distance are often have similar functions (carry vs. contain) or potentially related within a conceptual domain (gas pump vs. hydrant), making Task 1e more challenging.

We ran several experiments at closer and further cosine distances to test the hypothesis that the LM would choose more wrong answers if they had a closer cosine distance to the correct answer. We named this "Task 1e," or Task 1-encoded vectors. For each right answer, we compiled a subset of valid distractors from our original vocabulary list, then chose the wrong answers by their ranking in our query. Examples comparing sentences and answers for Task 1 and 1e are shown in Figure 2. This approach does not account for any changes in density within the vector space. However, for all experiments the standard deviation of distances remained fairly uniform. This led us to believe that the ranked distances were all similar enough that the comparison between functions is still fair.

4 Experimental Setup

We used Huggingface's pipeline class with the fill-mask task and the DistilBERT uncased model. We chose DistilBERT because it is lightweight while having very similar accuracy to the full base BERT model (Sanh et al., 2019). This allows the model, theoretically, to be loaded directly onto the robot platform, keeping its space to a minimum without sacrificing too much accuracy. To calculate the vector embedding's cosine distances, we followed in BERT-as-a-service's footsteps: we took the second to last layer of DistilBERT to represent each vocabulary term (McCormick and Ryan, 2019). We used Sci-kit Learn's implementation of a KD-Tree to store the resulting vectors (Pedregosa et al., 2011). All experiments were run with Pytorch and all scores were put into log space (Paszke et al., 2019).

	Task 1	**Task 2**
Sentences	608	236
Objects	183	21
Functions	65	2
LM Accuracy	**81.5%**	**79.7%**
Acc. Range	**22.8%**	**15.0%**

Table 1: Size and shape of the data, as well as DistilBERT's average accuracy for Task 1 and Task 2 and the range in its accuracy across documents.

Multi-token Vocabulary Terms One challenge we faced was how to fairly compare the scores of single-token vocabulary terms as opposed to multi-token vocabulary terms, since the WordPiece tokenizer used by DistilBERT can potentially break words into subwords. To solve this problem, we adapt the sentence level scoring scheme of pseudo-log likelihood from Salazar et al. (2020) when vocabulary items have multiple tokens. Specifically, for tokens $t_1 \dots t_n$ that make up word W with T_j tokens before the mask and T_k tokens afterwards, where j and k are both natural numbers, we calculate the probability as shown:

$$log(p(t_1|T_j, T_k)) + log(p(t_2|T_j, t_1, T_k)) \dots + \\ log(p(t_n|T_j, t_1, t_2, \dots t_{n-1}, T_k))$$

We found that normalizing the scores by the number of tokens improved accuracy results. We hypothesize that this normalization reduced the LM's bias towards single token answers, but more experimentation is required to fully understand the effects of normalizing scores by token length.

5 Results and Discussion

5.1 Task 1 and Task 2

The accuracies for Task 1 and Task 2 were nearly identical, as shown in Table 1. This was somewhat

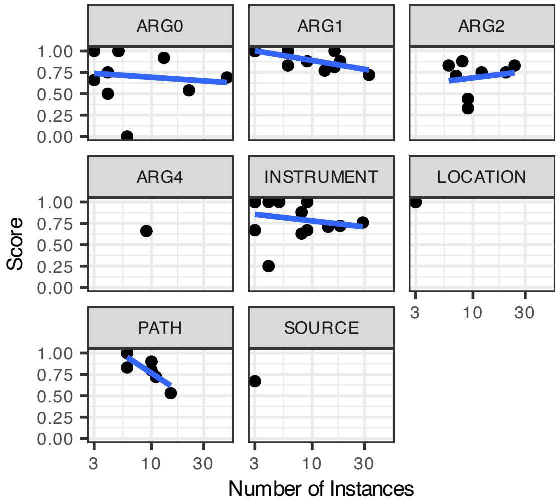


Figure 3: A breakdown of accuracy across PropBank roles for Task 1 by number of instances. Notably, accuracy decreases as the number of samples increases regardless of the role the vocabulary term plays in the sentence.

surprising, as we thought that DistilBERT would be more accurate when differentiating between words with different functions than within the same function, where we hypothesized more nuanced commonsense knowledge was required to recognize the correct answer. This could be from word co-occurrence probabilities. DistilBERT knows that *legs* is more likely to co-occur with *bicycle* than *boat*, so it doesn't necessarily need to do any reasoning. It's also possible that reporting bias played a role: annotators may have spent more time carefully differentiating between objects with similar functions than they do differentiating objects with significantly different functions because it is more self-explanatory to the reader what the latter differences are. Thus, the sentences for Task 2 may have inadvertently been more informative.

We also obtained DistilBERT's accuracy across each function. Some trends are immediately visible. First, regardless of the role the vocabulary term played in the sentence, the more sentences written for a specific function, the worse accuracy got. This can be seen in 3. Since so much expert knowledge was used when assigning the PropBank sense and role while deciding function, we do not believe this is because of labelling error. Rather, functions with fewer sentences tended to be more common, specific, and explicit than functions with many sentences. For instance, some functions that had only one or two sentences that scored well were dig.01 ARG2 (which corresponded with shovel), rotate.01 ARG1 (which corresponded with wheel), and buttress.01 ARG0 (which corresponded with column). All of these items are strongly correlated with the functions. Larger categories that struggled more included contain.01 ARG0, whose vocabulary items ranged from cabinet to can, and occupy.01, whose terms ranged from car to barn. Since the annotators were writing sentences that worked with all vocabulary of the same function, the sentences with "larger" functions had to be more general and likely had fewer semantic clues for DistilBERT to utilize. This suggests that LMs have room to improve on more general cases for objects for our use case, including handling a wider variation in object function use.

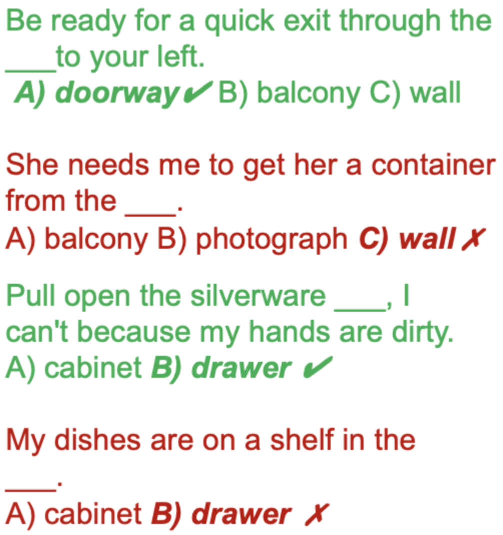
Ranked Distance	Accuracy	Accuracy Range
1st, 2nd	62.2%	15.4%
1st, 3rd	60.1%	15.2%
2nd, 3rd	66.5%	13.3%
2nd, 5th	67.2%	10.0%
6th, 11th	76.0%	9.1%
12th, 21st	77.1%	18.7%
26th, 36th	81.5%	7.3%

Table 2: Results for Task 1e. Ranked distances refer to the cosine distance from the wrong answers to the correct answer and are ranked by closeness to the correct answer, from 1st closest to 36th closest.

Even though the results for Task 1 were strong, within the task there was a wide range in accuracy over each document, with 2 documents in the same task differing in accuracy by as much as 22%. We attributed this wide range to annotator bias (as mentioned in Section 3.2). While annotator bias is a given in a dataset with few sentence creators, we wanted to minimize as much bias as possible to ensure the LM was a sufficient basis for our ultimate use case of collaborative, disaster-relief communication. One clear place to eliminate bias was in the selection of wrong answers, motivating the development of Task 1e.

5.2 Task 1e, Embedding Distances

For Task 1e, we achieved our initial goal of reducing the range in accuracy over all documents for all experiments, as shown in Table 2. This demonstrates that the wrong answers chosen by sentence authors did have an impact on accuracy, as we had hypothesized. The overall accuracy ranges also show that the impact of manually selected wrong answers is overall positive. In other words, the



Be ready for a quick exit through the ___to your left.
A) doorway ✔ B) balcony C) wall

She needs me to get her a container from the ___.
A) balcony B) photograph ***C) wall*** ✗

Pull open the silverware ___, I can't because my hands are dirty.
A) cabinet ***B) drawer*** ✔

My dishes are on a shelf in the ___.
A) cabinet ***B) drawer*** ✗

Figure 4: Example sentences that DistilBERT correctly (shown in green with check marks) and incorrectly (shown with a red *X*) answered from Tasks 1 & 2.

manually selected wrong answers in Task 1 were generally easier for the LM to eliminate than the wrong answers selected for all but the most distant wrong answer choices in Task 1e. The accuracy range also decreases as vocabulary terms get further away from the correct answer in vector space, demonstrating that the sentence alone does not give DistilBERT enough information to differentiate between the answers, and that it needs the answers choices to provide extra information for it to make a correct decision. We also examined the scores for each function as we did with Task 1, and we found that scores decreased rather evenly across the board, regardless of how many sentences were testing the function.

As we had hypothesized, the overall scores and the scores by function generally improved linearly as the wrong answers moved further away from the correct answer. However, when looking at individual documents and functions with wrong answers close to the correct answer, that linearity breaks down, and performance seems very dependent on the language choices of individual annotators. When examining the data qualitatively, it's often not clear from a linguistic perspective why DistilBERT assigned the probability it did. For instance, DistilBERT thought it was more likely that one would use a motorcycle to *catch their balance* than a rail, or even a television. It's also not immediately clear how the annotators writing styles are "easier" or "harder" for DistilBERT to work with. Other unclear examples can be seen in Figure 4 for both Task 1 and 2. We suspect that larger language models which utilyze larger vocabularies than DistilBERT would be more linguistically informed due to the increased data and training time, but we leave that to future work.

While the scores decreased significantly when going from annotator-selected wrong answers to ranked distance wrong answers, DistilBERT still scores far better than random and shows it does have a strong amount of knowledge on object functions. Overall, our expectations for DistilBERT's zero-shot knowledge were exceeded in both tasks. Nonetheless, given the high stakes of our application domain, we plan paths for improvements in future work (Section 7).

6 Related Work

We were inspired in our own research by Chen et al. (2022), who also test an LM's zero-shot knowledge with respect to physically situated settings. The authors' goal is to use LMs to help robots determine the type of room it is in for a given 3D scene. To test if LMs could be effective at this task, they automatically generate sentences from the template "The *r* often contains *o*", where *r* is a type of room and *o* is an object often found in that room. The authors ran their sentences through the masked LM BERT with the room masked to see how well BERT could predict the room based on the objects. The authors found that rooms with very specific items (bathrooms, bedrooms, kitchens) were easier to identify than rooms which had furniture that can be in many rooms (dining rooms, living rooms). This showed us the effects reporting bias can have on physical commonsense LMs and prompted us to research this for our own use case.

The ultimate goal of our research is to use LMs for robot policy planning with a strong understanding of the LM's decision-making process and embeddings, since high stakes situations demand accountability. Dipta et al. (2022) approach this task by creating linguistically informed embeddings within a custom encoder-compressor-decoder network. The network was trained to recognize the hierarchical nature of events by using frames from FrameNet (Baker et al., 1998) only partially describing said event. By injecting linguistically informed knowledge, while not requiring specific vocabulary to indicate that an event is occurring, Dipta et al. (2022) had strong performance with a reasonable explanation of what each part of the neural network is doing.

In terms of planning with LMs, there are multiple interesting approaches. Driess et al. (2023) trained an LM, called PaLM-E to also accept image and continuous sensor data, as well as text. By encoding the non-text data into vectors that are the same size as a text vector, the model can complete a variety of tasks straight out of the box while also allowing for downstream fine-tuning. Notably, it can output plain text that can be interpreted as a robotic policy, though PaLM-E has to interpret on its own what a particular robot's capabilities are. More testing needs to be done to see if the robot can behave consistently, and the authors caution that it is not meant for long-term tasks. Another model made by Song et al. (2022) utilizes an upper level LM, in their case GPT, with some few-shot training for high-level policy planning. They separately designed a lower level model that handles the execution of movement and other low-level tasks. Importantly, if the lower level model can't execute a task, it can query the higher level model with the information it perceived about the environment for an updated policy. This enables it to handle long term, complex tasks. However, both of these models lack the explainable nature of Dipta et al. (2022) with its basis in linguistic theory.

7 Future Work and Conclusion

Given the overall success of these experiments, we have several avenues of future work. First, we want to test how different LMs perform on our dataset. While DistilBERT satisfied our theoretical computational constraints, there's a strong chance that newer and larger masked LMs will perform even better on our dataset. Testing on other LMs will also further solidify our dataset as a useful analysis tool for object-related common sense. We also want to do a more in-depth statistical analysis of how DistilBERT performed by function, perhaps grouping functions to get coarser granularity to understand which functions need the most fine-tuning for a LM to succeed.

With the recent advent of multi-modal LMs like PaLM-E and GPT-4 (OpenAI, 2023), our research interests are quickly shifting towards utilizing these models for grounded common-sense understanding. It is possible these may be more aware of physical limitations due to images (and in PaLM-E's case, robotic policy) in the training data. While these models do have some ability to explain their decision making process, there is much to discover in terms of the models' full capabilities. We are also interested in examining few-shot fine-tuning with syntactic and semantic information to improve both common-sense performance and the model's ability to explain itself. Our hope is that combining new multi-modal models with linguistic insight will make a more trust-worthy model that can be successfully deployed in disaster-relief missions.

We set out to discover if LMs can provide the type of *apriori*, commonsense knowledge of the functions of various objects, especially those deemed important to robot-based, disaster relief missions. This is important because this technology could lead to replacing humans with robots in dangerous scenarios that have little room for error. We systematically identified the function each object plays in our domain, then created two tasks to test the granularity of a LM's ability to differentiate between these functions. DistilBERT performed quite strongly on our tasks, validating our proof of concept. Even when removing the bias of human-generated wrong answers, we still obtained strong results indicating that DistilBERT has significant knowledge about our domain. We are finding new avenues to expand our research into using more advanced LMs in tandem with resources encoding linguistic knowledge to improve collaborative, physically situated human-robot dialogue.

Acknowledgments

We would like to thank the anonymous reviewers for their comments, questions, and suggestions. This material is based in part upon work supported by the National Science Foundation under Grant No. IIS-2024878. This material is also based on research that is in part supported by the Army Research Laboratory, Grant No. W911NF2120076, and by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the Air Force Research Laboratory (AFRL), DARPA, or the U.S. Government.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

William Chen, Siyi Hu, Rajat Talak, and Luca Carlone. 2022. Extracting zero-shot common sense from large language models for robot 3d scene understanding.

Shubhashis Roy Dipta, Mehdi Rezaee, and Francis Ferraro. 2022. Semantically-informed hierarchical event modeling.

David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Matthew Marge, Claire Bonial, Brendan Byrne, Taylor Cassidy, A William Evans, Susan G Hill, and Clare Voss. 2017. Applying the wizard-of-oz technique to multimodal human-robot dialogue. *arXiv preprint arXiv:1703.03714*.

Chris McCormick and Nick Ryan. 2019. Bert word embeddings tutorial.

Ian Niles and Adam Pease. 2003. Linking lixicons and ontologies: Mapping wordnet to the suggested upper merged ontology. In *Ike*, pages 412–416.

OpenAI. 2023. Gpt-4 technical report.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. 2022. Llm-planner: Few-shot grounded planning for embodied agents with large language models.

SimpleMTOD: A Simple Language Model for Multimodal Task-Oriented Dialogue with Symbolic Scene Representation

Bhathiya Hemanthage , Christian Dondrup, Phil Bartie, Oliver Lemon†
School of Mathematical and Computer Sciences
Heriot-Watt University, †Alana AI
{hsb2000, c.dondrup, phil.bartie, o.lemon}@hw.ac.uk

Abstract

SimpleMTOD is a simple language model which recasts several sub-tasks in multimodal task-oriented dialogues as sequence prediction tasks. SimpleMTOD is built on a large-scale transformer-based auto-regressive architecture, which has already proven to be successful in uni-modal task-oriented dialogues, and effectively leverages transfer learning from pretrained GPT-2. In-order to capture the semantics of visual scenes, we introduce both local and *de-localized* tokens for objects within a scene. De-localized tokens represent the type of an object rather than the specific object itself and so possess a consistent meaning across the dataset. SimpleMTOD achieves a state-of-the-art BLEU score (0.327) in the Response Generation sub-task of the SIMMC 2.0 test-std dataset while performing on par in other multimodal sub-tasks: Disambiguation, Coreference Resolution, and Dialog State Tracking. This is despite taking a minimalist approach for extracting visual (and non-visual) information. In addition the model does not rely on task-specific architectural changes such as classification heads.

1 Introduction

Multimodal conversational agents have witnessed a rapidly growing level of interest among the conversational AI community as well as within the computer vision community. Most multimodal conversational datasets to-date are an extension of visual question answering (VQA) (Das et al., 2016; Hudson and Manning, 2019). Consequently building upon the success of other visio-linguistic tasks such as VQA, state-of-the-art multimodal conversational agents commonly depend on non-autoregressive models (Wang et al., 2020; Murahari et al., 2019) most of which are based on BERT (Devlin et al., 2018).

However, dialogues with such systems significantly differ from what the conversational AI community has typically viewed as a multi-turn dialogue. First, most of the current multimodal dialogue datasets are focused on querying the visual content whereas *external knowledge bases* have been an integral part of traditional unimodal dialogue datasets (Budzianowski et al., 2018; Galley et al., 2019). Second, in traditional unimodal dialogues, co-reference resolution (explicitly or implicitly) plays a major role within the dialogues. Additionally, state-of-the-art unimodal conversational agents predominantly rely on GPT-based auto-regressive models (Radford et al., 2018) due to their proven language generation capabilities (Peng et al., 2020; Hosseini-Asl et al., 2020; Ham et al., 2020). The SIMMC 2.0 (Kottur et al., 2021) task-oriented dialogue dataset bridges this gap between multimodality and the more traditional view of a multi-turn dialogue. Due to the simultaneous presence of signals from multiple modalities, which a user can refer to at any point in the conversation, the multimodal task-oriented dialogues proposed in the SIMMC 2.0 are challenging compared to both text-only counterparts and *image querying* dialogue datasets.

In spite of the inherent complexity of multimodal dialogues, we propose SimpleMTOD, recasting all sub-tasks into a simple language model. SimpleMTOD combines the idea of *'de-localized visual object representations'* with a GPT-like auto-regressive architecture. The idea of de-localized representations stems from the analogous process of *de-lexicalization* that has been extensively used in task-oriented dialogues. In de-lexicalization Mrksic et al. (2017), slot-values such as *vegan* are replaced by a more general abstracted token such as *food-type*. Likewise, when de-localized, objects are represented by the catalogue type of the object instance rather than the instance itself. These de-localized tokens then possess a consistent meaning throughout the dataset.

The main objective this work is to evaluate the

Proceedings of the 15th International Conference on Computational Semantics pages 293–304
June 21–23, 2023. ©2023 Association for Computational Linguistics

effectiveness of de-localized object representations within SimpleMTOD. Despite the simplicity, SimpleMTOD achieves the state-of-the-art BLEU score of 0.327 for assistant response generation in the SIMMC2.0 test-std [1] dataset . Furthermore, the model achieves an accuracy of 93.6% in Multimodal Disambiguation (MM-Disambiguation), Object-F1 of 68.1% in Multimodal Co-reference Resolution (MM-Coref), and 87.7% (Slot-F1) and 95.8 (Intent-F1) in Multimodal Dialogue State Tracking (MM-DST). Other than the proposed benchmark settings, we also evaluate SimpleMTOD in an end-to-end setting. Major contributions of our work are as follows:

- We formalise notion of *multimodal task oriented dialogues* as an end-to-end task.
- We propose a GPT-based simple language model combined with visual object delocalization and token based spatial information representation, that addresses four subtasks in multimodal dialogue state tracking with a *single architecture*.
- We analyse the behaviour of our model using salience scores from the Ecco (Alammar, 2021) framework, which provide an intuition into which previous token mostly influence predicting the next token.

2 Background

Traditional task-oriented dialogue datasets consist of a dialogue corpus, a dialogue ontology with a pre-defined set of slot-value pairs, and annotations required for related sub-tasks in a set of domains (Budzianowski et al., 2018). The SIMMC 2.0 dataset follows a similar structure and contains dialogues in both the fashion and the furniture domains. However, in the SIMMC 2.0 multimodal dialogue corpus, each dialogue is also associated with an image representing the scene where each dialogue takes place. A *scene* is made by re-arranging a known set of items (objects) in different configurations. Along with the raw-image, the dataset provides a file (scene JSON) containing details of the images such as objects and relationships between objects. Furthermore, a meta-data file contains visual and non-visual attributes of objects that recur within a scene.

2.1 Benchmark Tasks

Multimodal Disambiguation: In real-world conversations, references made by humans related to objects or entities can be ambiguous. For example, consider *A: Blue trousers are priced at $149.99. U: What about the red ones?*, in a setting where there are multiple red trousers. In these situations, there is insufficient information available for co-reference resolution. This task is aimed at identifying such ambiguous scenarios, given the dialogue history.

Multimodal Co-reference Resolution: The goal of this task is to resolve any reference in a user utterance to canonical object ids of the object as defined per each scene (see image in Figure 1(b)). Users may refer to 1) dialogue context 2) visual context, or 3) both.

Mutltimodal Dialogue State Tracking: Similar to unimodal DST, this tracks the belief states of users across multiple turns. The belief state consists of an intent, slot-value pairs, and user requested slots.

Assistant Response Generation Given the user utterance, ground-truth APIs, and ground-truth cannonical object ids (with meta-data), the model needs to generate a natural language response describing objects as *observed and understood* by the user.

3 Methods

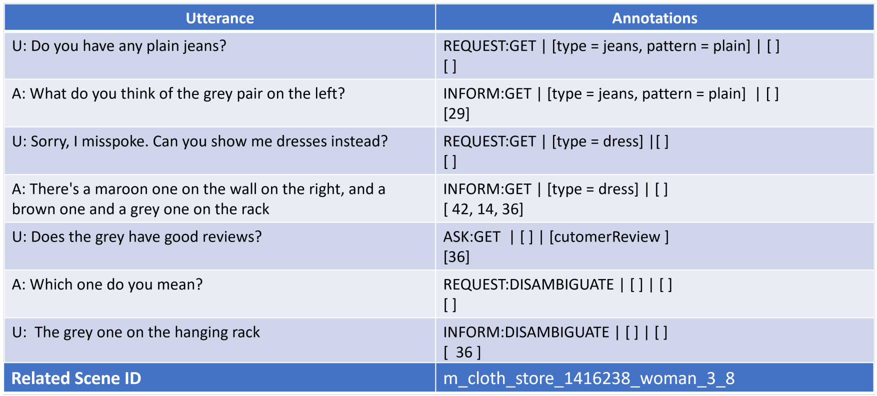
In the first part of this section, we model multimodal task oriented dialogues as a sequence generation task. We define the problem in a more general setup and discuss some empirical limitations applied to the model.

3.1 Multimodal Task-Oriented Dialogues

Similar to unimodal setting, we view dialogue state (belief-state) tracking, action prediction, and response generation to be the core components of multi-modal task-oriented dialogues. However, outputs of each of the sub-tasks should be conditioned not only on the dialogue history, but also on the associated scene.

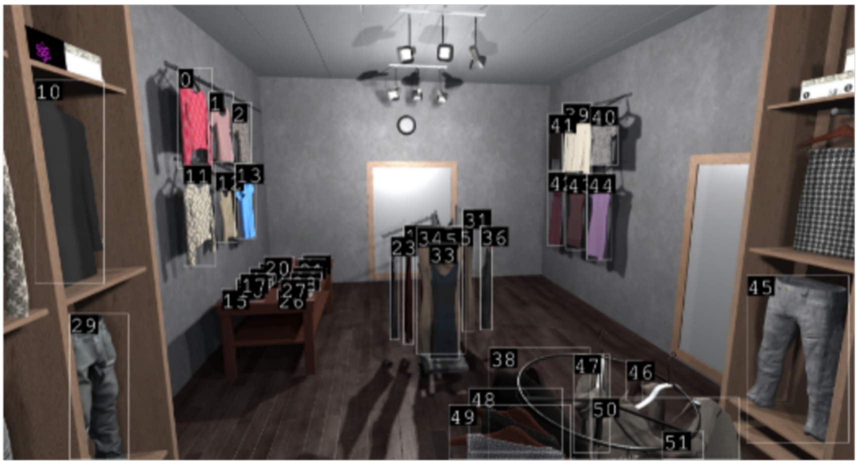
Multimodal dialogues consist of multiple turns. In a turn t, there exists an associated visual scene V_t, the user-provided input U_t and the system-generated response S_t. Theoretically, the dialogue context can be denoted as

[1] The testing dataset (test-std) is not publicly available and was part of the SIMMC 2.0 challenge used for scoring the submitted systems.

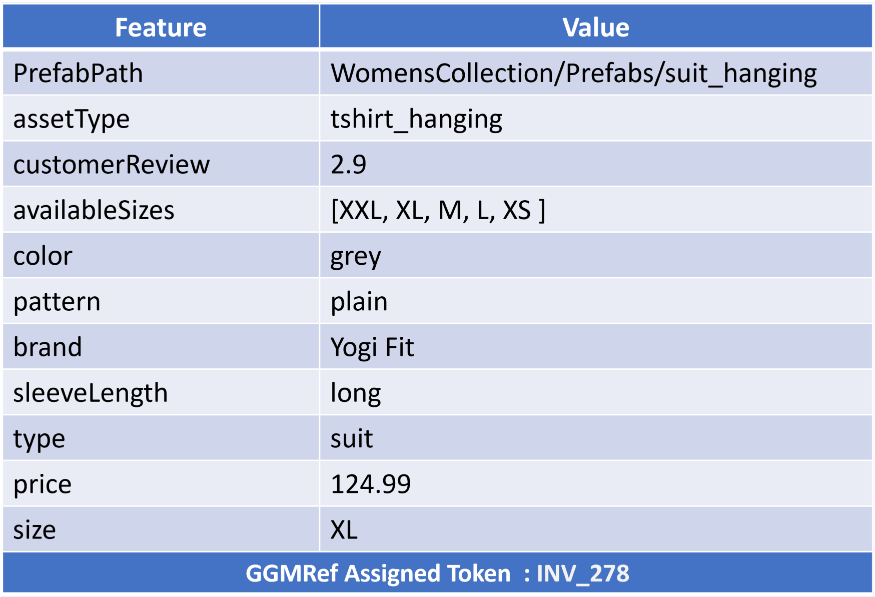


Utterance	Annotations
U: Do you have any plain jeans?	REQUEST:GET | [type = jeans, pattern = plain] | [ ] [ ]
A: What do you think of the grey pair on the left?	INFORM:GET | [type = jeans, pattern = plain] | [ ] [29]
U: Sorry, I misspoke. Can you show me dresses instead?	REQUEST:GET | [type = dress] |[ ] [ ]
A: There's a maroon one on the wall on the right, and a brown one and a grey one on the rack	INFORM:GET | [type = dress] | [ ] [42, 14, 36]
U: Does the grey have good reviews?	ASK:GET | [ ] | [cutomerReview] [36]
A: Which one do you mean?	REQUEST:DISAMBIGUATE | [ ] | [ ] [ ]
U: The grey one on the hanging rack	INFORM:DISAMBIGUATE | [ ] | [ ] [36]
Related Scene ID	m_cloth_store_1416238_woman_3_8

(a)



(b)



Feature	Value
PrefabPath	WomensCollection/Prefabs/suit_hanging
assetType	tshirt_hanging
customerReview	2.9
availableSizes	[XXL, XL, M, L, XS]
color	grey
pattern	plain
brand	Yogi Fit
sleeveLength	long
type	suit
price	124.99
size	XL
GGMRef Assigned Token : INV_278	

(c)

Figure 1: Sample dialogue instance in SIMMC 2.0: a) First four turns of a sample dialogue with user and system transcript annotations. U: and A: tokens are used to differentiate user and system utterances respectively. First row of annotations are in INTENT | SLOT-VALUE | REQUEST-SLOTS format. Second row identifies referred canonical objects id tags in the utterance (e.g. [29]). It should be noted that, these object ids are specific to a given scene. In the case of user utterances, this identifier is the target of the MM-Coref task. b) Sample image with cannonical object id tags over items. This image is mapped to the dialogue by scene id. c) Single entry of the fashion object meta-data file.

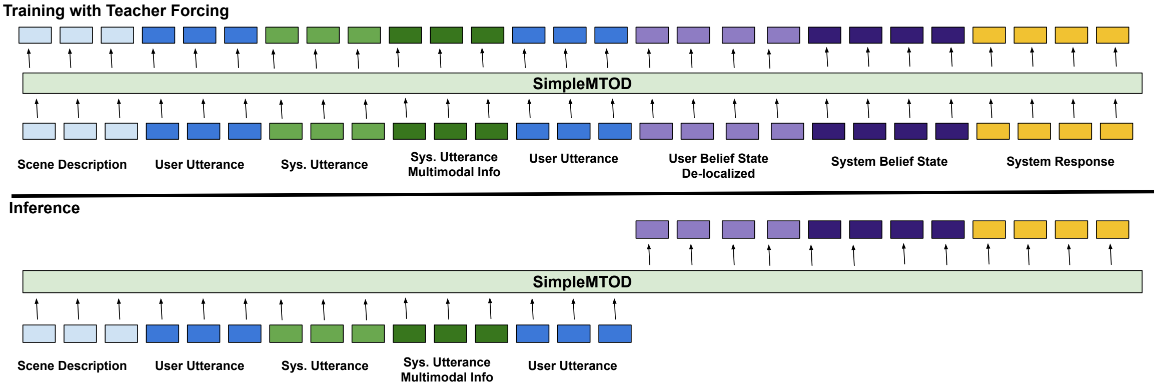


Figure 2: SimpleMTOD architecture with training and inference time setting

$C_t = [V_0, U_0, S_0|V_0, ...S_{t-1}|M_{t-1}, V_t, U_t]$. Here $S_{t-1}|M_{t-1}$ denotes that the statement S_{t-1} is associated with the representation of multimodal information such as objects viewed and mentioned to the user during that turn.

Given the context, C_t, SimpleMTOD generates the belief-state B_t:

$$B_t = SimpleMTOD(C_t) \tag{1}$$

B_t is a concatenation of intent, slot-values, requested slots, and resolved object references $MRef_t$.

However, it should be noted that, SimpleMTOD models the context as $C_t = [V_t, U_{t-n}, S_{t-n}|M_{t-n}, ...S_{t-1}|M_{t-1}, U_t,]$ where the n is the context window. Major deviations from the theoretical representation of C_t are, 1) we ignore the history of visual signals and only consider the current visual scene; 2) we consider only n previous turns in contrast to the entire dialogue.

Then, in a more generalized setting where the system have access to an external database, which can be queried,B_t would be used to retrieve database results D_t. These D_t along with context and belief states can be used to generate the system action A_t.

$$A_t = SimpleMTOD(C_t, B_t, D_t) \tag{2}$$

Action A_t is a triplet containing system intent, slot-value pairs, and details on requested slots. However, in our setup, no such database exists. Hence we model action A_t from B_t and C_t keeping $D_t = \emptyset$.

Finally, the concatenation of the context, belief state, (database results), and action is used to generate system responses S_t.

$$S_t = SimpleMTOD(C_t, B_t, D_t, A_t) \tag{3}$$

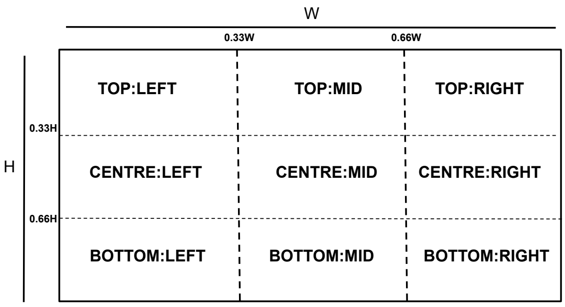


Figure 3: A scene is divided into 9 regions. Each region is identified by combination of 2 tokens.

3.2 De-localized Visual Representation

Here we discuss how visual information of a scene is represented within the SimpleMTOD as de-localized tokens and how V_t is derived from those tokens.

In the SIMMC 2.0 dataset a scene is a spatial configuration of a set of object instances. From here on we will refer to these instances simply as objects. Probable types of these objects are predefined in two meta-data files, with one for each domain. We will refer to these files as catalogues and an entry of these catalogues as a catalogue-item. See Figure1(c) for an example catalogue-item with visual and non-visual attributes defined. For benchmark tasks, non-visual attributes can be used during inference while visual attributes are not allowed. However, we use neither of these attributes in the SimpleMTOD visual representation explained below.

In our setup, we assign a unique token (eg: *INV_278*) to each catalogue-item. These catalogue-items are used as a de-localized version of objects within a scene. While these catalogue-item tokens are consistent across the entire dataset, spatial relationships associated with the objects will be lost. Therefore we encode spatial details of objects as follows: Each scene is divided into 9 regions as shown in Figure 3. Every object is assigned to a region based on the center-point of the object bounding box. Then concatenation of catalogue-item tokens and assigned region description (eg: *INV_278@TOP:LEFT*) tokens are used as object representations. A scene-description is obtained by concatenating all such tokens representing every object within a scene. This is our V_t in SimpleMTOD.

3.3 SimpleMTOD Training and Inference

For training, we follow routine causal language modeling with teacher forcing. A training sequence X_t in SimpleMTOD is obtained by concatenating all the components; context,user belief state, database results (which is null in our case), system actions and system utterance.

$$X_t = [C_t, B_t, D_t, A_t, S_t] \tag{4}$$

In terms of tokens, X_t can be denoted as $X_t = (x_t^0, x_t^1, ...x_t^{n(t)})$ when $n(t)$ represent the number of tokens in turn t. In general, the goal of the model is to learn $\rho(X)$ given $X = (x^0, x^1, ..x^i..x^n)$:

$$\rho(X) = \Pi_{i=1}^{n} \rho(x^i | x^{<i}) \tag{5}$$

For this, we train the neural network with parameterization θ minimizing the negative log-likelihood over the multimodal dialogue corpus MD where $MD = \{X_1, X_2....X_{|MD\|}\}$. However, in our setup the tokens related to scene-description V are ignored during the loss calculation. When $n(V)$ is the number of tokens related to the scene description:

$$L(D) = -\sum_{t=1}^{|MD|} \sum_{i=n(V)}^{n(t)} log\rho_\theta(x_t^i | x_t^{<i}) \tag{6}$$

During inference, the learnt parameter θ is used to predict a token at a time. Unlike training time where ground-truth tokens are used every time, generated tokens become part of the left-context. For inference, we stick to a simple greedy prediction approach with top-k=1. That is we always generate the token with highest probability as the next token.

4 Experiments

In Section 3.1 we defined an end-to-end setting for SimpleMTOD. However, some of the benchmark tasks allow more ground-truth information to be utilized during training and inference time.

For the MM-Disambiguation task, we consider two setups. In the task-specific scenario, we train the model to predict YES or NO tokens directly from context C_t. In the end-to-end setup, we consider the label to be YES only if the system intent predicted is to Disambiguate. Two similar setups are considered for MM-Coref as well. It should be noted that end-to-end version of SimpleMTOD predicts de-localized tokens with spatial information and we obtain the canonical object id by reversing the de-localization process explained in Section 3.2. If multiple objects were found in the same region with same catalogue-item token, the area of the object bounding box is used as a tie-breaker. In the case of assistant response generation, the benchmark task defined in SIMMC 2.0 allows ground-truth system belief state to be used as an input. Therefore, we evaluate both from action response generation as well as end-to-end setting.

4.1 Baselines

We consider 2 baselines which were provided as part of the SIMMC2.0 challenege.

GPT-2: This extends Ham et al. (2020) to multi modal task-oriented dialogues, encoding objects in a scene using canonical object ids concatenated with the token OBJECT_ID. For the MM-Disambiguation task, a classification head is used, while other tasks are modeled in a generative manner.

Multimodal Transformer Networks (MTN): Adapts Le et al. (2019) (only) for the MM-DST and Response Generation sub-tasks [2]. In contrast to the auto-regressive modeling of SimpleMTOD, MTN uses an encoder-decoder architecture.

4.2 Training and Evaluation

We follow the experimental setup of the SIMMC 2.0 challenge with same dataset-splits, inference time limitations, and performance metrics. See Appendix:B for details. It should be noted that the test-std split of the SIMMC2.0 dataset is not publicly available and is a held-out set for evaluating

[2] MTN-SIMMC2 implementation https://github.com/henryhungle/MTN/tree/simmc2

Model	Intent-F1	Slot-F1	Request Slot-F1	Joint Accuracy
GPT-2 Baseline	94.5	81.7	89.6	44.6
MTN-SIMMC	94.3	74.8	85.4	28.3
SimpleMTOD$_{Sub}$	**95.8**	83.3	89.7	57.3
SimpleMTOD	94.0	**85.8**	**91.7**	**63.1**

Table 1: Evaluation results for MM-DST task on Devtest split

submissions to SIMMC2.0 challenge. Therefore, the final version of our model could only be evaluated on the dev-test split. However, the prior version of the model SimpleMTOD$_{Sub}$, which did not encode region information or scene information, was submitted to the SIMMC2.0 challenge.

5 Results

Model	Accuracy	Object-F1
GPT-2 Baseline	73.5	36.6
SimpleMTOD$_{Sub}$	**92.17**	67.6
SimpleMTOD	92.12	**73.5**

Table 2: Accuracy and Object-F1 scores for MM-Disambiguation and MM-Coref tasks on Devtest split.

Model	BLEU
GPT-2 Baseline	0.192
MTN-SIMMC	0.217
SimpleMTOD$_{Sub}$	0.43
SimpleMTOD(ground truth actions)	**0.49**
SimpleMTOD	0.45

Table 3: BLEU scores for Assistant Response Generation task on Devtest split.

MM-Disambiguation As shown in Table 2 and Column 2 of Table 4, SimpleMTOD$_{Sub}$ achieves accuracy scores of 92.17% and 93.6 on devtest and test-std respectively when trained to predict YES/NO tokens. This is a 27% relative improvement over the GPT-2 based baseline with a classification head. Furthermore, we evaluate the model on the MM-Disambiguation task as part of the end-to-end model. based on the system intent predicted by the model. Here, we consider any *INFORM:DISAMBIGUATE* prediction as a YES. This approach demonstrates a very similar accuracy score of 92.12. The best performing model (94.5% : Team-6) on test-std, ensembles two models trained on RoBERTa and BART [3].

MM-Coref Table 2 and the Third column of the Table 4 show the MM-Coref Object-F1 scores of on devtest and test-std respectively. SimpleMTOD achieved 68.2 (54% relative gain over baseline) in test-std dataset and 67.6 (84% gain) on the devtest split. While there is no information available on Team-2's leading solution, the BART-based model of Team-4 which is trained end-to-end with task-specific heads achieves 75.8% on this task.

MM-DST Despite being a simple language model, both our Intent-F1 (95.8%) and Slot-F1 (87.7%) scores on test-std split are comparable with complex visual-language models. Furthermore, as in Table 1, there is significant improvement in the Joint Accuracy scores from 57.3% to 63.1% when positional information is used.

Response Generation A prior version of the model, SimpleMTOD$_{Sub}$ achieves a state-of-the-art BLEU score of 0.327 on the test-std split of the SIMMC2.0 dataset. This is in comparison with models which rely on sophisticated feature extraction processes. In our view, the simplified representation of visual information preserves and complements the generative capabilities of pre-trained models. Furthermore, as shown in Table 3, SimpleMTOD achieves a BLEU score of 0.49 on devtest when the ground-truth actions are used. The end-to-end version of SimpleMTOD also achieves a BLEU score of 0.45. It should be noted that this is an improvement over the $SimpleMTOD_{Sub}$ model score of 0.43. This indicates the importance of associating region related information.

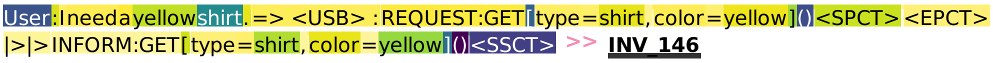
6 Discussion

In order to understand the behaviour of SimpleMToD, we use gradient-based salience (Atanasova et al., 2020) provided with the Ecco framework (Alammar, 2021). Using Ecco, we inspect salience

[3] This is based on the description provided at: https://github.com/NLPlab-skku/DSTC10_SIMMC2.0

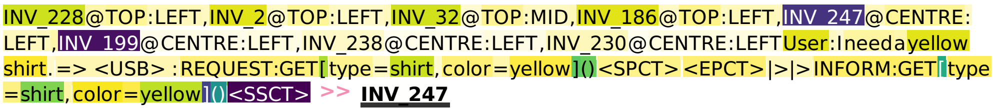
Model	MM-Disam'n	MM-Coref	DST		Response Generation
	Accuracy	Object-F1	Intent-F1	Slot-F1	BLEU
GPT-2 Baseline	73.5	44.1	94.1	83.8	0.202
MTN - Baseline	NA	NA	92.8	76.7	0.211
Team-2	NA	**78.3**	96.3	88.4	NA
Team-5	93.8	56.4	**96.4**	89.3	0.295
Team-6	**94.7**	59.5	96.0	**91.5**	0.322
SimpleMTOD$_{Sub}$	93.6	68.2	95.8	87.7	**0.327**

Table 4: Test-std results for SIMMC2.0 Challenge. NA denotes model is not applicable to the particular sub-task. Test-std split of SIMMC2.0 dataset is held-out set, which is not publicly available and used to evaluate submissions in SIMMC2.0 challenge. An earlier version of the system, SimpleMTOD$_{Sub}$, without scene information, was submitted for the evaluation.



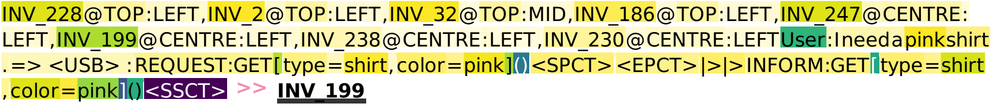
```
User:Ineedayellowshirt.=> <USB> :REQUEST:GET[type=shirt,color=yellow]()<SPCT> <EPCT>
|>|>INFORM:GET[type=shirt,color=yellow]()<SSCT> >> INV_146
```

Figure 4: Salience score heat-map when predicting the token *INV_146* for utterance *I need a yellow shirt* without scene information. Darker colors represents higher salience score. See Figure:8 in appendix for actual values



```
INV_228@TOP:LEFT,INV_2@TOP:LEFT,INV_32@TOP:MID,INV_186@TOP:LEFT,INV_247@CENTRE:
LEFT,INV_199@CENTRE:LEFT,INV_238@CENTRE:LEFT,INV_230@CENTRE:LEFTUser:Ineedayellow
shirt.=> <USB> :REQUEST:GET[type=shirt,color=yellow]()<SPCT> <EPCT>|>|>INFORM:GET[type
=shirt,color=yellow]()<SSCT> >> INV_247
```

Figure 5: Salience scores heat-map *with scene information* when predicting the token *INV_247* in utterance *I need a yellow shirt*. See Figure:9 in appendix for actual values



```
INV_228@TOP:LEFT,INV_2@TOP:LEFT,INV_32@TOP:MID,INV_186@TOP:LEFT,INV_247@CENTRE:
LEFT,INV_199@CENTRE:LEFT,INV_238@CENTRE:LEFT,INV_230@CENTRE:LEFTUser:Ineedapinkshirt
.=> <USB> :REQUEST:GET[type=shirt,color=pink]()<SPCT> <EPCT>|>|>INFORM:GET[type=shirt
,color=pink]()<SSCT> >> INV_199
```

Figure 6: Salience score heat-map when predicting the token *INV_199* for modified utterance *I need a pink shirt* See Figure:10 in appendix for actual values

Token Feature	INV_146	INV_199	INV_247
Color	yellow	pink	yellow
Type	shirt	shirt	shirt

Table 5: Relevant catalogue items represented by tokens INV_146, INV_199, INV_247. None of these metadata were explicitly presented to the model.

scores for all the tokens in the left side of the token of interest. In the heat-maps presented in this section, darker colors mean a higher salience score. It should also be noted that the model assigns high salience scores on separator tokens (such as $< USB >$, *[,]*) that define the structure of the generation. While proper attention to the structure is of paramount importance, our **discussion focuses on salience scores assigned to the rest of the tokens, which represent the semantics** of the multimodal conversations.

Effect of De-localization and Scene Descriptions: The introduction of de-localized tokens significantly improves the Object-F1 of MM-coref and joint accuracy of MM-DST. Accordingly, we first analyse the behaviour of the model when predicting co-references. Figures 5 and 4 show example utterances with and without scene descriptions respectively. In the case where scene description is not provided, the model puts a high salience on tokens 'yellow' and 'shirt', and predicts the token INV_146 which represents a yellow color shirt as shown in Table 5. (It should be noted that none of the metadata shown in the diagram are provided to the model explicitly and the model figures this out from globally consistent use of tokens). However, in this case, a particular catalogue item INV_146 is not present in the scene. When we observe the confidence values of the prediction from the last layer (shown in Table 6), it can be seen that the model is not quite certain about the prediction with

Original(color=yellow)	INV_247 (92.63)	INV_199 (7.17)	INV_155(0.08)
Original w/o desc.	INV_146(13.75)	INV_247 (13.04)	INV_249 (12.60)
Modified(color=pink)	INV_199(99.79)	INV_247 (0.19)	INV_235(<0.01)

Table 6: For the example utterances discussed, we inspected top-3 tokens and their confidence scores.

13.75 for INV_146 and 13.04 for INV_247, both of which represent yellow shirts. This is to indicate that even though the model has learnt to associate object attributes necessary for co-reference resolution, it lacks information to be certain about the prediction. To this end, we provide the model with a scene description as described in 3.2. When the scene descriptions are provided, SimpleMTOD correctly predicts the token INV_247 with 92.63% confidence and high salience score over the same token from the scene description, as well as tokens 'shirt' and 'yellow'.

Additionally from Figure 5 it can be noted that INV_199 also shows a high salience score. From the metadata, we can see it is a pink color shirt. However, there is a significant salience score over the token 'yellow' that results in generating the correct token INV_247 over INV_199 (which is the second ranked token with only had 7.17 confidence). Extending the analysis, we modified the original utterance to *"I need a pink shirt"* and generated the next token, and SimpleMToD accordingly predicted the token INV_199 (with high confidence of 99.79%) as observed in Figure 6.

Effect on Intent prediction: Even though scene descriptions play a key role in overall belief tracking as described earlier, the Intent-F1 score drops from 95.8% to 94.0% when the scene descriptions are encoded. In order to understand the effect, we inspect salience scores when predicting the user intent. It can be observed that when the scene descriptions are omitted, higher salience scores are assigned to the user utterance suggesting more focus on that. However, when the scene information is included, salience scores assigned to the utterance decreased to an extent, resulting in wrong predictions in certain cases. This is to indicate that scene descriptions are either redundant or act as a distractor when we consider intent-detection, which explains reduction in score. Furthermore, this behaviour aligns with our intuition that the intent parts of the user utterances are predominantly language-driven. Figure 7 shows an example where omitting the scene information produces the correct intent of *REQUEST:COMPARE*, whereas our final version of SimpleMTOD wrongly predicted the intent as *ASK:GET*

7 Related Work

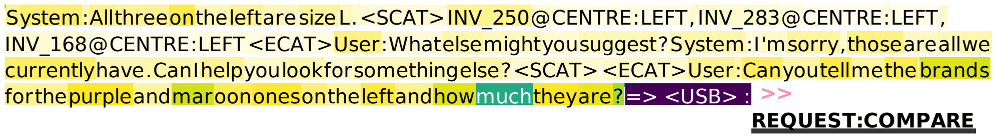
Peng et al. (2020); Hosseini-Asl et al. (2020); Ham et al. (2020) are closely related to our work as they all model task-oriented dialogues in an end-to-end manner with GPT-2-like large-scale transformer-based architectures. However, all those models focus on *text-only* task-oriented dialogues. The GPT-2 adaptation (Kottur et al., 2021), which is provided as a baseline along with the SIMMC2.0 dataset, is also closely related to our work. However, this baseline represents visual objects by canonical ids and demonstrates subpar results to our model in all four tasks.

Generative encoder-decoder models (Liang et al., 2020; Zhao et al., 2017) are a promising alternative to decoder-only (GPT-2 based) dialogue models that have been extensively investigated in unimodal task-oriented dialogues. The MTN-baseline (Le et al., 2019), which we compare to, is based on the encoder-decoder architecture. While being inferior with respect to performance in both the tasks considered, this model involves sophisticated feature extraction process.

Mrksic et al. (2017) coined the term 'delexicalization' for abstraction in neural dialogue state tracking tasks. This idea has been extensively used in goal oriented dialogues. Our notion of delocalized object representation is influenced by this work.

8 Conclusion

We explore a simple, single generative architecture (SimpleMTOD) for several sub-tasks in multimodal task-oriented dialogues. We build on large-scale auto-regressive transformer-based language modeling, which has been effectively utilized in task-oriented dialogues, and formalize the multimodal task-oriented dialogue as a sequence prediction task. Our model employs a 'de-localization' mechanism for visual object representation that ensures the consistency of those tokens throughout the dataset. Furthermore, we encoded spatial infor-



System:Allthreeontheleftaresize L. <SCAT> INV_250@CENTRE:LEFT, INV_283@CENTRE:LEFT,
INV_168@CENTRE:LEFT <ECAT> User:Whatelsemightyousuggest? System:I'msorry,thoseareallwe
currentlyhave.CanIhelpyoulookforsomethingelse? <SCAT> <ECAT> User:Canyoutellmethebrands
forthepurpleandmaroononesontheleftandhowmuchtheyare? => <USB> : >>
REQUEST:COMPARE

Figure 7: Salience score heat-map when predicting the correct intent token *REQUEST:COMPARE* for the dialogue turn with final utterance *"Can you tell me the brands for the purple and maroon ones on the left and how much they are?"* without providing scene information

mation of object instances with a very small number of special (globally consistent) tokens. Despite the simplicity in representing visual information, our model demonstrates comparable or better performance with models that heavily rely on visual feature extraction, on four multimodal sub-tasks in the SIMMC2.0 challenge.

9 Future Directions

Most current vision-language research relies on fusing pixel-level vision information with token-level language representations. However, their applicability for dialogues where the language is sophisticated remain sparsely studied. In contrast, we explore a symbolic approach for representing visual information and combining it with auto-regressive language models. While we rely on smaller scale models (with 17 million parameters), our work is readily extendable for large language models (LLMs). Unlike pixel level visual representations, special tokens representing visual information being more similar to the word tokens which the LLMs area trained on, symbolic visual representation would facilitate effective transfer learning.

SimpleMTOD represents visual information using carefully designed input tokens. Capturing these information through semantic scene-graphs, which would provide richer representation, and fusing them with LLMs would be an interesting future direction of research for multimodal dialogues. Development in knowledge-graph based language grounding would complement this line of work.

Acknowledgements

This work is partially supported by the European Commission under the Horizon 2020 framework programme for Research and Innovation (H2020-ICT-2019-2, GA no. 871245), SPRING project, https://spring-h2020.eu

References

J Alammar. 2021. Ecco: An Open Source Library for the Explainability of Transformer Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A Diagnostic Study of Explainability Techniques for Text Classification. In *EMNLP (1)*, pages 3256–3274. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. 2016. Visual Dialog. *CoRR*, abs/1611.08669.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Cite arxiv:1810.04805.

Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. 2019. Grounded Response Generation Task at DSTC7.

DongHoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In *ACL*, pages 583–592. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A Simple Language Model for Task-Oriented Dialogue. Cite arxiv:2005.00796Comment: 22 Pages, 2 figures, 16 tables.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *CVPR*, pages 6700–6709. Computer Vision Foundation / IEEE.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. SIMMC 2.0: A Task-oriented Dialog Dataset for Immersive Multimodal Conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hung Le, Doyen Sahoo, Nancy F. Chen, and Steven C. H. Hoi. 2019. Multimodal Transformer Networks for End-to-End Video-Grounded Dialogue Systems. In *ACL (1)*, pages 5612–5623. Association for Computational Linguistics.

Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. 2020. MOSS: End-to-End Dialog System Framework with Modular Supervision. In *AAAI*, pages 8327–8335. AAAI Press.

Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young. 2017. Neural Belief Tracker: Data-Driven Dialogue State Tracking. In *ACL (1)*, pages 1777–1788. Association for Computational Linguistics.

Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. 2019. Large-scale Pretraining for Visual Dialog: A Simple State-of-the-Art Baseline. *CoRR*, abs/1912.02379.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model. *CoRR*, abs/2005.05298.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Yue Wang, Shafiq R. Joty, Michael R. Lyu, Irwin King, Caiming Xiong, and Steven C. H. Hoi. 2020. VD-BERT: A Unified Vision and Dialog Transformer with BERT. *CoRR*, abs/2004.13278.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR*, abs/1910.03771.

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskénazi. 2017. Generative Encoder-Decoder Models for Task-Oriented Spoken Dialog Systems with Chatting Capability. In *SIGDIAL Conference*, pages 27–36. Association for Computational Linguistics.

A SIMMC 2.0 Dataset

The SIMMC 2.0 dataset (released under CC-BY-NC-SA-4.0 licence) [4] consists of three major components:

- **Dialogue Data:** Includes system and user utterance with relevant annotations. Figure 1(a) provide first 4 turns of a sample dialogue.
- **Scene Data:** Set of scenes representing environments in which dialogues take place. Figure 1(b) provide the scene related to the dialogue segment shown in Figure 1(a). Other than raw-images , an json file associated with each image provides detail of objects, such as bounding boxes and spatial relationships (left of, right of, over, under) among objects.
- **Meta-data:** acts as a catalogue of items related to the dialogue corpus. Scene images are made-up by positioning instances of catalogue items in different configurations. Entries contain both visual and non-visual attributes of each item. Visual attributes of items from the meta-data file are not allowed to be used during inference.Figure 1(c) shows a single entry in meta-data file.

A.1 Data Statistics

Split	# of Dialogues
Train (64%)	7307
Dev (5%)	563
Test-Dev(15%)	1687
Test-Std (15%)	1687

Table 7: Number of dialogues in each split.

B Training and Evaluation

Task	**Metric**
MM-Disambiguation	Accuracy
MM-Coref	Object-F1
MM-DST	Intent-F1
	Slot-F1
DST+Coref	Joint Accuracy
Response Generation	BLEU-4

Table 8: Evaluation metrics used for different tasks in SIMMC 2.0

[4] https://github.com/facebookresearch/simmc2

We conduct our experiments with the SIMMC 2.0 (Kottur et al., 2021) dataset. Further, we follow the experimental setup of the SIMMC 2.0 challenge with the same dataset splits, inference time limitations, and performance metrics.

Implementation: We conduct our experiments using PyTorch Huggingface's transformers (Wolf et al., 2019). All SimpleMTOD model variants were initialized with Open AI GPT-2 pretrained weights and exhibits computational speed identical to Open AI GPT-2. We use Adam optimizer (Kingma and Ba, 2014) with default parameter of Huggingface's AdamW implementation ($lr = 1e - 3, eps = 1e - 6, weight\_decay = 0$).

We use the GPT-2 tokenizer for encoding user and system utterances. However, we noticed that the default tokenizer encoder mechanism chunks special tokens introduced for visual object representation. Therefore, we implemented an encoding mechanism which selectively skips the default byte-pair encoding for object tracking tokens.

Evaluation: We use the same evaluation metrics and evaluation scripts provided with the SIMMC2.0 challenge. Table 8 shows metrics that are used for evaluating each benchmark task.

C Salience scores

For the discussion we use input X gradient (IG) method from (Alammar, 2021) as suggested in (Atanasova et al., 2020). In the IG method of input saliency, attribution values are calculated across the embedding dimensions. With the values from embeddings dimension, the L2 norm is used to obtain a score per each token Then resulting values are normalized by dividing by the sum of the attribution scores for all the tokens in the sequence. Here we provide actual salience scores for heat-maps provided in the discussion in Section: 6.

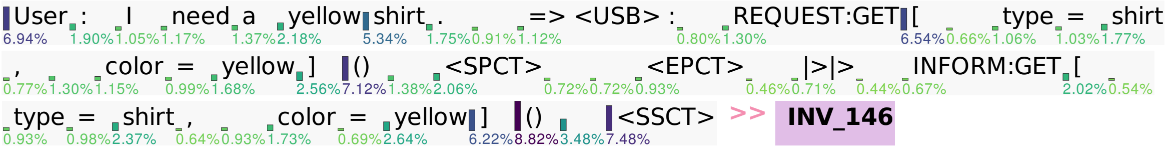


Figure 8: Salience score when predicting the token *INV_146* for utterance *I need a yellow shirt* without scene information.

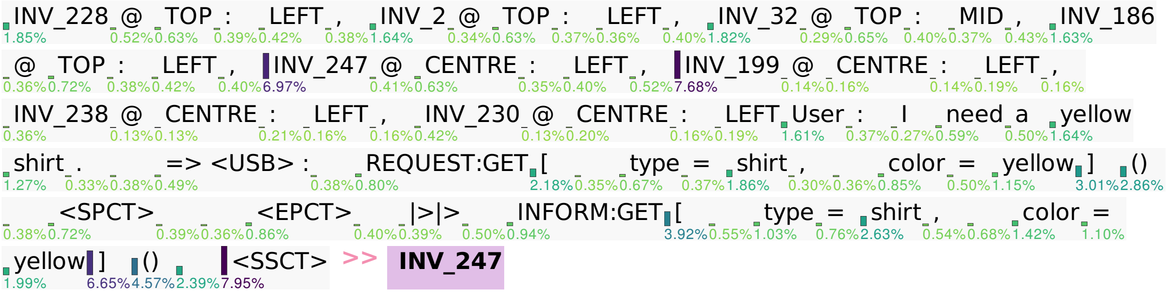


Figure 9: Salience scores *with scene information* when predicting the token *INV_247* in utterance *I need a yellow shirt*.

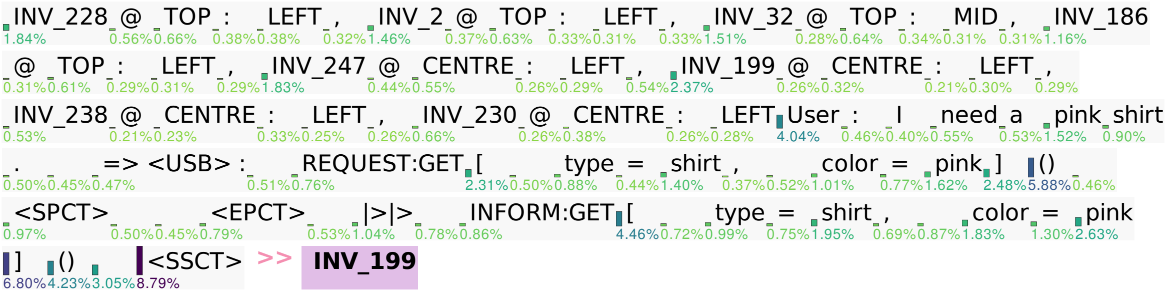


Figure 10: Salience scores when predicting the token *INV_199* for modified utterance *I need a pink shirt*

Grounding and Distinguishing Conceptual Vocabulary Through Similarity Learning in Embodied Simulations

Sadaf Ghaffari and **Nikhil Krishnaswamy**
Situated Grounding and Natural Language (SIGNAL) Lab
Department of Computer Science, Colorado State University
Fort Collins, CO, USA
{sadafgh,nkrishna}@colostate.edu

Abstract

We present a novel method for using agent experiences gathered through an embodied simulation to ground contextualized word vectors to object representations. We use similarity learning to make comparisons between different object types based on their properties when interacted with, and to extract common features pertaining to the objects' behavior. We then use an affine transformation to calculate a projection matrix that transforms contextualized word vectors from different transformer-based language models into this learned space, and evaluate whether new test instances of transformed token vectors identify the correct concept in the object embedding space. Our results expose properties of the embedding spaces of four different transformer models and show that grounding object token vectors is usually more helpful to grounding verb and attribute token vectors than the reverse, which reflects earlier conclusions in the analogical reasoning and psycholinguistic literature.

1 Introduction

A common critique of modern large language models (LLMs) is that they lack *understanding* in the sense of being able to link an utterance to a specific communicative intent (Bender and Koller, 2020). This shortcoming is often characterized as being due to a lack of ability to *ground* or link lexical items to real-world entities such as classes of objects, or associated properties or actions. For instance, a modern generative LLM like ChatGPT[1] may be able to generate coherent text describing an object (e.g., "a *coconut* has a hard, often hairy outer shell"), without any inherent underlying conceptualization of what the item actually *is*.

Crucially, these underlying conceptualizations necessarily invoke other modalities. Existing approaches to grounding in NLP typically treat the problem as one of making the correct kind of link between text and another modality, usually images (Socher et al., 2014; Yatskar et al., 2016; Zhu et al., 2020, 2021). However, still images do not capture the wealth of information humans receive when interacting with objects or experiencing events, and video data requires orders of magnitude more data and computational power to effectively process. Additionally, humans do not use vision alone as their only non-linguistic modality.

As humans develop object concept representations and map them to associated nouns, they are also learning to individuate objects from the perceptual flow not just based on visual features but also based on experience that includes interacting with them in real time (Spelke, 1985; Spelke et al., 1989; Spelke, 1990; Baillargeon, 1987). Gentner (2006) argues that Talmy (1975)'s findings on variability in verbal semantics helped to explain why nouns are typically learned before words for verbs or other properties. Concrete nouns are more easily "groundable" not just because of their visual manifestations but also because of their physical presences that leave traces in the world, and these physical properties provide a scaffold on which to build representations of related concepts that are supervenient upon understanding of objects.

In this paper we take an *embodied simulation* approach to grounding, using a virtual environment to create experiences for an agent interacting with objects. We show that similarity learning over data gathered during the agent's experience in the virtual world can not only make comparisons between objects, but also appears to learn information pertaining to more abstract properties of the objects. Fig. 1 shows a schematic view of our overall approach. We map token vectors from different transformer-based LLMs into the resulting representation space, and show that with just a few samples, grounding noun representations alone is

[1] https://chat.openai.com

Proceedings of the 15th International Conference on Computational Semantics pages 305–317
June 21–23, 2023. ©2023 Association for Computational Linguistics

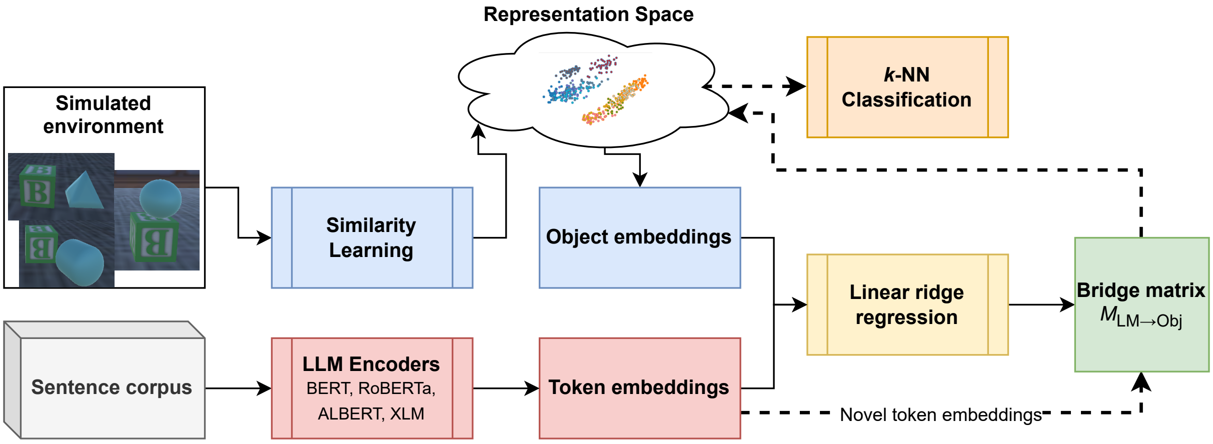


Figure 1: Overview of grounding architecture. In this figure, M denotes the computed affine transformation matrix between language model (LM) and object classifier (Obj) space. Similarity learning in this figure is performed only over a subset of the available classes (see Sec. 3.2). The solid lines depict the flow of information used to "train" or compute the affine transformation "bridge" matrix, and the dashed lines depict the flow of information of novel "test" samples, including transformation by the precomputed bridge matrix.

helpful for subsequent grounding of verbal tokens, abstract properties, and attributive terms, but that grounding verbal or attributive token representations is less helpful for subsequent grounding of object concepts.

2 Related Work

Multiple works in cognitive science have identified contrastive mechanisms, and the ability to analogize by applying previous experiences to novel scenarios, as a cornerstone of problem-solving (Gentner, 1983; Forbus et al., 1995; McLure et al., 2010; Hofstadter and Sander, 2013; Smith and Gentner, 2014; Lovett and Forbus, 2017).

In visual analogy, Hill et al. (2019) created analogies by contrasting relational structure. For solving Raven's Progressive Matrices (RPMs) (Raven, 1936), Małkiński and Mańdziuk (2022) applied a generalization of the Noise Contrastive Estimation (NCE) algorithm (Gutmann and Hyvärinen, 2010). Wu et al. (2018) performed feature learning using visual similarity via unsupervised learning at the instance-level with NCE. Oh Song et al. (2016) used deep feature embedding based on lifted structure loss, and evaluated their method via clustering and retrieval tasks on images from unseen classes. Bell and Bala (2015) trained a Siamese CNN with contrastive loss (Hadsell et al., 2006) to learn an embedding space of interior design images and applied the embeddings to image search applications like finding visually similar products across categories.

Since evaluating AI agents in physical environment can be expensive, many works have used both embodied and non-embodied simulations to explore language learning. Hermann et al. (2017) combined reinforcement and unsupervised learning to teach agents to correlate linguistic symbols with physical percepts and action sequences. However, this still-computationally-expensive method required millions of training episodes. The SNARE benchmark (Thomason et al., 2022) was evaluated on grounding to objects but not in context or under interaction. Tucker et al. (2021) demonstrated an emergent clustering of semantic tokens from a (non-embodied) continuous representation space and Tucker et al. (2022) extended that method with an application of an information bottleneck. Our work integrates concepts from the above areas: embodied simulation environments, language grounding using both situated and linguistic context, and emergent semantic categorization.

Merullo et al. (2022) examined 2D and 3D visual and interactive data for learning object affordances and found that 3D and interactive data performed better. Ebert et al. (2022) extracted verbal semantics from object trajectories in 3D space, but focused only on verbs whereas we examine nouns, verbs, and attributes. We show that objects and properties can also be encoded by object trajectories or behavior in 3D space, using a stacking task that exposes richer correlations between object properties and behaviors. Like us, Patel and Pavlick (2022) investigated word grounding but they evaluated on within-domain concepts (e.g., learning "left" to help ground "right" where we investigate how, say, learning "sphere" can help ground "round") in a grid world (our world representation is continuous), and where their transformation passed input through a whole LLM, our transformation is a simple affine map between embedding spaces. Lazaridou et al. (2015) mapped vision embeddings to language via ridge regression, but their Multimodal Skip-Gram used static word vectors, not contextualized vectors from transformers.

Pezzelle et al. (2021) evaluated the representations of transformer models and found that multimodal representations better align with human judgments in the domain of concrete nouns, but not abstract terms. Our work arrives at a related conclusion using cross-model transfer.

3 Methodology

Our methodology comprises two primary components: similarity learning to create a representation space of objects by making comparisons between geometric properties, and linear projection to ground language representations to this space.

3.1 Data

We use the dataset from Ghaffari and Krishnaswamy (2022), in which an agent in a simulated environment built on the VoxWorld platform (Krishnaswamy et al., 2022), stacks 9 different types of *theme* objects[2] on top of a cube. Each object's behavior when stacked is different, based on its geometric structure and therefore *affordances* (Gibson, 1977). For instance, a cube, if placed correctly on another cube, will remain stacked, while a sphere placed in the same position will roll off and keep moving. An egg will likely do the same, but the direction of motion may be subtly different based on the symmetry of the object. The dataset contains 10,000 total samples, each with 43 numerical values describing the behavior of the objects in the course of this stacking task: theme object type; object orientation before the agent acts upon it; numerical action describing the placement of the theme relative to the destination object; resulting spatial relations between the two objects; object orientation after the action; and position of the theme relative to the destination object before action, immediately after action, and after the world physics are applied to the scene. See Ghaffari and Krishnaswamy (2022) for further details. This information about object behaviors and trajectories in space, unlike still images, *situates* the objects in an embodied environment and encodes richer information than visuals alone do. The dataset does also contain images but these are not used here.

Two of the object types in the data, *cylinder* and *cone*, have both flat sides and round edges, and as this distinction strongly affects the behavior of these objects when stacked (i.e., given proper placement, a cylinder or cone will stack on top of a cube but only if also placed in the correct orientation), the dataset preserves these distinctions nicely, and we split the cone and cylinder samples into "flat-side-down" and "round-edge-down" for similarity learning of properties (Sec. 3.2).

3.2 Similarity Learning of Object Properties

Since comparing pairs of examples plays a role in analogy-making, we apply deep pair-based learning to compare structural object properties. The main goal in deep pair-based learning techniques is to learn an embedding space where embeddings of similar samples are closer together and dissimilar samples are pushed apart, after the projection of input space to the embedding metric space. In our case, the trained model should be able to infer contrasts and comparisons between different structural properties of objects (in this case *flatness* and *roundness*), and apply it to novel objects based on commonalities in behavior and relational structure.

In training, we consider only samples of *cube*, *rectangular prism*, *pyramid*, and *small cube* that stacked successfully, and samples of *capsule*, *sphere*, and *egg* that did not. For testing data, we take a test split of the same object classes, and also samples of *cone* and *cylinder*. These samples behave differently according to, among other things, their orientation when placed. We split cone and cylinder instances into "flat-side-down" (stacked successfully) and "round-edge-down" (did not stack successfully). Therefore we train on 7 classes and evaluate on 11 classes, including 4 never seen in training.

To train, we take 500 samples of each training class, zero-center the data and make it unit variance. Our model architecture consists of 4 1D convolutional layers (32, 32, 64, and 64 units, respectively, with filter size 3, stride length 1). The network applies ReLU activation to the output feature maps, with a max-pooling layer after the first two convolutional layers. The final convolutional layer output

[2] *cube*, *sphere*, *cylinder*, *capsule*, *small cube*, *egg*, *rectangular prism*, *pyramid*, and *cone*.

is flattened, followed by an L_2 normalized dense layer.

We use multi-similarity loss (Wang et al., 2019) which uses two iterative steps: pair-mining and weighting. This approach considers both self-similarity and relative similarity to collect more informative pairs, and takes a weighted combination of selected positive and negative pairs. Like other pairwise-based losses, this loss function maximizes the distance between dissimilar examples and minimizes it between similar examples.

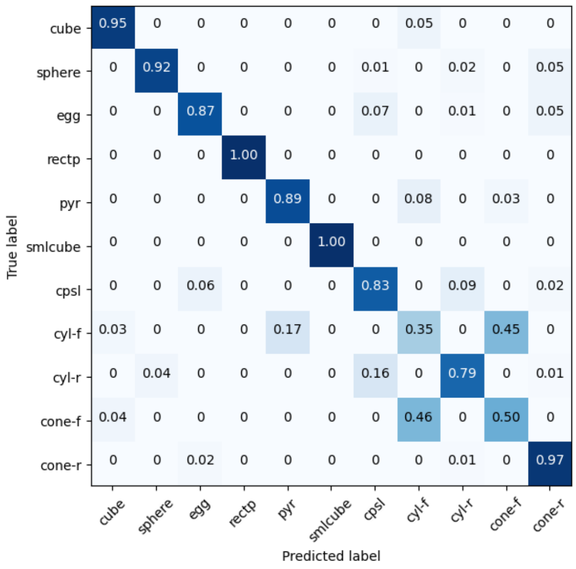
Equation 1 provides the formulation of the multi-similarity loss function:

$$\frac{1}{m}\sum_{i=1}^{m}\{\frac{1}{\alpha}\log[1+\sum_{k\in P_i}1+e^{-\alpha(S_{ik}-\lambda)}] \\ +\frac{1}{\beta}\log[1+\sum_{k\in N_i}1+e^{\beta(S_{ik}-\lambda)}]\}, \quad (1)$$

where N_i represents negative pairs (samples from different classes) in the batch while P_i denotes positive pairs (samples from the same class). S_{ik} represents element (i,k) of the similarity matrix, indicating the similarity of two samples $\{x_i, x_k\}$, $S_{ik} := f(x_i;\theta)\cdot f(x_k;\theta)$ where f is the neural network with parameters θ. The cosine embedding size is 64.

We use Adam optimization (Kingma and Ba, 2015) with a learning rate of 5×10^{-6}, with batch size 70, and train for 20 epochs. Training was performed on a Mac M1 Max with Metal acceleration. In every mini-batch, 10 inputs ($m = 10$) from each of the 7 training classes are randomly sampled. In Equation 1, $\alpha = 2$ (weight for positive pairs), $\beta = 40$ (weight for negative pairs), $\lambda = 0.5$ (used to weight the distance). Margin $\epsilon = 0.1$ is used to remove easy positive and negative pairs such that negative pairs are sampled if they are greater than $(\min_{y_i=y_k}(S_{ik})-\epsilon)$ where $\min_{y_i=y_k} S_{ik}$ represents the positive pair with the lowest similarity, and positive pairs are sampled if they are less than $(\max_{y_i\neq y_k}(S_{ik})+\epsilon)$ where $\max_{y_i\neq y_k} S_{ik}$ represents the negative pair with the highest similarity. y denotes the one-hot label vectors.

Since during training only *7* types of flat-sided and round objects are used, the model learns to output an embedding that represents pure round and flat objects samples in the cosine space. The extracted embeddings are indexed. Given that the index of the embedding space represents only purely round or flat objects, we consider 100 test samples each from all *11* classes (seen and unseen) and find the closest matches to the test samples using a nearest neighbour search ($K = 10$).



	cube	sphere	egg	rectp	pyr	smlcube	cpsl	cyl-f	cyl-r	cone-f	cone-r
cube	0.95	0	0	0	0	0	0	0.05	0	0	0
sphere	0	0.92	0	0	0	0	0.01	0	0.02	0	0.05
egg	0	0	0.87	0	0	0	0.07	0	0.01	0	0.05
rectp	0	0	0	1.00	0	0	0	0	0	0	0
pyr	0	0	0	0	0.89	0	0	0.08	0	0.03	0
smlcube	0	0	0	0	0	1.00	0	0	0	0	0
cpsl	0	0	0.06	0	0	0	0.83	0	0.09	0	0.02
cyl-f	0.03	0	0	0	0.17	0	0	0.35	0	0.45	0
cyl-r	0	0.04	0	0	0	0	0.16	0	0.79	0	0.01
cone-f	0.04	0	0	0	0	0	0	0.46	0	0.50	0
cone-r	0	0	0.02	0	0	0	0	0	0.01	0	0.97

Figure 2: Confusion matrix on the test split of 11 objects. Only 7 pure flat and round objects are used during training. `cyl-f` = cylinder, flat side down; `cyl-r` = cylinder, round edge down; likewise for `cone-f/r`. The values shown in the matrix are normalized between 0 and 1.

Similarity Learning Results Fig. 2 shows the confusion matrix for nearest neighbor search on the test split of objects, using 100 test samples per class. Interestingly, even though the model was not trained on any *cone* and *cylinder* instances, it is still able to not only match them to the correct object type, but also to the correct orientation. Where confusions arise, it is between different flat-sided objects and different round objects, but never across these categories. In other words, this model can capture and distinguish the main distinguishing concepts—roundness and flatness—in the different object classes, and draw comparisons between them across classes. It also applies what is already learned to novel objects to find similar examples with respect to these concepts. Overall classification accuracy is 82%.

3.3 Grounding Conceptual Vocabulary

First, we extracted the embeddings of 800 object test samples from the learned object space. These were 64D embeddings that defined the object representation space, and objects clustered into two broad regions defining the "flat-sided" and "round-sided" objects (see Fig. 3).

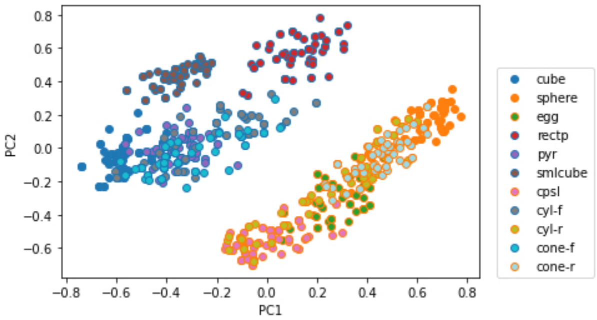


Figure 3: PCA of test object embeddings. Points outlined in blue represent “flat” object embeddings. Points outlined in orange represent “round” object embeddings.

Individual embedding vectors of different instances of the same object type form a region defining the object representation where some subset of these vectors form the region’s spanning set; Ethayarajh (2019) observed similar phenomena in the representations of contextualized token vectors from LLMs, suggesting there exists a structure-preserving mapping between equivalent regions in different embedding spaces.

To assess this, we needed to generate appropriately contextualized vector representations of terms to ground to the object representation space. For this we turned to OpenAI’s ChatGPT model to rapidly generate a sentence corpus. ChatGPT was given prompts to generate short, unique sentences that would explicitly mention the objects by name and describe their behavior in a stacking task (e.g., “*Write 40 short sentences about how cubes can be stacked*”). In the process, ChatGPT also generated mentions of properties of the objects (*flat/round*), associated behaviors (*stack/roll*), properties of the resulting structure (*stable/unstable*), and resulting state of the structure (*stand/fall*). We generated 40 sentences describing each object type, plus 20 sentences each for *block* and *ball*, synonyms for *cube* and *sphere*. In total, a 440-sentence corpus was generated.

We then took the most frequently-occurring domain-relevant terms (these were the object names and aforementioned related conceptual terms) and extracted the word-level embeddings for each occurrence. We extracted word embeddings using the BERT, RoBERTa, ALBERT (all 768D), and XLM (2,048D) base models. Embeddings were creating by summing over the encoder hidden states of the last four encoder layers. Where tokenization split the target word into multiple tokens, the individual contextualized token embeddings were averaged to create a single embedding.

To actually ground the word embeddings into the object space, we used a simple affine transformation. We took 5 contextualized embeddings of each target word, paired each with an embedding for the object whose name occurs in the sentence the target word came from, and use them to compute an affine transformation from LLM space to object embedding space, using a ridge regressor that minimizes the mean squared-error distance between the paired embeddings. The resulting transformation matrix serves as a “bridge” between the two representation spaces. This affine transformation technique has previously been used to compare image embeddings from different vision models (McNeely-White et al., 2022) and to map information from monolingual LLMs into multilingual LLMs (Nath et al., 2022). Here we apply this technique in a cross-modal setting.

We perform iterative experiments, starting by using only a subset of the different words and objects to compute the mapping, and incrementally add conceptual vocabulary to improve the quality of the calculated transformation. We evaluate the transformation by transforming word vectors for concepts not used in computing the transformation matrix and seeing if those embeddings cluster with the correct set of objects that bear those properties, have those affordances, etc. The order in which object concepts are introduced follows the order we used previously in Ghaffari and Krishnaswamy (2022), with the exception of moving *cylinder* and *cone* to the end, due to their exclusion from initial training of the similarity learning model, and pairing one flat-sided with one round object (e.g., *pyramid* + *capsule*) at each step.

As a final step, a “hint” is provided by adding 5 embedding pairs that explicitly include the concept to be grounded to the computation of the transformation. We evaluate this by transforming new instances of that concept into the object embedding space and seeing where they cluster. We quantify the clusters of different concepts when transformed into the object space using separation of cluster centers and a K-nearest neighbor (KNN) classifier with $K = 5$.

4 Results of Conceptual Grounding

For illustrative purposes, let us first examine the concepts of “flat” and “round” using word embeddings drawn from XLM, the best-performing model

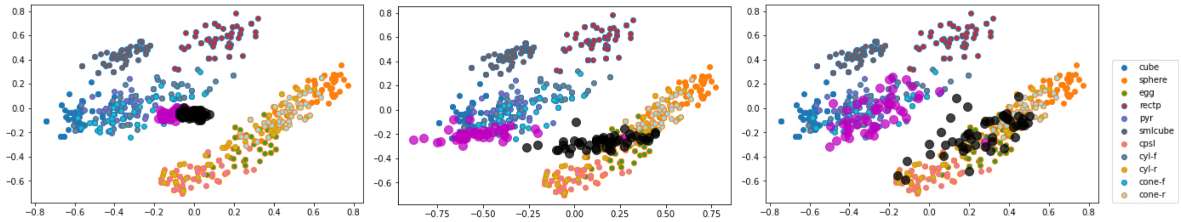


Figure 4: PCA of "flat" (pink) and "round" (black) test embeddings from XLM mapped into object embedding space. L: with mapping computed using only information about *cubes*, *spheres*, and *eggs*. C: using information about all objects. R: using all objects and a 5-sample "hint" about "flat" vs. "round."

in this domain when "hinting" is used. Further results from other models are given in the Appendix.

In Fig. 4, we see word embeddings for "flat" and "round" transformed into the object embedding space. At first (the left figure), when only information about cubes, spheres, and eggs are used to compute the mapping, there is only a slight separation between the two transformed embedding clusters and neither term clusters cleanly with either flat or round objects. When information about all objects is used to compute the transformation (center), the two word embedding clusters distinctly separate, with most "flat" embeddings clearly overlapping with the flat-sided objects, *mutatis mutandis* "round" embeddings and the round objects. Finally (right), the "hint" is provided, by explicitly pairing a small set of 5 "flat" or "round" word embeddings to object embeddings whose type appears in a generated sentence collocated with the target word (e.g., "*The cubes were flat on all sides, making it easy to stack them neatly.*"). With this hint we see that the "flat" and "round" word embeddings more completely overlap with the objects that have the respective attributes.

When little information about related object concepts is provided when computing the mapping from LLM space to object embedding space, the transformed clusters of contrasting terms share a high level of similarity in object space, but as more information about related object concepts is introduced into the transformation, the separation of the transformed novel concept clusters start to cleanly separate and become distinguished from each other. Fig. 5 shows the mean similarity of the transformed clusters of attributive, verbal, and object synonym terms as different object terms are mapped into the object space, using embeddings drawn from the four different LLMs. Fig. 6 shows the same change in the similarity between cluster centers, but this time evaluated over the transformed *object* terms when the transformation is computed using attributive and verbal terms. In both plots, dashed lines show where an explicit "hint" is given about specific concepts. We see that just by using a few samples of each concept and projecting them into object space using an affine transformation, grounding object terms is helpful in distinguishing the meaning of terms denoting related properties, attributes, and verbs, but grounding the more abstract concept vocabulary first does not usually cause the transformed clusters of object terms to separate before explicit hinting is provided, reflecting the psycholinguistic hypothesis of Gentner (1983).

Table 1 shows the results of the KNN classifier over the transformed attributive and verbal word embeddings, both when the transformation was computed using only object information (top section) and with "hints" about the attributive concepts (bottom). Table 2 shows KNN classifier results over transformed object embeddings without hints about the objects, and with. We report macroaveraged F1 scores, so that successful performance on high support classes does not obscure poor performance on low support classes. Numbers in parentheses show how much "hinting" helped improve performance of the particular model on the concept in question. *Block* and *ball* are included in both the "object" test set and the "predicate" test set (even though they are not predicative terms in this sense), because these terms were not used in computing the affine mapping in either case. They are included as synonyms for *cube* and *sphere*. Further discussion is provided in Sec. 5.

5 Discussion

Separation of conceptual clusters In Fig. 5, we can see that for object concept vectors from certain models, as information about certain other concepts is included in the transformation from LLM space to object space, the centers of the conceptual clusters in question start to organically separate. This is particularly true for ALBERT object word vectors and to some extent XLM and BERT vectors. In

	flat/round	**stack/roll**	**stable/unstable**	**stand/fall**	**block/ball**
Models	$N = 103$	$N = 56$	$N = 22$	$N = 10$	$N = 30$
BERT	0.89	0.16	**0.58**	0.60	0.33
RoBERTa	0.34	0.16	0.29	0.37	0.67
ALBERT	**0.92**	**0.65**	**0.58**	**0.89**	0.60
XLM	0.73	0.53	0.37	0.29	**0.79**
BERT+hint	0.96 (+0.07)	0.78 (+0.62)	0.91 (+0.63)	**1.00** (+0.40)	0.93 (+0.60)
RoBERTa+hint	0.90 (+0.56)	0.89 (+0.73)	**1.00** (+0.71)	**1.00** (+0.63)	0.90 (+0.23)
ALBERT+hint	0.89 (-0.03)	0.85 (+0.20)	0.86 (+0.28)	**1.00** (+0.11)	0.66 (+0.06)
XLM+hint	**0.98** (+0.25)	**1.00** (+0.47)	0.73 (+0.36)	**1.00** (+0.71)	**0.97** (+0.18)

Table 1: Macroaveraged KNN F1 over transformed attribute/verb/synonym word embedding test sets (mapping computed using object embeddings). Numbers in parentheses show performance increase with "hinting."

Models	**cube/sphere**	**pyr/cpsl**	**cyl-f/r**	**cone-f/r**	**block/ball**
BERT	0.77	0.46	0.34	0.40	**0.83**
RoBERTa	0.81	0.44	0.40	0.49	0.55
ALBERT	**0.88**	**0.88**	**0.81**	**0.78**	0.46
XLM	0.40	0.46	0.49	0.36	0.55
BERT+hint	0.97 (+0.20)	**1.00** (+0.54)	0.78 (+0.44)	0.84 (+0.44)	0.93 (+0.10)
RoBERTa+hint	0.81 (±0.00)	0.94 (+0.50)	0.78 (+0.38)	0.87 (+0.38)	0.90 (+0.35)
ALBERT+hint	0.88 (±0.00)	0.94 (+0.06)	**0.87** (+0.06)	0.88 (+0.10)	0.66 (+0.20)
XLM+hint	**1.00** (+0.60)	0.97 (+0.51)	0.81 (+0.32)	**0.91** (+0.55)	**0.97** (+0.42)

Table 2: Macroaveraged KNN F1 over transformed object word embedding test sets (mapping computed using attribute/verb embeddings). Numbers in parentheses show performance increase with "hinting." $N = 30$ for all.

other words, if the model already "knows" about the dual aspects of cones and cylinders, it becomes easier to distinguish an abstract concept of *flatness* from *roundness*. Clusters of transformed RoBERTa object word vectors tend not to separate very clearly until explicit hints about them are provided.

Flat/round is the easiest of the attributive or verbal concepts to distinguish, through affine transformations that include information about flat and round objects. *Stable/unstable* is a particularly hard distinction for most model representations, in part because of the low support for these terms in the training corpus but also because in the scenario captured in the simulation data and described in training sentences, the terms refer to properties not of the objects themselves, but of the objects in the context of the stacking task (i.e., spheres are not inherently "unstable" but are if someone attempts to stack them). This suggests data gathered from either stacking more objects, or from tasks involving more complex balancing acts would be useful to learn a robust interpretation of such terms.

Inverse trends are observable in Fig. 6, where we see that when the transformation includes only information about attributes and verbs, transformed BERT and XLM object word vectors for contrasting objects do not meaningfully separate until explicit hints are provided (and even then sometimes they don't separate much). Some of the RoBERTa object word clusters do appear to appreciably separate as more attribute and verb information is added to the transformation, and ALBERT object word clusters, actually at first grow *closer* as related conceptual information is added to the transformation, until suddenly separating at the provision of an explicit hint. This suggests that ALBERT, perhaps due to its smaller training size and architecture, learns vocabulary representations that are more "entangled" or that representations of flat-sided or round object words carry with them a bias toward object-related interpretations of "flat," "round," and associated terms. Meanwhile XLM and other representations of abstract vocabulary are perhaps less correlated with concrete nouns, making them less easy to ground but also in principle more compositional with less bias toward certain interpretations.

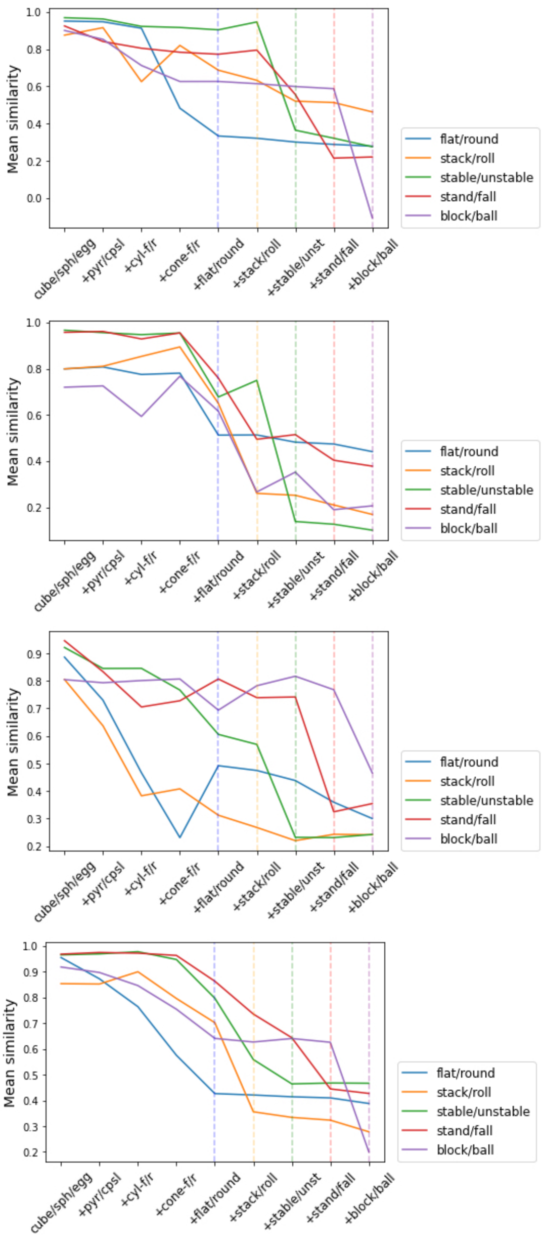


Figure 5: Separation of cluster centers for transformed (in order) BERT, RoBERTa, ALBERT, and XLM embeddings for verb and property concepts, as more information about other concepts is progressively added to compute the transformation. Dashed lines show where a "hint" is given about the concept to be grounded (denoted by the similarly-colored solid line).

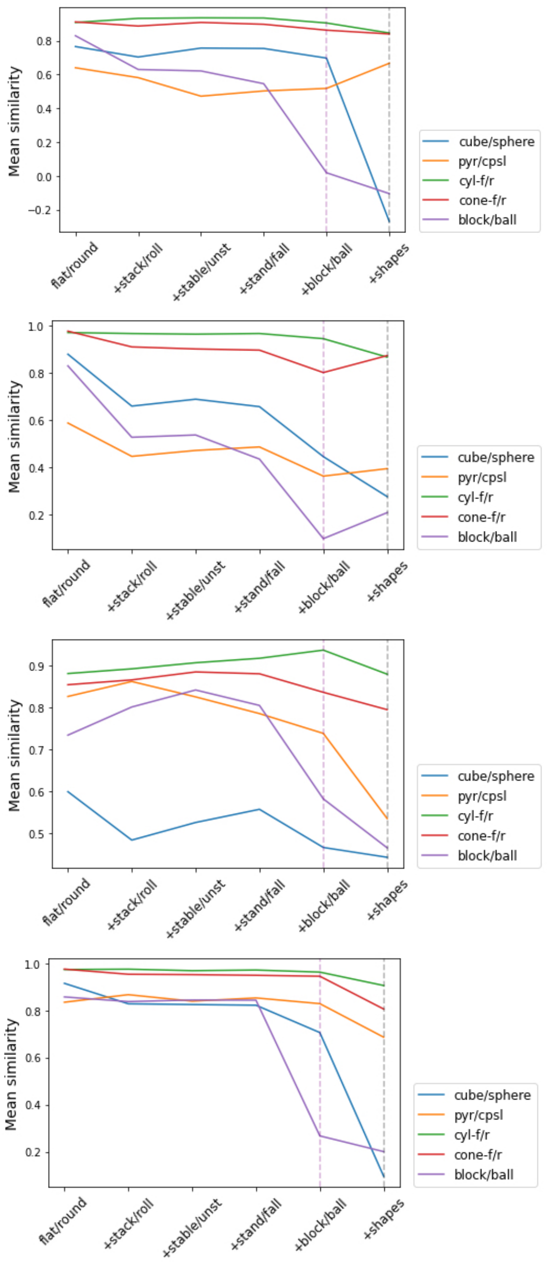


Figure 6: Separation of cluster centers for transformed embeddings for object concepts, as more information about other concepts is progressively added to compute the transformation. Format is identical to Fig. 5. `+shapes` denotes adding information about all objects.

Classification of conceptual terms With hinting, XLM vectors perform best in the term classification task. XLM is the largest of the four models and has the largest embedding size (2,048 where all other models use an embedding size of 768). Hinting typically provides the biggest boost in performance to XLM vectors, both for grounding concrete object and abstract terms. This suggests that the object concepts and the attributive/verbal concepts form distinct and possibly distant regions in the original XLM embedding space, and that an affine transformation into the object space does not always put pairs of contrastive attributes or verbs closer to distinct objects that display those respective properties. Providing hints helps all models achieve high performance by matching objects and related concepts, but the performance boost is particularly high for XLM vectors, which often perform very badly in KNN classification of some concepts (e.g., *stable/unstable*, *stand/fall*) until hints are provided. Hinting is still less helpful for transforming XLM object vectors when only previous information about attributes or verbs is provided.

Interestingly, hinting is least helpful when grounding word vectors from ALBERT, the smallest of the four models. On eight out of ten concept pairs explored, ALBERT vectors perform the best by far in the KNN classifier before any hints are provided, but providing subsequent hints makes only a small difference to classification F1, and sometimes none at all, while boosting the performance of other representation ahead of ALBERT vectors. This suggests that the object and related concept representations already share some level of correlation and possibly overlap in ALBERT embedding space. In turn, these results suggest that larger models like XLM may be better able to represent multiple word senses and figurative, non-physically-grounded usages of terms like these. However, grounding these concepts to a physical environment without some explicit "nudges" may be more challenging for larger pretrained models than smaller ones, in which the abstract concepts may already be biased toward correlations with the associated concrete object concepts. Further discussion is provided in the Appendix.

6 Conclusion and Future Work

In this paper, we have presented evidence that similarity learning over rich object behavior and trajectory data from an embodied simulation environment can create a representation space that not only successfully classifies concrete objects but can make analogical comparisons between them based on abstract properties that inhere across multiple object types. We used the resulting representation to conduct investigations into the properties of token embeddings from different LLMs by mapping them into the object space using a linear ridge regression technique. We found that computing a mapping using representations of objects/object terms correlated with increased ability to distinguish and assign related conceptual vocabulary to the right categories, but that representations from different LLMs behaved quite differently. We also observed that computing mappings using information about abstract properties was less useful for distinguishing and classifying object terms. This reflects earlier arguments from psycholinguistics and analogical reasoning, e.g., Gentner (2006)'s hypothesis that names for concrete objects should be learnable by humans very early but that associated verbs and attributes are harder.

Our approach uses numerical data that situates and embodies an agent's positioning in the environment relative to the objects it interacts with. This method allows us to build a model over rich information without visual artifacts like occlusion or perspective distortion, Prior research, e.g., Krishnaswamy and Pustejovsky (2022); Pustejovsky and Krishnaswamy (2022) has demonstrated that embodiment is also influenced by other factors like events and habitats, and that purely linguistic representations of objects, attributes, and activities may not capture these types of information. In fact, the corpus generated using ChatGPT, an unembodied language model trained solely over text, is likely not entirely representative of these aspects beyond cooccurrences between object terms, habitats, and affordances in the training data. What our embodied approach brings is a way to correlate representations extracted from unembodied models to representations learned from embodied data, and provides evidence that the ability to ground real-world entities, properties, or actions to lexical items could enable LLMs to simulate the human ability to link utterances to specific communicative intents. However, further investigation is necessary.

Since the primary objective of this research is to provide a method that achieves human-like "understanding" of communicative intents, we should note that we do not argue that human learners use the same mathematical transformations we use herein, but just that we can use them to make AI systems behave similarly.

Directions for future work include 1) investigating the effects of intra-class order when grounding tokens, e.g., introducing object concepts to the affine mapping in a different order; 2) using similarity learning over images, or images combined with the embodied data, to create the representation space; 3) using data gathered in other embodied tasks to investigate other concepts like concavity or directedness, that are not captured in this stacking task; 4) evaluating token representations directly from a decoder like a GPT-style model; and 5) directly operationalizing analogical comparisons in a real-time embodied simulation, e.g., by making an agent solve problems using analogical reasoning in a live environment.

Acknowledgments

We would like to thank our reviewers for their helpful suggestions on improving the final copy of this paper. This research was supported in part by a grant from the U.S. Army Research Office (ARO) to Colorado State University, under award #W911NF-23-1-0031. The positions expressed herein do not reflect the official position of the U.S. Department of Defense or the United States government. Any errors or omissions are the responsibility of the authors.

References

Renee Baillargeon. 1987. Object permanence in $3\frac{1}{2}$-and $4\frac{1}{2}$-month-old infants. *Developmental psychology*, 23(5):655.

Sean Bell and Kavita Bala. 2015. Learning visual similarity for product design with convolutional neural networks. *ACM transactions on graphics (TOG)*, 34(4):1–10.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Dylan Ebert, Chen Sun, and Ellie Pavlick. 2022. Do trajectories encode verb meaning? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2860–2871.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Kenneth D. Forbus, Dedre Gentner, and Keith Law. 1995. Mac/fac: A model of similarity-based retrieval. *Cognitive science*, 19(2):141–205.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170.

Dedre Gentner. 2006. Why verbs are hard to learn. *Action meets word: How children learn verbs*, pages 544–564.

Sadaf Ghaffari and Nikhil Krishnaswamy. 2022. Detecting and accommodating novel types and concepts in an embodied simulation environment. In *Proceedings of the Tenth Annual Conference on Advances in Cognitive Systems*.

James J. Gibson. 1977. The theory of affordances. *Hilldale, USA*, 1(2):67–82.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. 2017. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*.

Felix Hill, Adam Santoro, David Barrett, Ari Morcos, and Timothy Lillicrap. 2019. Learning to make analogies by contrasting abstract relational structure. In *International Conference on Learning Representations*.

Douglas R. Hofstadter and Emmanuel Sander. 2013. *Surfaces and essences: Analogy as the fuel and fire of thinking*. Basic books.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikhil Krishnaswamy, William Pickard, Brittany Cates, Nathaniel Blanchard, and James Pustejovsky. 2022. The VoxWorld platform for multimodal embodied agents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1529–1541, Marseille, France. European Language Resources Association.

Nikhil Krishnaswamy and James Pustejovsky. 2022. Affordance embeddings for situated language understanding. *Frontiers in Artificial Intelligence*, 5.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining language and vision with a multimodal skip-gram model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163, Denver, Colorado. Association for Computational Linguistics.

Andrew Lovett and Kenneth D. Forbus. 2017. Modeling visual problem solving as analogical reasoning. *Psychological review*, 124(1):60.

Mikołaj Małkiński and Jacek Mańdziuk. 2022. Multilabel contrastive learning for abstract visual reasoning. *IEEE Transactions on Neural Networks and Learning Systems*.

Matthew D. McLure, Scott E. Friedman, and Kenneth D. Forbus. 2010. Learning concepts from sketches via analogical generalization and near-misses. In *Proceedings of the annual meeting of the cognitive science society*, volume 32.

David McNeely-White, Ben Sattelberg, Nathaniel Blanchard, and Ross Beveridge. 2022. Canonical face embeddings. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(2):197–209.

Jack Merullo, Dylan Ebert, Carsten Eickhoff, and Ellie Pavlick. 2022. Pretraining on interactions for learning grounded affordance representations. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 258–277.

Abhijnan Nath, Rahul Ghosh, and Nikhil Krishnaswamy. 2022. Phonetic, semantic, and articulatory features in assamese-bengali cognate detection. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 41–53.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012.

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*.

Sandro Pezzelle, Ece Takmaz, and Raquel Fernández. 2021. Word representation learning in multimodal pre-trained transformers: An intrinsic evaluation. *Transactions of the Association for Computational Linguistics*, 9:1563–1579.

James Pustejovsky and Nikhil Krishnaswamy. 2022. Multimodal semantics for affordances and actions. In *Human-Computer Interaction. Theoretical Approaches and Design Methods: Thematic Area, HCI 2022, Held as Part of the 24th HCI International Conference, HCII 2022, Virtual Event, June 26–July 1, 2022, Proceedings, Part I*, pages 137–160. Springer.

James C. Raven. 1936. Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive. *Unpublished master's thesis, University of London*.

Linsey Smith and Dedre Gentner. 2014. The role of difference-detection in learning contrastive categories. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.

Elizabeth S. Spelke. 1985. Perception of unity, persistence, and identity: Thoughts on infants' conceptions of objects.

Elizabeth S. Spelke. 1990. Principles of object perception. *Cognitive science*, 14(1):29–56.

Elizabeth S. Spelke, Claes von Hofsten, and Roberta Kestenbaum. 1989. Object perception in infancy: Interaction of spatial and kinetic information for object boundaries. *Developmental Psychology*, 25(2):185.

Leonard Talmy. 1975. Semantics and syntax of motion. In *Syntax and Semantics volume 4*, pages 181–238. Brill.

Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. 2022. Language grounding with 3d objects. In *Conference on Robot Learning*, pages 1691–1701. PMLR.

Mycal Tucker, Huao Li, Siddharth Agrawal, Dana Hughes, Katia Sycara, Michael Lewis, and Julie A. Shah. 2021. Emergent discrete communication in semantic spaces. *Advances in Neural Information Processing Systems*, 34:10574–10586.

Mycal Tucker, Julie Shah, Roger Levy, and Noga Zaslavsky. 2022. Towards human-agent communication via the information bottleneck principle. *arXiv preprint arXiv:2207.00088*.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5022–5030.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5534–5542.

Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Multimodal joint attribute prediction and value extraction for E-commerce product. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2129–2139, Online. Association for Computational Linguistics.

Wanrong Zhu, Xin Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, and William Yang Wang. 2021. Multimodal text style transfer for outdoor vision-and-language navigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1207–1221, Online. Association for Computational Linguistics.

A Appendix: Additional Results

The following figures are provided for comparison with Fig. 4 and Table 1.

Fig. 7 shows the projection of “flat” and “round” token embeddings from **BERT** into the learned object representation space when the mapping is computed using paired embeddings of objects and object terms, but no explicit hint is provided about the meaning of “flat” and “round.” The two clusters clearly separate from each other but do not map clearly onto flat and round object representations at this stage. Fig. 8 shows the same projection after a 5-sample hint about “flat” and “round” is added to the mapping.

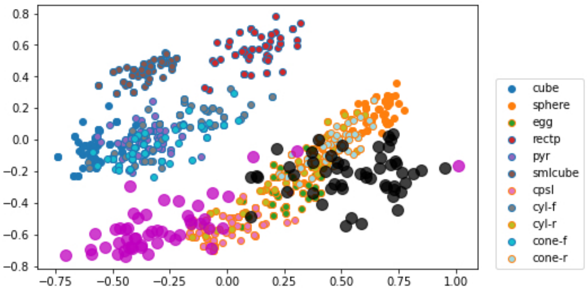


Figure 7: PCA of “flat” (pink) and “round” (black) test embeddings from BERT mapped into object representation space. Mapping is computed using information about all objects but without flat/round hinting.

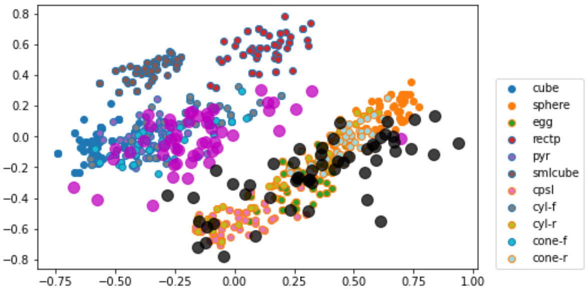


Figure 8: PCA of “flat” (pink) and “round” (black) test embeddings from BERT mapped into object representation space. Mapping is computed using information about all objects and flat/round hinting.

Fig. 9 shows the projection of “flat” and “round” token embeddings from **RoBERTa** into the learned object representation space when the mapping is computed using paired embeddings of objects and object terms, but no explicit hint is provided about the meaning of “flat” and “round.” Again, the two clusters clearly separate from each other at this stage, but the “flat” embeddings are closer to the round objects embeddings in the 64D space while the “round” embeddings are distinct from each object cluster. Fig. 10 shows the same projection after a 5-sample hint about “flat” and “round” is added to the mapping.

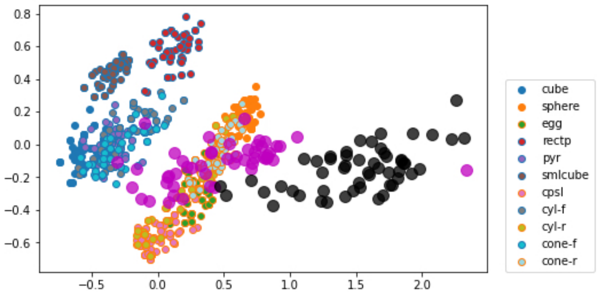


Figure 9: PCA of “flat” (pink) and “round” (black) test embeddings from RoBERTa mapped into object representation space. Mapping is computed using information about all objects but without flat/round hinting.

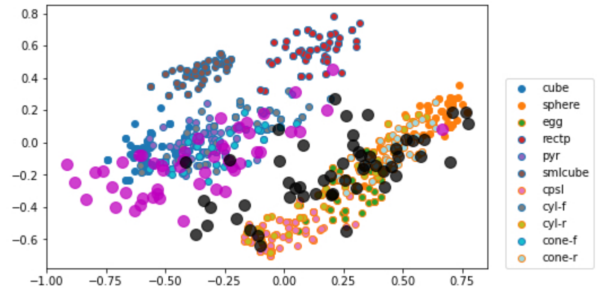


Figure 10: PCA of “flat” (pink) and “round” (black) test embeddings from RoBERTa mapped into object representation space. Mapping is computed using information about all objects with flat/round hinting.

Figs. 11 and 12 show the equivalent using the **ALBERT** “flat”/“round” token embeddings. Here, without hinting, the transformed “flat” embeddings mostly cluster with flat-sided objects and the transformed “round” embeddings mostly cluster with round objects, suggesting that in ALBERT, the representations of “flat”, “round”, and other associated object-related concepts are relatively entangled with the object terms themselves. Hinting solidifies this correlation somewhat but the effect is relatively small, as discussed in Sec. 5.

Fig. 13 shows contextualized token embeddings for *all* vocabulary items from (top to bottom) BERT, RoBERTa, ALBERT, and XLM mapped into the object representation space when the mapping is computed using information about all concepts, including hinting. For all points, the outer color denotes the token it represents and the inner color (blue or orange) indicates whether the transformed embedding clusters with flat-sided or round object representations. Therefore, a black point with an orange center indicates a “round” token embedding

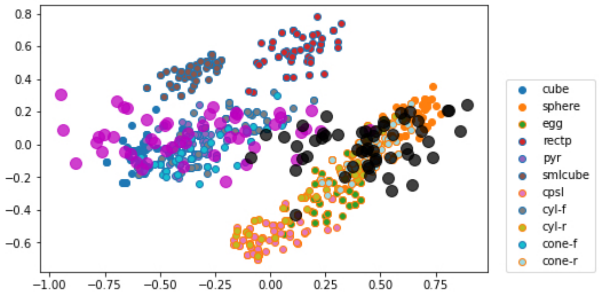


Figure 11: PCA of "flat" (pink) and "round" (black) test embeddings from ALBERT mapped into object representation space. Mapping is computed using information about all objects but without flat/round hinting.

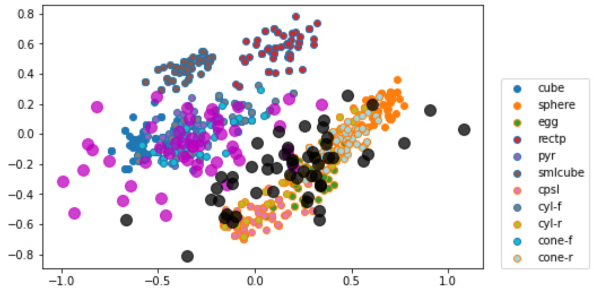


Figure 12: PCA of "flat" (pink) and "round" (black) test embeddings from ALBERT mapped into object representation space. Mapping is computed using information about all objects with flat/round hinting.

that clusters with round objects (correctly), but a black point with a *blue* center indicates one that incorrectly clusters with flat-sided objects. We see that when using the full set of concepts in computing the mapping between spaces, the larger models show the strongest correlations between correctly-mapped token embeddings and the expected set of object representations. Mapped XLM vectors show the strongest separation between the flat-related concepts and round-related concepts, while mapped ALBERT vectors display a fairly significant overlap between those correlated with flat objects and those correlated with round object (this is evident in the center of the plot "between" the two main flat and round clusters). Mapped RoBERTa and to a lesser extent BERT embeddings show a similar overall separation to mapped XLM embeddings, but with a wider dispersion in the distribution of mapped embeddings, where some (particularly in the case of RoBERTa embeddings) have a very high Euclidean distance from the two core object representation clusters to which they are compared.

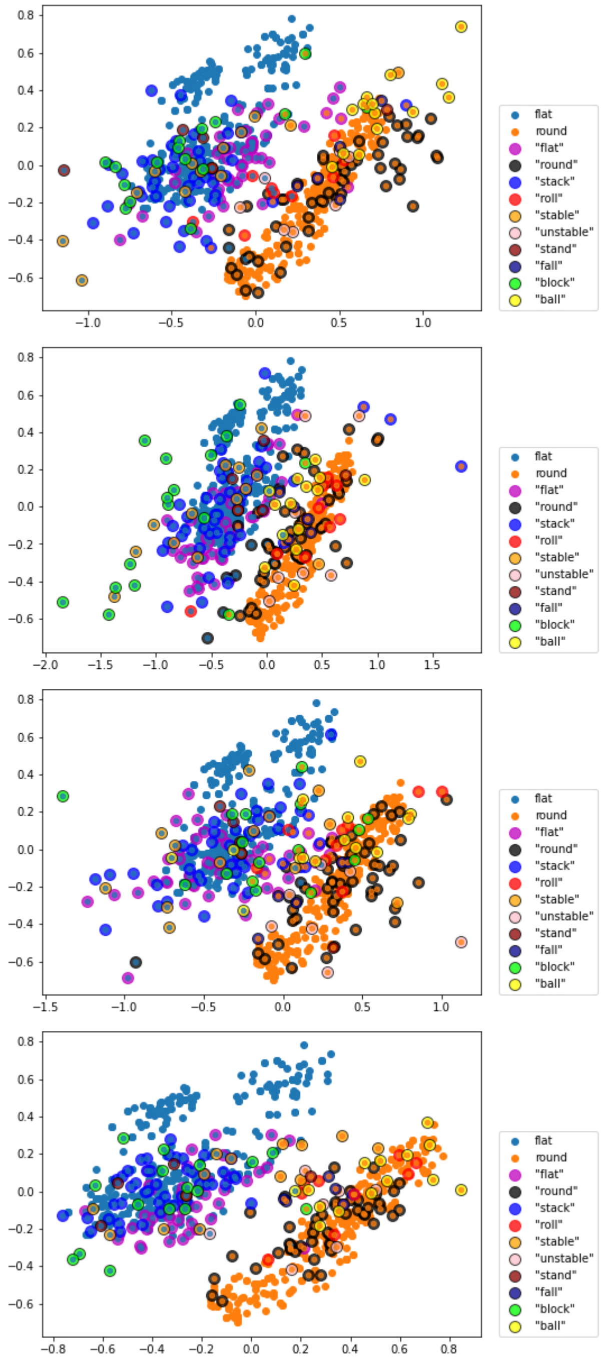


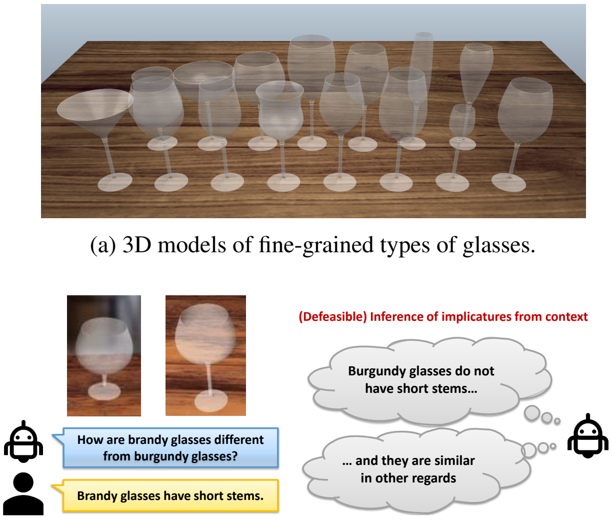
Figure 13: PCA of (top to bottom) BERT, RoBERTa, ALBERT, and XLM test word embeddings for all concepts mapped into object representation space, including hinting in the mapping. Innermost colored point indicates whether that transformed embedding clusters with flat-sided objects or round objects.

Interactive Acquisition of Fine-grained Visual Concepts by Exploiting Semantics of Generic Characterizations in Discourse

Jonghyuk Park and **Alex Lascarides** and **Subramanian Ramamoorthy**
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, UK
jay.jh.park@ed.ac.uk, alex@inf.ed.ac.uk, s.ramamoorthy@ed.ac.uk

Abstract

Interactive Task Learning (ITL) concerns learning about unforeseen domain concepts via natural interactions with human users. The learner faces a number of significant constraints: learning should be online, incremental and few-shot, as it is expected to perform tangible belief updates right after novel words denoting unforeseen concepts are introduced. In this work, we explore a challenging symbol grounding task—discriminating among object classes that look very similar—within the constraints imposed by ITL. We demonstrate empirically that more data-efficient grounding results from exploiting the truth-conditions of the teacher's generic statements (e.g., "Xs have attribute Z.") and their implicatures in context (e.g., as an answer to "How are Xs and Ys different?", one infers Y lacks attribute Z).


(Defeasible) Inference of implicatures from context
Burgundy glasses do not have short stems...
How are brandy glasses different from burgundy glasses?
... and they are similar in other regards
Brandy glasses have short stems.

(a) 3D models of fine-grained types of glasses.

(b) Example interaction between a teacher and a learner discussing generic knowledge about types of glasses.

Figure 1: Learning via embodied dialogue in a simulated tabletop domain.

1 Introduction

Consider a general-purpose robot assistant purchased by a restaurant, which must acquire novel domain knowledge to operate in this particular venue. For example, the agent must learn to distinguish brandy glasses from burgundy glasses (Fig. 1a), but these subcategories of glasses are entirely absent from the agent's domain model in its factory setting. Learning to distinguish among fine-grained visual subcategories is a nontrivial feat (Wei et al., 2021); most current approaches require careful engineering by ML practitioners, making them unsuitable for lay users to readily inspect and update the robot's domain knowledge.

There are also challenges regarding data efficiency, which using natural language can potentially address (Laird et al., 2017). A single *generic statement*—e.g., "Brandy glasses have short stems"—expresses content that would take many visual examples to infer. Such statements, given their dialogue context, may also carry additional meaning that is linguistically implicit. For instance, if the statement "Brandy glasses have short stems" is given as an answer to the contrastive question "How are brandy glasses and burgundy glasses different?", then it implies that burgundy glasses don't have short stems, and also, defeasibly, that these two types of glasses are similar in other conceivable respects (Grice, 1975; Asher, 2013). Vision processing models that exploit natural language data exist (He and Peng, 2017; Xu et al., 2018; Chen et al., 2018; Song et al., 2020), but they generally treat language as supplementary signals for augmenting training examples, rather than leveraging a range of symbolic inferences licensed by purposeful utterances in dialogue.

In this work, we develop an *interactive* symbol grounding framework, in which the teacher presents to the learner evidence for grounding during embodied dialogues like those illustrated in Fig. 1b. The framework is based on a highly modular neurosymbolic architecture, in which subsym-

Proceedings of the 15th International Conference on Computational Semantics pages 318–331
June 21–23, 2023. ©2023 Association for Computational Linguistics

bolic perceptual inputs and symbolic conceptual knowledge obtained during dialogues gracefully combine. We run proof-of-concept experiments to show that agents that exploit semantic and pragmatic inferences from generic statements in discourse outperform baselines that don't exploit semantics and pragmatics, or don't exploit symbolic inference at all.

2 Related Work

In fine-grained image analysis (FGIA), a model learns to distinguish (patches of) images of subcategories that belong to the same basic category. FGIA is challenging because images exhibit small inter-class variance and large intra-class variance, and labeling often requires specific domain expertise, hence high annotation costs (Wei et al., 2021).

A natural approach to FGIA is to utilize information of different modalities, including unstructured text descriptions (He and Peng, 2017; Song et al., 2020), structured knowledge bases (Xu et al., 2018; Chen et al., 2018) and human-edited attention maps (Duan et al., 2012; Mitsuhara et al., 2021). However, to our knowledge, no existing FGIA models exploit NL generic statements provided *in vivo* during natural dialogues. Existing interactive FGIA methods (Branson et al., 2010; Wah et al., 2011, 2014; Cui et al., 2016) query humans to refine predictions from off-the-shelf vision models at inference time but do not update the grounding models. In contrast, our framework supports continuous learning, updating the grounding model as and when the teacher says something noteworthy.

Our use case, described in §1, can be subsumed under the framework of Interactive Task Learning (ITL; Laird et al., 2017). Motivated by scenarios where unforeseen changes may happen to the domain after deployment, the core goal of ITL is to acquire novel concepts that the learner is unaware of but are critical to task success. ITL systems gather evidence from *natural embodied interactions* with a teacher that take place while the learner tries to solve its task. Thus a key desideratum in ITL is that learning should be online and incremental: the learner should change its beliefs and behaviours whenever the teacher provides guidance.

Natural language is a common mode of teacher-learner interaction in ITL (Kirk et al., 2016; She and Chai, 2017). Accordingly, several ITL works draw inspiration from linguistic theories to make learning more effective and efficient. While the formal semantics of quantifiers and negation (Rubavicius and Lascarides, 2022) and of discourse coherence (Appelgren and Lascarides, 2020) has been explored in ITL settings, none of the works in the ITL literature have investigated the utility of exploiting the logical inferences licenced by the semantics and pragmatics of contrastive questions and their generic statement answers.

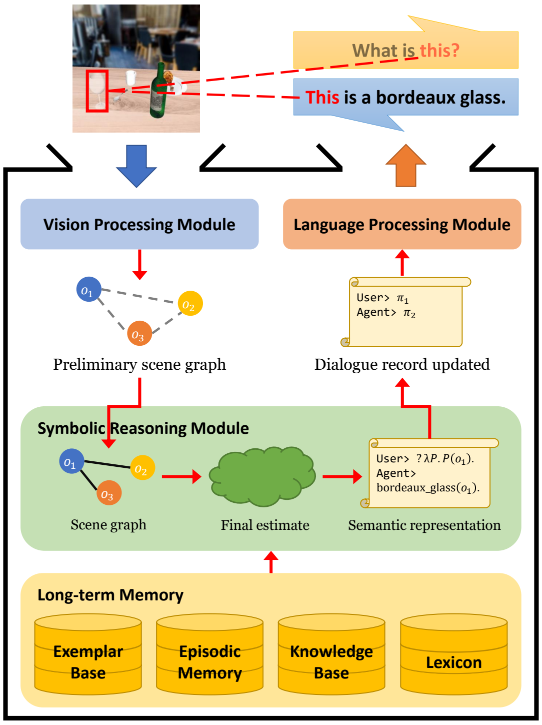


Figure 2: Overview of the architecture in inference mode, in which the component modules interact to generate an answer to a user question.

3 Agent architecture

Fig. 2 illustrates our neurosymbolic architecture for situated ITL agents that can engage in extended dialogues with a teacher. Its design enables both subsymbolic-level learning of visual concepts from perceptual inputs ("This looks like a X") and symbolic-level learning and exploitation of relational knowledge between concepts ("Xs generally have attribute Z") *during* task execution. Here we stress that our proposed approach is not in direct competition with wide-coverage neural vision-language models, but actually complements them. As an ITL framework, we offer a coping mechanism, to be employed when an existing pre-trained model is deployed in a domain where concepts are frequently introduced and changed, requiring the model to quickly adapt with only a few exemplars

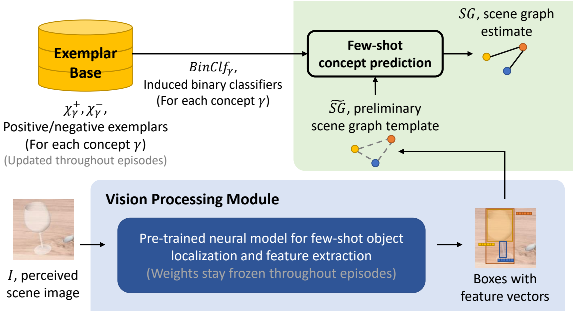

Exemplar Base
$BinClf_\gamma$,
Induced binary classifiers
(For each concept γ)
$\chi_\gamma^+, \chi_\gamma^-$,
Positive/negative exemplars
(For each concept γ)
(Updated throughout episodes)
Few-shot concept prediction
SG, scene graph estimate
$\widetilde{SG}$, preliminary scene graph template
Vision Processing Module
I, perceived scene image
Pre-trained neural model for few-shot object localization and feature extraction
(Weights stay frozen throughout episodes)
Boxes with feature vectors

Figure 3: Abridged illustration of the few-shot scene graph generation process (Full version in Appendix A)

of unforeseen concepts.

3.1 Vision processing module

Given a visual scene perceived by the vision sensor, the agent first summarizes the raw input into a graph-like data structure (*scene graph* hereafter). A scene graph SG encodes a set of salient objects in the scene with their distinguishing features and their pairwise relationships, serving as the agent's internal, abstracted representation of the scene.

Our architecture makes exemplar-based few-shot predictions to generate scene graphs, so as to quickly learn novel visual concepts after a few training instances in an online, incremental fashion. Specifically, our vision processing module employs a neural model extended from Deformable DETR (Zhu et al., 2021), trained to learn distinct low-dimensional metric spaces for each concept type (object class/attribute/relation). The module makes binary concept predictions based on similarity distances between embedded vectors. As illustrated in Fig. 3, the role of the vision module is to process an RGB image input $\mathcal{I}$ into a preliminary scene graph template $\widetilde{SG}$. The template is further processed along with the agent's store of concept exemplars in its long-term memory (§3.3) to yield SG. For further details about the inner working of the neural vision module and the translation process from $\widetilde{SG}$ to SG, refer to Appendix A.

3.2 Language processing module

The language processing module parses natural language utterances into formal semantic representations, maintains dialogue records, and generates natural language utterances as needed. For controlled experiments, we constrain our attention to a class of simple sentences that discuss primarily two types of information: 1) instance-level descriptions about conceptual identities of scene objects (e.g., “This is a brandy glass”, “This has a wide bowl”); and 2) relational knowledge about generic properties shared across instances of the same concepts (e.g., “Brandy glasses have short stems”).

More formally, we represent the propositions expressed by NL sentences via a simple antecedent-consequent pair (PROP hereafter). PROPs draw on a first-order language $\mathcal{L}$ which includes constants referring to objects in the visual scene and predicate symbols for their classes, attributes and pairwise relations (i.e., visual concepts from §3.1). Indicative NL sentences are generally represented with a PROP $\psi = Ante \Rightarrow Cons$, where $Ante$ and $Cons$ are each a $\mathcal{L}$-formula (for ψ, we refer to these as $Ante(\psi)$ and $Cons(\psi)$). $Ante(\psi)$ is empty (and thus omitted) if ψ represents a non-conditional, factual statement. Further, we notate a PROP that stands for a generic characterization with a ‘generic quantifier’ $\mathbb{G}$. For example, the sentences “o is a brandy glass” and “Brandy glasses have short stems” are translated into PROPs respectively as $brandyGlass(o)$ and $\mathbb{G}O.brandyGlass(O) \Rightarrow haveShortStem(O)$.[1]

We represent questions (QUES hereafter) following notation similar to Groenendijk and Stokhof (1982). The answer to a polar question, represented as $?\psi$, is ψ (if true) or $\neg\psi$ (if false). Answers to a *wh*-question $?\lambda X.\psi(X)$ provide values a of X that make $\psi[X/a]$ true (i.e., all occurrences of X in ψ are substituted with a). For the question “How are p_1 and p_2 different?”, we avoid the complexity of higher-order formal languages and simply introduce a reserved formalism $?\texttt{conceptDiff}(p_1, p_2)$, which our implemented dialogue participants can handle by invoking a dedicated proecdure. The answer to $?\texttt{conceptDiff}(p_1, p_2)$ is the set of attributes that all objects of class p_1 have and p_2 lack, and *vice versa*.

The language processing module is implemented as a pipeline with two components: an off-the-shelf large-coverage parser of the English Resource Grammar (Copestake and Flickinger, 2000) followed by manual heuristics that map the parser's outputs to the above forms, as required by the symbolic reasoner (see §3.4). The module also keeps

[1] In the interest of brevity and simplicity, we have translated “have short stems” into an ‘agglomerate’ predicate $haveShortStem$ in this text. This is contrary to the actual implementation, where we introduced the concepts $stem$, $short$ (unary predicates) and $have$ (binary predicate) as elementary units. See Appendix B for a more accurate exposition.

track of the current dialogue history as a sequence of utterances: each one logged as a PROP or QUES, its NL surface form and its speaker.

3.3 Long-term memory module

Our agent stores new knowledge acquired over the course of its operation in its long-term memory. We implement four types of knowledge storage: visual exemplar base (XB), symbolic knowledge base (KB), episodic memory and lexicon.

Visual XB For each visual concept γ, the visual XB stores χ^+_γ and χ^-_γ, a set of positive/negative exemplars worth remembering. The exemplars serve as the basis of the agent's few-shot prediction capability as mentioned in §3.1. The visual XB is expanded each time the agent makes an incorrect prediction. Specifically, when the learner incorrectly states "This is $\tilde{\gamma}$", the teacher provides a corrective response, saying "This is not $\tilde{\gamma}$, this is γ", thereby augmenting χ^+_γ and $\chi^-_{\tilde{\gamma}}$. New sets $\chi^{+/-}_\gamma$ are created whenever the teacher introduces a novel concept γ via a neologism.

Symbolic KB The symbolic KB is a collection of generic PROPs describing relations between symbolic concepts, such as $\mathbb{G}O.brandyGlass(O) \Rightarrow haveShortStem(O)$. Each KB entry is annotated with the source of the knowledge: a generic rule may be explicitly uttered by the teacher or inferred as an implicature, given the dialogue context. We'll discuss how the learner can extract unstated knowledge in §4.2.2 in further detail.

Episodic memory The episodic memory stores the summary of each episode of situated interactions between the agent and the teacher.

Lexicon The lexicon stores a set of content words the teacher introduces into the discourse, along with linguistic metadata like part-of-speech.

3.4 Symbolic reasoning

For symbolic reasoning, we employ a probabilistic variant of a logic programming[2] technique known as answer set programming (ASP; Lifschitz, 2008). The formalism of ASP represents a reasoning problem as a *normal logic program* that consists of rules of the following form:

$$a \leftarrow b_1, \ldots, b_m, \texttt{not}\ c_1, \ldots, \texttt{not}\ c_n. \quad (1)$$

where the rule head atom a and the rule body atoms $\{b_i\}_{i=1}^m, \{c_j\}_{i=1}^n$ can be propositional or (quantifier-free) first-order logic formulas. An intuitive reading of the rule, by itself, is that a is logically justified if and only if all of the positive body atoms $\{b_i\}_{i=1}^m$ hold and none of the negative body atoms $\{c_j\}_{i=1}^n$ are proven to hold. For instance, the ASP rule $fly(X) \leftarrow bird(X), \texttt{not}\ abnormal(X)$ would roughly correspond to the meaning of the generic NL statement "Birds (generally) fly". A rule whose head is empty ($\perp$) represents an *integrity constraint* that its rule body should not hold in answer models.

In probabilistic ASP (Lee and Wang, 2016), each rule is associated with a weight, such that possible worlds satisfying a set of rules with higher total weights are assigned greater probability. Thus a rule may be violated at the expense of its weight. Formally, a probabilistic ASP program $\Pi = \{w : R\}$ is a finite set of weighted rules where R is a rule of the form (1) and w is its associated weight value. The probability of a possible world I according to Π is computed via a log-linear model on the total weight of rules in Π_I, where Π_I is the maximal subset of Π satisfiable by I.

$$W_\Pi(I) = exp\left(\sum_{w:R\in\Pi_I} w\right) \quad (2)$$

$$P_\Pi(I) = \frac{W_\Pi(I)}{\sum_{J\in\text{possible worlds by }\Pi} W_\Pi(J)} \quad (3)$$

For more rigorous technical definition, refer to Lee and Wang (2016).

Each symbol grounding problem is cast into an appropriate program as follows. First, serialize the learner's visual observations contained in the scene graph SG into $\Pi_O = \{\text{logit}(s) : \gamma(o_1, ...).\}$, where each $\gamma(o_1, ...)$ is a visual observation in SG with confidence score $s \in [0, 1]$. Then we export the KB into a program Π_K, built as follows:

- For each KB entry κ, add to Π_K:

  $$\text{logit}(U_d) : \perp \leftarrow Ante(\kappa), \texttt{not}\ Cons(\kappa).$$

  which penalizes 'deductive violation' of κ.

- For each set of KB entries $\{\kappa_i\}$ that share identical $Cons(\kappa_i)$, add to Π_K:

  $$\text{logit}(U_a) : \perp \leftarrow Cons(\kappa_i), \bigwedge_{\kappa_i}\{\texttt{not}\ Ante(\kappa_i)\}$$

[2] In contrast to first-order logic, logic programming is based on the notion of *minimal models*, where any true atom must be justified (founded) by a clause in the logic program.

which penalizes failure to explain $Cons(\kappa_i)$.

Here, $U_d, U_a \in [0, 1]$ are parameters encoding the extent to which the agent relies on its symbolic knowledge; we use $U_d = U_a = 0.95$ in our experiments. For instance, the KB consisting of a single PROP parsed from "Brandy glasses have short stems" will be translated into Π_K consisting of the two rules (7) and (8) in Example 1 below. Finally, the program $\Pi = \Pi_O \cup \Pi_K$ is solved using a belief propagation algorithm (Shenoy, 1997) modified to accommodate the semantics of logic programs.

Example 1. *The program* Π *below encodes a scenario where the agent sees an object* o_1 *and initially estimates* o_1 *is equally likely to be a brandy or burgundy glass. The agent also notices with high confidence it has a short stem, and knows brandy glasses have short stems:*

$$\text{logit}(0.61): \ \ brandyGlass(o_1). \tag{4}$$

$$\text{logit}(0.62): \ \ burgundyGlass(o_1). \tag{5}$$

$$\text{logit}(0.90): \ \ haveShortStem(o_1). \tag{6}$$

$$\text{logit}(0.95): \ \ \bot \leftarrow brandyGlass(O), \quad \mathtt{not}\ haveShortStem(O). \tag{7}$$

$$\text{logit}(0.95): \ \ \bot \leftarrow haveShortStem(O), \quad \mathtt{not}\ brandyGlass(O). \tag{8}$$

This results in $P_\Pi(brandyGlass(o_1)) = 0.91$, *whereas* $P_\Pi(burgundyGlass(o_1)) = 0.62$. *Thus the agent forms a stronger belief that* o_1 *is a brandy glass than it is a burgundy glass.*

See Appendix C for more examples.

4 Interactive Visual Concept Acquisition

4.1 Task description

In our symbol grounding task, each input is a tuple $x_i = (\mathcal{I}_i, b_i)$, where $\mathcal{I}_i \in [0, 1]^{3\times H\times W}$ is an RGB image and b_i is a specification of a bounding box encasing an object in $\mathcal{I}_i$. That is, x_i is essentially reference to an object in an image. The task output y_i is dependent on two possible modes of querying the agent about the identity of the object referenced by x_i. The first 'polar' mode amounts to testing the agent's knowledge of a concept in isolation (i.e., "Is this a X?"; so y_i is yes or no). The second 'multiple-choice' mode demands the agent selects a single object class y_i describing the object among possible candidates (i.e., "What is this?", and y_i is a class). The teacher's response to y_i is dependent on the content of y_i and the teacher's dialogue strategy as described in §4.2.1. The learner updates its symbol grounding model from the teacher's moves using the methods described in §3 and §4.2.2.

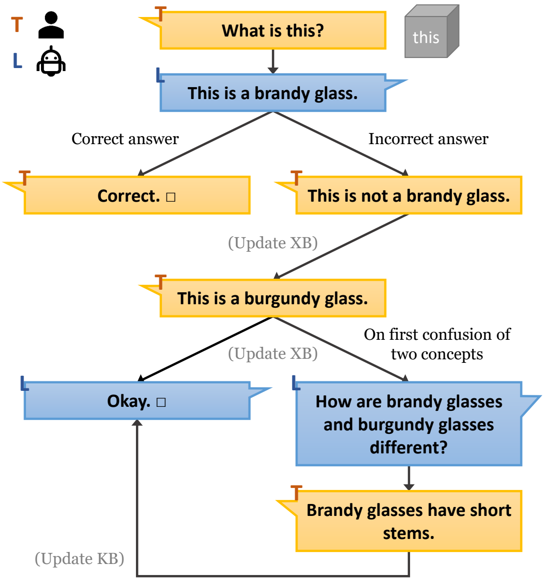

T
L
What is this?
this
This is a brandy glass.
Correct answer
Incorrect answer
Correct. □
This is not a brandy glass.
(Update XB)
This is a burgundy glass.
(Update XB)
On first confusion of two concepts
Okay. □
How are brandy glasses and burgundy glasses different?
Brandy glasses have short stems.
(Update KB)

Figure 4: Flowchart covering the range of training dialogues modeled in this study. □ signals termination of an interaction episode.

As mentioned earlier, the agent's domain model may entirely lack the concept of interest for labelling x_i. The agent acquires unforeseen concepts via teacher utterances. For example, if "This (x_i) is a brandy glass" introduces the agent to the unforeseen concept "brandy glass", then $BrandyGlass$ is added to $\mathcal{L}$ and the visual XB is augmented with newly generated sets $\chi^+_{BrandyGlass} = \{x_i\}$ and $\chi^-_{BrandyGlass} = \varnothing$.

4.2 Flow of dialogues

We focus on a family of dialogues illustrated in Fig. 4. As depicted, each interaction episode is initiated by a teacher query. Dialogues will proceed according to the learner's responses and the teacher's strategy. In this research, we want to investigate how different interaction and learning strategies affect learning efficiency.

4.2.1 Teacher's strategy options

The teacher starts off each interaction episode by presenting an instance o of some visual concept p, querying the learner with a probing QUES "$?\lambda P.P(o)$".[3] If the learner provides the correct

[3] Note that the expression $?\lambda P.P(o)$ does not fully capture the intended meaning of "What is this?" in its own right, since the discourse contexts set up additional semantic/pragmatic

answer as the PROP "$p(o)$", the teacher responds "correct" and the episode terminates without agent belief updates. Otherwise, if the learner provides an incorrect answer, "$\tilde{p}(o)$" or "I am not sure", the teacher needs to provide some corrective information so that the learner can adjust its beliefs. We implement and compare the following variations in the teacher's response, in increasing order of information content:

- minHelp: Provides only boolean feedback to the learner's answer, i.e., "$\neg\tilde{p}(o)$".
- medHelp: In addition to minHelp, provides the correct answer label, saying "$p(o)$".
- maxHelp: In addition to medHelp, provides a set of generic PROPs that characterize p or $\tilde{p}$. The feedback is provided after the learner's QUES "$\texttt{?conceptDiff}(p, \tilde{p})$", asked once on the first confusion between p and $\tilde{p}$.

The generic PROPs provided by maxHelp teachers originate from the teacher's domain knowledge, which we assume here to be correct and exhaustive. The set of PROPs to be delivered is computed as the symmetric difference between the set of properties of p versus that of $\tilde{p}$ (see Appendix D for an example). minHelp and medHelp serve as vision-only baselines since only concept exemplars with binary labels are communicated as teaching signal.

4.2.2 Learner's strategy options

Another dimension of variation we model is the learner's strategy for interpreting generic statements within dialogue contexts. Note that the variation in this dimension is meaningful only when the teacher deploys the maxHelp strategy, thereby allowing exploitation of generic statements.

In human dialogues, interlocutors infer, and speakers exploit, *implicatures* that are validated by linguistically explicit moves, given the context of utterance (Grice, 1975). As a core contribution of this study, we model how generic statements given as an answer to a question about similarities and differences give rise to certain implicatures (Asher, 2013) that can be exploited for more data-efficient learning.

Suppose a question "How are X and Y different?" is answered with a generic statement "Xs have attribute Z". The following implicatures can arise from this discourse context: 1) "Ys do not have attribute Z", and 2) "X and Y are otherwise similar". The former follows from the assumption that the generic is a coherent answer to a contrastive question (Asher and Lascarides, 2003). The latter, which arguably is more defeasible (Grice, 1975), is what's known as a scalar implicature: if there were other important differences that the learner should know, then Gricean maxims of conversation predict that the teacher would have included them in the answer as well.

Situation	
Confusion	`brandy glass` vs. `burgundy glass`
Teacher input	"Brandy glasses have short stems."
Current KB	$\mathbb{G}O.\mathit{brandyGlass}(O) \Rightarrow \mathit{haveWideBowl}(O)$

Strategy	New KB entries added
semOnly	$\mathbb{G}O.\mathit{brandyGlass}(O) \Rightarrow \mathit{haveShortStem}(O)$
semNeg	$\mathbb{G}O.\mathit{brandyGlass}(O) \Rightarrow \mathit{haveShortStem}(O)$ $\mathbb{G}O.\mathit{burgundyGlass}(O) \Rightarrow \neg\mathit{haveShortStem}(O)$
semNegScal	$\mathbb{G}O.\mathit{brandyGlass}(O) \Rightarrow \mathit{haveShortStem}(O)$ $\mathbb{G}O.\mathit{burgundyGlass}(O) \Rightarrow \neg\mathit{haveShortStem}(O)$ $\mathbb{G}O.\mathit{burgundyGlass}(O) \Rightarrow \mathit{haveWideBowl}(O)$

Table 1: An example of how different learner strategies update their KBs from the teacher's generic statement feedback after the learner has confused a burgundy glass for a brandy glass. The learner has already learned that burgundy glasses have wide bowls. PROPs in black is obtained from the teacher's NL utterance; PROPs in red from 'negative' implicatures (ψ^{neg} from ψ) as demanded by coherence; and PROPs in blue from scalar implicatures (κ^{scl} from κ).

For a PROP ψ, let $\psi^{p \leftrightarrow q}$ denote a PROP which is identical to ψ except that occurrences of the predicate p are all substituted with the predicate q and *vice versa*. We consider the following strategies the learner can take when interpreting a set of generic PROPs $\{\psi_i\}$ provided during an episode:

- semOnly: Simply add all ψ_i's to KB.
- semNeg: In addition to semOnly, infer a generic PROP $\psi_i^{neg} = \mathit{Ante}(\psi_i^{p \leftrightarrow \tilde{p}}) \Rightarrow \neg\mathit{Cons}(\psi_i^{p \leftrightarrow \tilde{p}})$ for each ψ_i given as answer to $\texttt{?conceptDiff}(p, \tilde{p})$.
- semNegScal: In addition to semNeg, infer a generic PROP $\kappa^{scl} = \mathit{Ante}(\kappa^{p \leftrightarrow \tilde{p}}) \Rightarrow \mathit{Cons}(\kappa^{p \leftrightarrow \tilde{p}})$ for each KB entry κ that has either p or $\tilde{p}$ mentioned, only if κ^{scl} is not inconsistent with any of ψ_i's or ψ_i^{neg}'s.

For example, consider the example situation illustrated in Tab. 1. The semNeg learner adds "Bur-

constraints on what counts as acceptable answers. We have approximated those constraints via our pre-defined dialogue strategies.

gundy glasses do not have short stems" to its KB, and semNegScal in addition adds "Brandy glasses have X" for every property X that, according to its KB, burgundy glasses have (e.g., wide bowls).

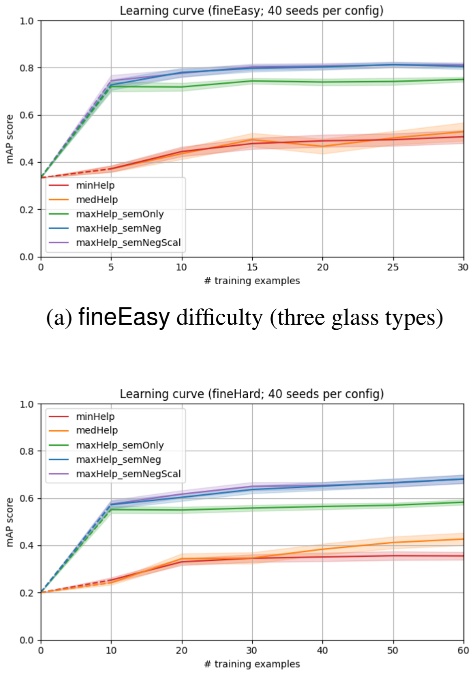
While the semNeg inference stems from the demand that the teacher's move is a coherent answer (Asher and Lascarides, 2003), the scalar implicatures inferred by semNegScal are defeasible presumptions (Grice, 1975). That is, semNegScal risks misunderstanding the teacher's intended meaning, inferring general rules that are incorrect—yet cancellable (see Appendix E for an example failure case). Subsequent pieces of refuting evidence may falsify the inferred implicatures without rendering the conversation incoherent. Therefore, we equip our agents with some risk management faculty that can assess and reject contents of scalar implicatures. This is achieved by periodically testing KB entries whose origin is solely from scalar implicatures, rejecting those whose counterexamples can be found in the episodic memory.

5 Experiments

5.1 Evaluation Scheme

We run a suite of experiments that evaluate the data efficiency of the learner's and teacher's strategies from §4.2. Results are averaged over multiple sequences of interaction episodes for each of five combinations of teacher's and learner's strategies: minHelp, medHelp, maxHelp+semOnly, maxHelp+semNeg and maxHelp+semNegScal. Each episode-initial probing question "$?\lambda P.P(o)$" is associated with a randomly selected instance o of a concept selected from a round-robin of the target concepts to be acquired. For controlled random selections of concept instances and shuffling of the round-robin, 40 seeds are shared across different configurations. Each sequence continues until the learner makes N_t mistakes in total.

As is common in ITL scenarios, training and inference are fully integrated. Learning has to take place during use whenever the teacher imparts information. In this work, we evaluate our learners by having them take 'mid-term exams' on a separate test set after every N_m mistakes made ($N_m \leq N_t$). The mid-term exams comprise binary prediction problems "$?p(o)$" asked per every target concept p for each test example o, and we collect confidence scores between 0 and 1 as response. The primary evaluation metric reported is mean average precision (mAP)[4]; we do not use an F1 score because we are more interested in relative rankings between similar-looking concepts than the learners' absolute performances at some fixed confidence threshold. We also report averaged confusion matrices collected for the sequence-final exams (partially in Fig. 6, fully in the supplementary material).



(a) fineEasy difficulty (three glass types)

(b) fineHard difficulty (five glass types)

Figure 5: Averaged learning curves (with 95% confidence intervals): effective training examples vs. mAP.

5.2 Setup

The learner agents start with relatively good, but still error-prone, priors of what bowls and stems and their attributes (e.g., "short stem") look like, but completely lacks the vocabulary, concepts and related visual features for the various glass types. The prior knowledge is injected into the learner agents by exposing them to the full set of positive examples of stems and bowls in our data set, and randomly sampled non-instances for negative examples. The average binary classification accuracies on balanced test sets were 98.11% for the part concepts and 86.12% for their attributes.

Our training and testing images are randomly generated from a simulation framework CoppeliaSim (Rohmer et al., 2013), using a toolkit for controlled sampling of 3D environments (Innes and Ramamoorthy, 2021). Each image features a

[4] Mean of areas under interpolated precision-recall curves.

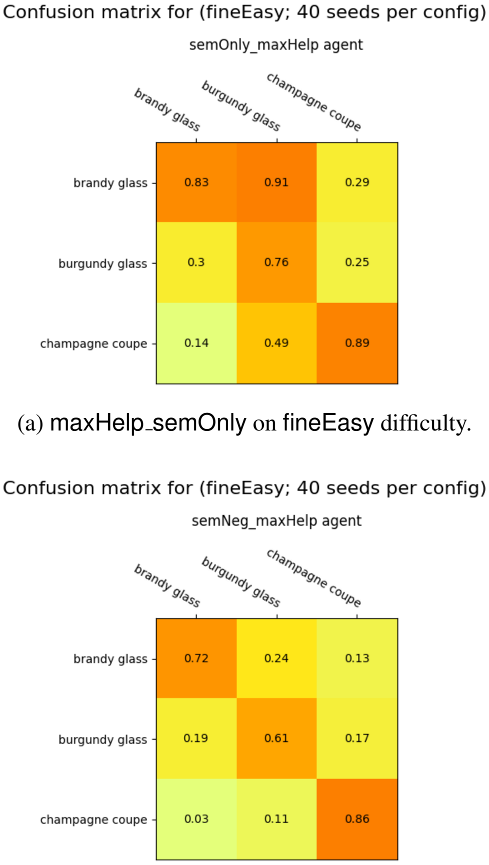
scene of several objects from the restaurant domain laid on a tabletop (e.g., the image in Fig. 2). Each type of glass in our tabletop domain can be characterized by its parts having different attributes; see Appendix D for the complete list. We implement simulated teachers in place of real human users for the experiments, which perform rule-based pattern matching just sufficient for participating as a teacher in our training dialogues. Our implementation and datasets are publicly released in https://github.com/itl-ed/ns-arch.

The more distractor concepts we have, the more difficult the task becomes; difficulty scales roughly quadratically with respect to the number of concepts, since C concepts enable $\binom{C}{2}$ different pairwise confusions. Our experiments cover two levels of difficulty: fineEasy and fineHard. For fineEasy, we set $(C, N_t, N_m) = (3, 30, 5)$, where target concepts are {brandy glass, burgundy glass, champagne coupe}. For fineHard, we set $(C, N_t, N_m) = (5, 60, 10)$, where target concepts as those for fineEasy plus {bordeaux glass, martini glass}.

5.3 Results and Discussion

Fig. 5 and Tab. 2 display the averaged learning curves for the five strategy combinations in each task difficulty setting, along with 95% confidence intervals. It is obvious that learners exploiting the semantics of generic statements from maxHelp teachers are significantly faster in picking up new concepts, compared to the vision-only baseline configurations with minHelp or medHelp teachers. Among the maxHelp results, the learners which extract and exploit additional, unstated information from the context (i.e., semNeg and semNegScal) outperform the learner semOnly, which doesn't exploit pragmatics.

Our error analysis reveals that the significant performance boosts enjoyed by semNeg and semNegScal learners comes from the ability to infer non-properties from property statements (i.e. ψ^{neg} from ψ). The confusion matrices reported in Fig. 6 allow us to study the mechanism. Specifically, notice how the maxHelp_semOnly learner in Fig. 6a frequently misclassifies brandy glasses as burgundy glasses, whereas it is considerably less likely to make such mistakes in the opposite direction: 91% vs. 30%. We can see this is because semOnly learners do not have access to the negative property of burgundy glasses of not having short stems ($\mathbb{G}O.burgundyGlass(O) \Rightarrow \neg haveShortStem(O)$). Therefore, while semOnly learners can confidently dismiss instances of burgundy glasses as non-instances of brandy glasses, they are not able to dismiss instances of brandy glasses as non-instances of burgundy glasses. We can observe in Fig. 6b that this is precisely remedied by semNeg and semNegScal learners, which are able to reliably distinguish the two types in both directions: 24% vs. 19%.



(a) maxHelp_semOnly on fineEasy difficulty.

(b) maxHelp_semNeg on fineEasy difficulty.

Figure 6: Averaged confusion matrices taken from the sequence-final evaluations for two configurations.

The difference between semNeg and semNegScal learner is more subtle. Although their performances generally tend to converge after sufficient training, learners that exploit scalar implicatures seem to show higher data efficiency at earlier stages, especially in the fineHard task. Nonetheless, the two learning curves have largely overlapping confidence intervals; we cannot make a strong scientific claim based on these results, and we will have to conduct experiments at a larger scale to corroborate this difference.

Task difficulty	fineEasy			fineHard		
# training examples	5	15	30	10	30	60
minHelp	0.372	0.478	0.507	0.253	0.345	0.355
medHelp	0.371	0.494	0.529	0.241	0.346	0.426
maxHelp_semOnly	0.719	0.743	0.750	0.551	0.558	0.582
maxHelp_semNeg	0.727	0.797	0.805	0.572	0.636	**0.681**
maxHelp_semNegScal	**0.744**	**0.803**	**0.811**	**0.574**	**0.649**	**0.681**

Table 2: Task performances of agents by mAP scores after different numbers of effective training examples.

6 Conclusion and Future Directions

In this research, we have proposed an interactive symbol grounding framework for ITL, along with a neurosymbolic architecture for the learner agent. We empirically showed that learners who can comprehend and exploit valid inferences from generic statements, including pragmatic content given their context of use, can learn to ground novel visual concepts more data-efficiently. Our findings confirm it pays to study human-AI natural language interactions through the lens of discourse semantics, not only the truth conditions of isolated sentences but also their coherent connections to their context.

In future, we plan to relax some of many simplifying assumptions we made for controlled experiments, possibly exploring other domains. For instance, the ideal assumption that teachers are infallible and communication is noise-free does not hold in most real-world scenarios (Appelgren and Lascarides, 2021). Further, the set of linguistic constructions we have studied in this work is very constrained (as intended), and a natural next step is to accommodate a wider range of diverse and free-form NL constructions. It is also a strong assumption that the learner agent already has relatively reliable beliefs about object part and concept attributes. For example, if the learner does not know what the “stem” of a wine glass means, the absence of the concept must be resolved before communicating any generic characterizations involving stems. Finally, our approach does not fully exploit the semantics of generic statements, which express qualitative rules that admit exceptions (Pelletier and Asher, 1997). The generic quantifier $\mathbb{G}$ did not play any significant role in this work. One major strength of ASP is that it is well suited for modeling non-monotonic inferences, and it would be interesting to study how to model ITL scenarios that can robustly address exceptions to generic rules.

Acknowledgements

This work was supported by the UKRI Research Node on Trustworthy Autonomous Systems Governance and Regulation (grant EP/V026607/1) and by Informatics Global PhD Scholarships, funded by the School of Informatics at The University of Edinburgh. We thank the anonymous reviewers for their feedback on an earlier draft of this paper, and Mattias Appelgren, Rimvydas Rubavicius and Gautier Dagan for continued feedback over the course of this research.

References

Mattias Appelgren and Alex Lascarides. 2020. Interactive task learning via embodied corrective feedback. *Autonomous Agents and Multi-Agent Systems*, 34(2):1–45.

Mattias Appelgren and Alex Lascarides. 2021. Symbol grounding and task learning from imperfect corrections. In *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*, pages 1–10.

Nicholas Asher. 2013. Implicatures and discourse structure. *Lingua*, 132:13–28.

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.

Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. 2010. Visual recognition with humans in the loop. In *European Conference on Computer Vision*, pages 438–451. Springer.

Tianshui Chen, Liang Lin, Riquan Chen, Yang Wu, and Xiaonan Luo. 2018. Knowledge-embedded representation learning for fine-grained image recognition. *arXiv preprint arXiv:1807.00505*.

Ann A Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage english grammar using hpsg. In *LREC*, pages 591–600. Athens.

Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. 2016. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1153–1162.

Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. 2012. Discovering localized attributes for fine-grained recognition. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3474–3481. IEEE.

Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. 2004. Neighbourhood components analysis. *Advances in neural information processing systems*, 17.

H. P. Grice. 1975. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics Volume 3: Speech Acts*, pages 41–58. Academic Press.

J. Groenendijk and M. Stokhof. 1982. Semantic analysis of wh-complements. *Linguistics and Philosophy*, 5(2):175–233.

Xiangteng He and Yuxin Peng. 2017. Fine-grained image classification via combining vision and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5994–6002.

Craig Innes and Subramanian Ramamoorthy. 2021. Probrobscene: A probabilistic specification language for 3d robotic manipulation environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9446–9452. IEEE.

James Kirk, Aaron Mininger, and John Laird. 2016. Learning task goals interactively with visual demonstrations. *Biologically Inspired Cognitive Architectures*, 18:1–8.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

J. Laird, K. Gluck, J. Anderson, K. Forbus, O. Jenkins, C. Lebiere, D. Salvucci, M. Scheutz, A. Thomaz, J. Trafton, Robert. Wray, S. Mohan, and J. Kirk. 2017. Interactive task learning. *IEEE Intelligent Systems*, 32:6–21.

Joohyung Lee and Yi Wang. 2016. Weighted rules under the stable model semantics. In *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Vladimir Lifschitz. 2008. What is answer set programming? AAAI'08, page 1594–1597. AAAI Press.

Masahiro Mitsuhara, Hiroshi Fukui, Yusuke Sakashita, Takanori Ogata, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. 2021. Embedding human knowledge in deep neural network via attention map. In *VISIGRAPP*.

Francis Jeffry Pelletier and Nicholas Asher. 1997. Generics and defaults. In *Handbook of logic and language*, pages 1125–1177. Elsevier.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Eric Rohmer, Surya PN Singh, and Marc Freese. 2013. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326. IEEE.

Rimvydas Rubavicius and Alex Lascarides. 2022. Interactive symbol grounding with complex referential expressions. In *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Lanbo She and Joyce Chai. 2017. Interactive learning of grounded verb semantics towards human-robot communication. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1634–1644.

Prakash P Shenoy. 1997. Binary join trees for computing marginals in the shenoy-shafer architecture. *International Journal of approximate reasoning*, 17(2-3):239–263.

Kaitao Song, Xiu-Shen Wei, Xiangbo Shu, Ren-Jie Song, and Jianfeng Lu. 2020. Bi-modal progressive mask attention for fine-grained recognition. *IEEE Transactions on Image Processing*, 29:7006–7018.

Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie. 2011. Multiclass recognition and part localization with humans in the loop. In *2011 International Conference on Computer Vision*, pages 2524–2531. IEEE.

Catherine Wah, Grant Van Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie. 2014. Similarity comparisons for interactive fine-grained categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866.

Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. 2021. Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Huapeng Xu, Guilin Qi, Jingjing Li, Meng Wang, Kang Xu, and Huan Gao. 2018. Fine-grained image classification by visual-semantic embedding. In *IJCAI*, pages 1043–1049.

Nick Zangwill. 2011. Negative properties. *Noûs*, 45(3):528–556.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*.

A Vision processing module: implementational details

Our implementation of the few-shot neural vision processing module is based on the pretrained model of two-staged Deformable DETR (Zhu et al., 2021). We train new lightweight multilayer perceptron (MLP) blocks for embedding image regions into low-dimensional feature spaces. The MLP blocks replace the existing pretrained prediction heads that have fixed number of output categories, enabling metric-based few-shot predictions of incrementally learned visual concepts.

Let C, A and R denote open sets of visual concepts of different types: object classes[5], attributes[6] and pairwise relations[7]. In principle, we need one metric space for each concept type for their separate handling, hence three MLP blocks to train. But for this work, $|R| = 1$, where the only relation concept we need to capture is 'have' (whole-part relationship). We can make proxy predictions for the concept by the ratio of the area of bounding box intersection to the area of the candidate object part's bounding box. Therefore, in the interest of simplicity, we prepare only two embedder blocks for C and A respectively; in future extensions where we need to deal with a truly open R, we will have to implement a relation-centric embedder block for R as well.

Fig. 7 depicts how our vision module summarizes the raw RGB image input $\mathcal{I} \in [0,1]^{3\times H\times W}$ into a preliminary scene graph $\widetilde{SG}$, and then makes few-shot predictions to finally yield a scene graph SG. $\mathcal{I}$ is first passed through the feature extractor backbone to produce $\{f_l\}_{l=1}^{L}$, a set of feature maps $f^l \in \mathbb{R}^{C\times H^l\times W^l}$ at L different scales. $\{f_l\}_{l=1}^{L}$ are flattened into a single sequence of input tokens (thus in $\mathbb{R}^{C\times\sum_l H^l\cdot W^l}$), combined with appropriate positional encodings and fed into the encoder. We obtain from the encoder an objectness logit score s_i and a proposal bounding box coordinate $\mathbf{b}_i \in [0,1]^4$ for the input tokens, out of which the top k proposals with the highest s_i scores are selected. The selected proposals are fed into the decoder along with corresponding feature vectors to generate $\mathbf{f}_i^c, \mathbf{f}_i^a \in \mathbb{R}^D$, the class/attribute-centric embeddings of each input token, in addition to the (refined) bounding box coordinates $\mathbf{b}_i$. The decoder outputs are collated into the preliminary scene graph template $\widetilde{SG}= (\tilde{N}, \tilde{E})$. $\tilde{N}$ is the node set containing $\mathbf{b}_i, \mathbf{f}_i^c, \mathbf{f}_i^a$ for each detected object. $\tilde{E}$ is the edge set that *would* contain pairwise relation-centric embeddings $\mathbf{f}_{i,j}^r$ for each pair of detected objects. However, $\tilde{E}$ is essentially empty in our current implementation since as mentioned above, we fall back to proxy prediction by area ratio for the only relation concept of interest 'have'.

For each visual concept $\gamma \in C, A(, R)$, the agent's visual XB stores $\chi_\gamma^{+/-}$, a set of positive/negative exemplars, which together naturally induce a binary classifier $BinClf_\gamma$. We are free to choose any binary classification algorithm as long as it can return probability scores for concept membership from χ_γ^+ and χ_γ^-. We use Platt-scaled SVM with RBF kernel (Platt et al., 1999) in our implementation. Then, $SG = (N, E)$ is generated from $\widetilde{SG}$ and a set of $BinClf_\gamma$'s, where N and E are each the scene graph node set and the scene graph edge set. For each scene object, N contains $\mathbf{c}_i \in [0,1]^{|C|}$ and $\mathbf{a}_i \in [0,1]^{|A|}$, each a vector designating the probabilistic beliefs of whether the object classifies as an instance of concepts in C and A, as well as the box specification $\mathbf{b}_i$. E contains information about binary relationships between ordered pairs of objects, namely $\mathbf{r}_{i,j} \in [0,1]^{|R|}$, the probabilistic beliefs of whether the pair (i,j) is an instance of concepts in R. As a reminder, in our setting, N is computed from $\tilde{N}$ and $BinClf_\gamma$ for each $\gamma \in C, A$, whereas E is computed from bounding box area ratios.

Our new embedder blocks are trained on 50% of the Visual Genome dataset (Krishna et al., 2017) with NCA loss objective (Goldberger et al., 2004) for metric learning, for 80,000 steps using SGD optimizer with the batch size of 64, the learning rate of 3×10^{-4} and the momentum factor of 0.1. The prediction heads are then fine-tuned on our tabletop domain dataset[8] for 2,000 steps using Adam optimizer, with the batch size of 16, the initial learning

[5] Intuitively corresponding to concepts denoted by nouns—e.g.,'brandy glass', 'stem'.

[6] intuitively corresponding to concepts denoted by adjectives—e.g., 'wide', 'short'.

[7] intuitively corresponding to concepts denoted by transitive verbs and adpositions—e.g., 'have', 'of'

[8] Excluding the fine-grained types of drinking glasses.

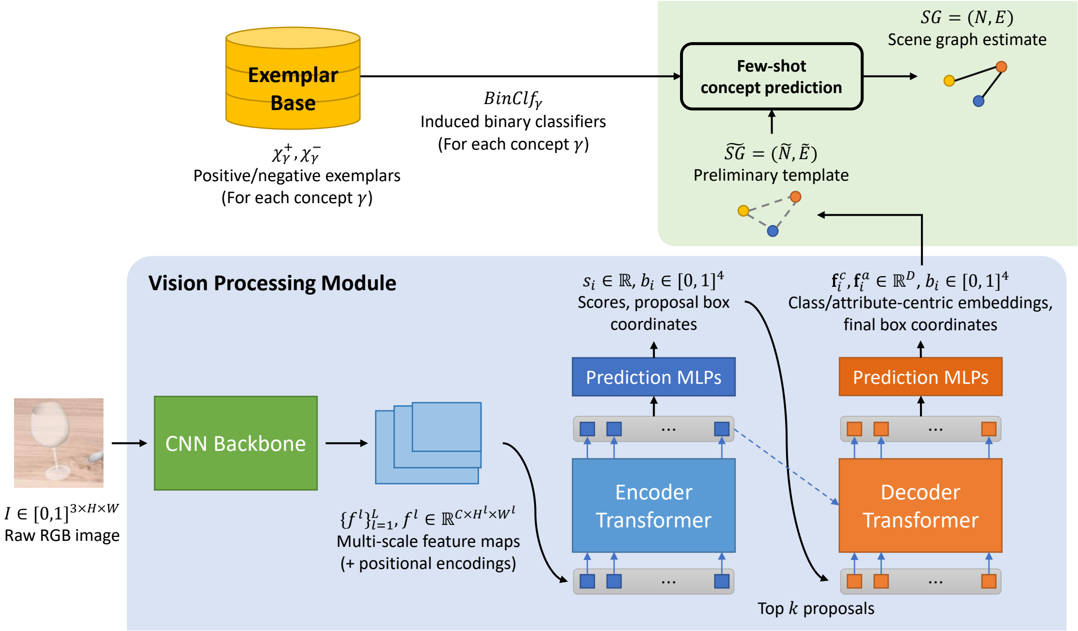

Exemplar Base
$\chi_\gamma^+, \chi_\gamma^-$
Positive/negative exemplars
(For each concept γ)
$BinClf_\gamma$
Induced binary classifiers
(For each concept γ)
Few-shot concept prediction
$SG = (N, E)$
Scene graph estimate
$\widetilde{SG} = (\tilde{N}, \tilde{E})$
Preliminary template
Vision Processing Module
$I \in [0,1]^{3\times H\times W}$
Raw RGB image
CNN Backbone
$\{f^l\}_{l=1}^L, f^l \in \mathbb{R}^{C\times H^l\times W^l}$
Multi-scale feature maps
(+ positional encodings)
Encoder Transformer
Prediction MLPs
$s_i \in \mathbb{R}, b_i \in [0,1]^4$
Scores, proposal box coordinates
Top k proposals
Decoder Transformer
Prediction MLPs
$\mathbf{f}_i^c, \mathbf{f}_i^a \in \mathbb{R}^D, b_i \in [0,1]^4$
Class/attribute-centric embeddings, final box coordinates

Figure 7: A schematic of the structure of the vision processing module component in our agent architecture, and the pipeline through which raw visual inputs are processed into the final scene graph estimate.

rate of 2×10^{-4} and PyTorch default values[9] for the hyperparameters $\beta_1, \beta_2, \epsilon$.

B FOL representation of concept properties

In the main paper, we have represented the NL predication "have short stems" with an agglomerate predicate *haveShortStem* for the sake of brevity, so that "Brandy glasses have short stems" would be translated into the PROP $\mathbb{G}O.brandyGlass(O) \Rightarrow haveShortStem(O)$. However, this is an oversimplification of what is actually happening under the hood in our implementation. The predication "have short stems" ought to be broken down into its constituent meanings delivered by the individual tokens "have", "short" and "stem" respectively, for primarily two reasons: 1) they are the elementary units of concepts handled by the vision module and included in the output scene graphs, and 2) the object parts should be explicitly acknowledged as entities separate from the objects they belong to, and the generic PROPs should model relations between objects and their parts (plus their attributes).

In light of this, we choose to read NL sentences of the form "`{object}`s have `{attribute}` `{part}`s" as follows: "If O is an `object`, there exists an entity P such that O has P as its part, and P is a `part` that is `attribute`". Then, for example, the sentence "Brandy glasses have short stems" would be represetned by the following PROP:

$$\mathbb{G}O.brandyGlass(O) \Rightarrow \\ (\exists P.have(O,P), short(P), stem(P))$$

or alternatively,

$$\mathbb{G}O.brandyGlass(O) \Rightarrow \\ have(O, f(O)), short(f(O)), stem(f(O))$$

where f is a skolem function that maps from the instance of brandy glass to its (only) short stem. We opt for the latter option because it is more compliant with the formalism commonly used by logic programming methods, in which existential quantifiers are not admitted and variables are all implicitly universally quantified.

C More examples of grounding problems as probabilistic ASP programs

Example 2 below illustrates how lack of high-confidence observation of a short stem of a glass o_1 results in a weaker belief that o_1 is a brandy glass.

Example 2. *The agent sees an object o_1 and initially estimates it's equally likely to be a brandy or burgundy glass. The agent also notices with high confidence it does NOT have a short stem, and*

[9] https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

knows brandy glasses have short stems:

$$\text{logit}(0.61) : \ \mathit{brandyGlass}(o_1). \tag{9}$$

$$\text{logit}(0.62) : \ \mathit{burgundyGlass}(o_1). \tag{10}$$

$$\text{logit}(0.10) : \ \mathit{haveShortStem}(o_1). \tag{11}$$

$$\text{logit}(0.95) : \ \bot \leftarrow \mathit{brandyGlass}(O), \ \texttt{not}\ \mathit{haveShortStem(O)}. \tag{12}$$

$$\text{logit}(0.95) : \ \bot \leftarrow \mathit{haveShortStem}(O), \ \texttt{not}\ \mathit{brandyGlass(O)}. \tag{13}$$

This results in $P_\Pi(\mathit{brandyGlass}(o_1)) = 0.20$, *whereas* $P_\Pi(\mathit{burgundyGlass}(o_1)) = 0.62$.

Example 3 shows how knowledge of *negative* properties of an object class can affect symbolic reasoning. The example supposes the agent's KB only consists of the knowledge "Burgundy glasses do *not* have short stems", namely the PROP $\mathbb{G}O.\mathit{burgundyGlass}(O) \Rightarrow \neg \mathit{haveShortStem}(O)$. Note how we translate a generic PROP whose Cons is a negation of some $\mathcal{L}$-formula into probabilistic ASP rules. Only rules penalizing deductive violations are generated, in which the negation ($\neg$) that wraps around Cons 'cancels out' the default negation `not`. We do not generate rules for penalizing failures to explain Cons from negative PROPs, as abductive inferences of object classes from lack of properties would give rise to far-fetched conclusions: e.g., inferring something might be a banana because it does not have wheels.

Example 3. *The agent sees an object* o_1 *and initially estimates it's equally likely to be a brandy or burgundy glass. The agent also notices with high confidence it has a short stem, and knows burgundy glasses do NOT have short stems:*

$$\text{logit}(0.61) : \ \mathit{brandyGlass}(o_1). \tag{14}$$

$$\text{logit}(0.62) : \ \mathit{burgundyGlass}(o_1). \tag{15}$$

$$\text{logit}(0.90) : \ \mathit{haveShortStem}(o_1). \tag{16}$$

$$\text{logit}(0.95) : \ \bot \leftarrow \mathit{burgundyGlass}(O), \ \mathit{haveShortStem}(O). \tag{17}$$

This results in $P_\Pi(\mathit{brandyGlass}(o_1)) = 0.61$, *whereas* $P_\Pi(\mathit{burgundyGlass}(o_1)) = 0.19$.

Note that knowledge about brandy glasses do not affect the likelihood of o_1 being a burgundy glass, and *vice versa*: i.e., for an object, the events of being a brandy glass vs. a burgundy glass are independent. This is because the KBs in the examples do not introduce any type of dependency between the two glass types. For instance, if we inject mutual exclusivity relation between the two types in the KB, both probability values $P_\Pi(\mathit{brandyGlass}(o_1))$ and $P_\Pi(\mathit{burgundyGlass}(o_1))$ would be affected by knowledge about either.

D Task domain: Fine-grained types of drinking glasses to distinguish

Type	Properties	Sample image
`bordeaux glass`	Bowl: elliptical, tapered.	
`brandy glass`	Bowl: wide, tapered, round. Stem: short.	
`burgundy glass`	Bowl: wide, tapered, round.	
`champagne coupe`	Bowl: broad, round.	
`martini glass`	Bowl: broad, conic.	

Table 3: Fine-grained types of drinking glasses modeled in our tabletop domain. (Note only brandy glasses have characteristic stems, whereas bowls of all glass types can be characterized by some set of attributes.)

Tab. 3 lists the set of fine-grained types of drinking glasses that are modeled in our simulated tabletop domain, along with their properties and sample images. 3D meshes of the glasses are obtained from a website that lists stock models made by third-party providers,[10] then imported into the simulation environment.

As illustrated, properties of each fine-grained type comprise its part attributes. For instance, the full set of properties of a brandy glass could be expressed as a set {(wide, bowl), (tapered, bowl), (round, bowl), (short, stem)}. When asked, our simulated teacher computes the answer to `?conceptDiff` QUES as pairwise symmetric differences between property sets: e.g., for `?conceptDiff`(*brandyGlass*,*burgundyGlass*),

[10] https://www.turbosquid.com/3d-models/wine-glasses-3d-1385831

Confusion	champagne coupe - burgundy glass	burgundy glass - bordeaux glass
KB state	$\mathbb{G}O.champagneCoupe(O) \Rightarrow haveBroadBowl(O)$ $\mathbb{G}O.burgundyGlass(O) \Rightarrow haveWideBowl(O), haveTaperedBowl(O)$ $\mathbb{G}O.burgundyGlass(O) \Rightarrow \neg haveBroadBowl(O)$ $\mathbb{G}O.champagneCoupe(O) \Rightarrow \neg(haveWideBowl(O), haveTaperedBowl(O))$	$\mathbb{G}O.burgundyGlass(O) \Rightarrow haveWideBowl(O), haveRoundBowl(O)$ $\mathbb{G}O.bordeauxGlass(O) \Rightarrow haveEllipticalBowl(O)$ $\mathbb{G}O.bordeauxGlass(O) \Rightarrow \neg(haveWideBowl(O), haveRoundBowl(O))$ $\mathbb{G}O.burgundyGlass(O) \Rightarrow \neg haveEllipticalBowl(O)$ $\underline{\mathbb{G}O.bordeauxGlass(O) \Rightarrow haveWideBowl(O), haveTaperedBowl(O)}$ $\mathbb{G}O.bordeauxGlass(O) \Rightarrow \neg haveBroadBowl(O)$

Table 4: An example illustration of how semNegScal learners can infer incorrect and unintended knowledge. The underlined PROP denotes a generic rule which is neither correct nor intended by the teacher.

we obtain $\{$(short, stem)$\}$ for brandy glasses and $\varnothing$ for burgundy glasses.

These properties of glasses did not ship with the 3D models; instead, we hand-coded them based on information available on the internet. We have put effort to prepare an annotation scheme that is faithful to properties of the glasses in the reality, yet the domain knowledge may still have inconsistency against the 'ground-truth'—any error in that regard remains our own.

E Rule acquisition by inference of implicatures and failure case analysis

In this work, we assume that all generic NL statements given by the teacher are characterizations of object classes by their positive properties (those described in Appendix D), and statements of *negative* properties are never explicitly provided. This reflects the fact that we usually characterize things by their positive properties rather than by their negative properties because the former generally have more determining power (Zangwill, 2011). Therefore, in our experiments, negative properties can be obtained only by virtue of inference of implicatures. That is, for example, only semNeg or semNegScal learners have access to the negative PROP $\mathbb{G}O.burgundyGlass(O) \Rightarrow \neg haveShortStem(O)$.

Nevertheless, semNegScal learners risk acquisition of incorrect and unintended knowledge when they make inferences of scalar implicatures. To see this, study the example illustrated in Tab. 4, where two successive confusions take place in the order of brandy glass vs. burgundy glass and then burgundy glass vs. champagne coupe. In the example, the underlined PROP successfully infiltrates into the learner's KB without being suppressed by explicitly stated PROPs or their negative implicature counterparts. This is why inference of the scalar implicatures should be cancellable, so that they can be retracted in the face of contradictory evidence. In our implementation, this is achieved by periodically inspecting the KB entries against the episodic memory, removing any rules whose counterexamples are found.

Author Index