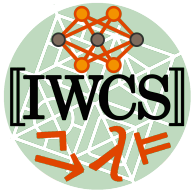
IWCS 2021



The 14th International Conference on Computational Semantics

Proceedings of the Conference

June 17 - 18, 2021
Groningen, The Netherlands (Online)

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 978-1-954085-19-0

Message from the Organizers

These are the conference proceedings of IWCS 2021, the 14th edition of the International Conference on Computational Semantics. This conference is supported by the NWO-VICI grant "Lost in Translation—Found in Meaning" (288-89-003), known from its research and creation of the Parallel Meaning Bank.

The original plan was to have this conference take place in the beautiful city of Groningen, situated in the north of the Netherlands. However, due to the outbreak of the pandemic last year it became soon clear that this would not be a feasible option. So we decided to go ahead anyway and organize it as a fully online event. And indeed, it is the first time that IWCS is organized as an online event, spread over two days from 17–18 June 2021. Four satellite workshops are organised in the days before the conference:

- ISA-17: The Seventeenth Joint ACL - ISO Workshop on Interoperable Semantic Annotation;
- MMSR I: Beyond Language: Multimodal Semantic Representations;
- NALOMA'21: Natural Logic meets Machine Learning 2021;
- SemSpace 2021: Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science.

Going back to the main conference, the call for papers triggered 50 submissions (39 long and 11 short). Each paper was reviewed by three reviewers. There were three desk rejects. Eventually, 24 papers were accepted for the conference, of which one was withdrawn. This results in 19 long and 4 short papers with a final acceptance rate of 46% (49% for long and 36% for short papers). The final programme shows a lot of diversity with topics ranging from semantic parsing, question answering, knowledge extraction, and frame semantics. Two keynotes complement the programme, given by Rachel Rudinger (University of Maryland) and Mirella Lapata (University of Edinburgh).

We wish you a pleasant conference!

Groningen, 1 June 2021

Lasha Abzianidze
Johan Bos
Rik van Noord
Sina Zarrieß

Organisation

Program Chairs:

Johan Bos	University of Groningen
Rik van Noord	University of Groningen
Sina Zarrieß	University of Bielefeld

Local Chairs:

Johan Bos	University of Groningen
Rik van Noord	University of Groningen

Publication Chair:

Lasha Abzianidze	Utrecht University

Program Committee:

Rodrigo Agerri	Daisuke Bekki	Jean-Philippe Bernardy
Yuri Bizzoni	António Branco	Ellen Breitholz
Chris Brew	Paul Buitelaar	Harry Bunt
Aljoscha Burchardt	Stergios Chatzikyriakidis	Rui Chaves
Philipp Cimiano	Daoud Clarke	Paul Cook
Robin Cooper	Montse Cuadros	Paula Czarnowska
Philippe de Groote	Gerard de Melo	Rodolfo Delmonte
Markus Egg	Guy Emerson	Katrin Erk
Arash Eshghi	Kilian Evang	Meaghan Fowlie
Anette Frank	Diego Frassinelli	André Freitas
Mehdi Ghanimifard	Jonathan Ginzburg	Christine Howes
Elisabetta Jezek	Rohit Kate	Gene Kim
Ralf Klabunde	Nikhil Krishnaswamy	Staffan Larsson
Alex Lascarides	Alessandro Lenci	Zhaohui Luo
Aleksandre Maskharashvili	Louise McNally	Koji Mineshima
Yusuke Miyao	Richard Moot	Larry Moss
Sebastian Padó	Ludovica Pannitto	Sandro Pezzelle
Manfred Pinkal	Paul Piwek	Massimo Poesio
Christopher Potts	Violaine Prince	Stephen Pulman
Matthew Purver	Allan Ramsay	Christian Retoré
Kyle Richardson	German Rigau	Mats Rooth
Michael Roth	Mehrnoosh Sadrzadeh	Asad Sayeed
Nathan Schneider	Sabine Schulte im Walde	Rolf Schwitter
Ravi Shekhar	Carina Silberer	Mark Steedman
Tim Van de Cruys	Kees van Deemter	Carl Vogel
Shan Wang	Christian Wartena	Bonnie Webber
Matthijs Westera	Gijs Wijnholds	Hitomi Yanaka
Annie Zaenen	Roberto Zamparelli	Fabio Massimo Zanzotto

Keynote Speakers

Mirella Lapata, University of Edinburgh

Title: **The Democratization of Semantic Parsing via Zero-Shot Cross-lingual Learning**

Abstract: Semantic parsing is the task of mapping natural language utterances to machine-interpretable expressions such as SQL or a logical meaning representation. It has emerged as a key technology for developing natural language interfaces, especially in the context of question answering where a semantically complex question is mapped to an executable query to retrieve an answer, or denotation.

Datasets for semantic parsing scarcely consider languages other than English and professional translation can be prohibitively expensive. Recent work has successfully applied machine translation to localize parsers to new languages. However, high-quality machine translation is less viable for lower resource languages, and can introduce performance limiting artifacts, struggling to accurately model native speakers.

In this talk view cross-lingual semantic parsing as a zero-shot learning problem. We propose a multi-task encoder-decoder model to transfer parsing knowledge to additional languages using only English-Logical form paired data and unlabeled, mono-lingual utterances in each target language. Our encoder learns language-agnostic representations and is jointly optimized for generating logical forms or utterance reconstruction and against language discriminability. We frame zero-shot parsing as a latent-space alignment problem and find that pre-trained models can be improved to generate logical forms with minimal cross-lingual transfer penalty. Our parser performs above back-translation baselines and, in some cases, approaches the supervised upper bound.

Bio: Mirella Lapata is professor of natural language processing in the School of Informatics at the University of Edinburgh. Her research focuses on getting computers to understand, reason with, and generate natural language. She is the first recipient (2009) of the British Computer Society and Information Retrieval Specialist Group (BCS/IRSG) Karen Sparck Jones award, a Fellow of the ACL and the Royal Society of Edinburgh. She has also received best paper awards in leading NLP conferences and has served on the editorial boards of the Journal of Artificial Intelligence Research, the Transactions of the ACL, and Computational Linguistics. She was president of SIGDAT (the group that organized EMNLP) in 2018.

Rachel Rudinger, University of Maryland

Title: **When Pigs Fly and Birds Don't: Exploring Defeasible Inference in Natural Language**

Abstract: Commonsense reasoning tasks are often posed in terms of soft inferences: given a textual description of a scenario, determine which inferences are likely or plausibly true. For example, if a person drops a glass, it is likely to shatter when it hits the ground. A hallmark of such inferences is that they are defeasible, meaning they may be undermined or retracted with the introduction of new information. (E.g., we no longer infer that the dropped glass is likely to have shattered upon learning that it landed on a soft pile of laundry.) While defeasible reasoning is a long-standing topic of research in Artificial Intelligence (McCarthy, 1980; McDermott and Doyle, 1980; Reiter, 1980), it is less well studied in the context of contemporary text-based inference tasks, like Recognizing Textual Entailment (Dagan et al., 2005), or Natural Language Inference (MacCartney, 2009; Bowman et al., 2015). In this talk, I will present a new line of work that merges traditional defeasible reasoning with contemporary data-driven textual inference tasks. I argue that defeasible inference is a broadly applicable framework for different types of language inference tasks, and present examples for physical, temporal, and social reasoning.

Bio: Rachel Rudinger is an Assistant Professor of Computer Science at the University of Maryland, College Park. Previously, she obtained her PhD at John Hopkins University and spent a year as a Young Investigator at AI2 in Seattle. Her research focuses on problems in natural language understanding, including knowledge acquisition from text, commonsense inference, computationally-tractable semantic representations, and semantic parsing. She is also a contributing member of the Decompositional Semantics Initiative.

Table of Contents

Switching Contexts: Transportability Measures for NLP

Guy Marshall*†, Mokanarangan Thayaparan*†, Philip Osborne†, André Freitas†‡
Department of Computer Science, University of Manchester, United Kingdom†
Idiap Research Institute, Switzerland‡
{guy.marshall, philip.osborne}@postgrad.manchester.ac.uk
{mokanarangan.thayaparan, andre.freitas}@manchester.ac.uk

Abstract

This paper explores the topic of transportability, as a sub-area of generalisability. By proposing the utilisation of metrics based on well-established statistics, we are able to estimate the change in performance of NLP models in new contexts. Defining a new measure for transportability may allow for better estimation of NLP system performance in new domains, and is crucial when assessing the performance of NLP systems in new tasks and domains. Through several instances of increasing complexity, we demonstrate how lightweight domain similarity measures can be used as estimators for the transportability in NLP applications. The proposed transportability measures are evaluated in the context of Named Entity Recognition and Natural Language Inference tasks.

1 Introduction

The empirical evaluation of the quality of NLP models under a specific task is a fundamental part of the scientific method of the NLP community. However, commonly, many proposed models are found to perform well in the specific context in which they are evaluated and state-of-the-art claims are usually found not transportable to similar but different settings. The current evaluation metrics may only indicate which algorithm or setup performs best: they are unable to estimate performance in a new context, to demonstrate internal validity, or to verify causality. To offset this, statistical significance testing is sometimes applied in conjunction with performance measures (e.g. F1-score, BLEU) to attempt to establish validity. However, statistical significance testing has been shown to be lacking. Dror et al. (2018) reviewed NLP papers from ACL17 and TACL17 and found that only a third of these papers use significance testing. Further, many papers did not specify the type of test used, and some even employed an inappropriate statistical test.

Performance is measured in NLP tasks primarily through F1 score or task-specific metrics such as BLEU. The limited scope of these as performance evaluation techniques has been shown to have issues. Søgaard et al. (2014) highlights the data selection bias in NLP system performance. Gorman and Bedrick (2019) show issues of using standard splits, as opposed to random splits. We support their statement that "practitioners who wish to firmly establish that a new system is truly state-of-the-art augment their evaluations with Bonferroni-corrected random split hypothesis testing". In an NLI task, using SNLI and MultiNLI datasets with a set of different models, it has been shown that permutations of training data leads to substantial changes in performance (Schluter and Varab, 2018).

Further, the lack of transportability for NLP tasks has been raised by specialists in applied domains. For example, healthcare experts have expressed their frustration in the limitations of algorithms built in research settings for practical applications (Demner-Fushman and Elhadad, 2016) and the reduction of performance "outside of their development frame" (Maddox and Matheny, 2015). More generally, "machine learning researchers have noted current systems lack the ability to recognize or react to new circumstances they have not been specifically programmed or trained for" (Pearl, 2019).

The advantages of "more transportable" approaches, such as BERT, in terms of their performance in multiple different domains, is currently not expressed (other than the prevalence of such architectures across a range of state-of-the-art tasks and domains). To support analysis and investigation into the insight that could be gained by examination of these properties, we suggest metrics

* equal contribution

Proceedings of the 14th International Conference on Computational Semantics, pages 1–10
June 17–18, 2021.

and a method for measuring the transportability of models to new domains. This has immediate relevance for domain experts, wishing to implement existing solutions on novel datasets, as well as for NLP researchers wishing to assemble new dataset, design new models, or evaluate approaches.

To support this, we propose feature gradient, and show it to have promise as a way to gain lexical or semantic insight into factors influencing the performance of different architectures in new domains. This differs from data complexity, being a comparative measure between two datasets. We aim to start a conversation about evaluation of systems in a broader setting, and to encourage the creation and utilisation of new datasets.

This paper focuses on the design and evaluation of a lightweight transportability measure in the context of the empirical evaluation of NLP models. A further aim is to provide a category of measures which can be used to estimate the stability of the performance of a system across different domains. An initial transportability measure is built by formalising properties of performance stability and variation under a statistical framework. The proposed model is evaluated in the context of Named Entity Recognition tasks (NER) and Natural Language Inference (NLI) tasks across different domains.

Our contribution is to present a measure that evaluates the transportability and robustness of an NLP model, to evaluate domain similarity measures to understand and anticipate the transportability of an NLP model, and to compare state of the art models across different datasets for NER and NLI.

2 Relevant background and related work

2.1 Terminology

To quote Campbell and Stanley (2015), "External validity asks the question of generalizability: To what populations, settings, treatment variables, and measurement variables can this effect be generalized?". For Pearl and Bareinboim (2014), transportability is how generalisable an experimentally identified causal effect is to a new population where only observational studies can be conducted. "However, there is an important difference, not often distinguished, between what might be called the potential (or generic) transferability of a study and its actual (or specific) transferability to another policy or practice decision context at another time and place." (Walker et al., 2010)

Bareinboim and Pearl (2013) explore transfer of causal information, culminating in an algorithm for identifying transportable relations. Transportability in this sense does not permit retraining in the new population, and guides our choices in this paper. Other definitions of transfer learning allow for training of the model in the new context (Pan and Yang, 2010), or highlight the distinction between evidential knowledge and causal assumptions (Singleton et al., 2014).

2.2 Transportability: Models evaluated across different datasets

Rezaeinia et al. (2019) consider improving transportability by demonstrating word embeddings' accuracy degrades over different datasets, and propose an algorithmic method for improved word embeddings by using word2vec, adding gloVe when missing, and filling any further missing values with random entries. In a medical tagging task, Ferrández et al. (2012) used different train/test datasets, and compared precision and recall with self-trained vs transported-trained, finding that some tag-categories performed better than others. They postulate that degradation differences were due to the differing prevalence of entities in the transported training data. Another term from this domain is "portability", in the sense that a model could be successfully used with consideration of implementation issues such as different data formats and target NLP vocabularies (Carroll et al., 2012). Blitzer et al. (2007) created a multi-domain dataset for sentiment analysis, and propose a measure of domain similarity for sentiment analysis based on the distance between the probability distributions in terms of characteristic functions of linear classifiers.

In image processing, domain transfer is an active area of research. Pan et al. (2010) propose transfer component analysis as a method to learn subspaces which have similar data properties and data distributions in different domains. They state that domain adaptation is "a special setting of transfer learning which aims at transferring shared knowledge across different but related tasks or domains". In computer vision, Peng et al. (2019) combine multiple datasets into a larger dataset DomainNet, and consider multi-source domain adaptation, formalising for binary classification. They demonstrate multi-source training improves model accuracy, and publish baselines for state of the art methods.

2.3 Generalisability

The language used in literature is not consistent. Bareinboim and Pearl (2013) highlights that generalisability goes under different "rubrics" such as external validity, meta-analysis, overgeneralisation, quasi-experiments and heterogeneity.

Boulenger et al. (2005) disambiguate terms in the context of healthcare economics (such as generalisability, external validity, and transferability), and created a self-reporting checklist to attempt to quantify transferability. They define generalisability as "the degree to which the results of a study hold true in other settings", and "the data, methods and results of a given study are transferable if (a) potential users can assess their applicability to their setting and (b) they are applicable to that setting". They advocate a user-centric view of transferability, considering specific usability aspects such as explicit currency conversion rates.

Antonanzas et al. (2009) create a transferability index at general, specific and global levels. Their "general index" is comprised of "critical factors", which utilise Boulenger et al.'s factors, adding subjective dimensions.

3 Transportability in NLP

3.1 Definitions

To support a rigorous discussion, notational conventions are introduced. Extending the choices of Pearl and Bareinboim (2011), we denote a domain $\mathcal{D}$ with population Π, governed by feature probability distribution P, which is data taken from a particular domain. We denote the *source* with a 0 subscript.

Definition 1. *Generalisability: A system Ψ has performance p for solving task T_0 in domain $\mathcal{D}_0$. Generalisability is how the system Ψ performs for solving task T_i in domain $\mathcal{D}_j$, relative to the original task, without retraining.*

Special cases, such as transportability or transference, have some $i, j = 0$ in the definition above.

Definition 2. *Transportability: A system Ψ has performance p for solving task T_0 in domain $\mathcal{D}_0$. Transportability is the performance of system Ψ for solving task T_0 in a new domain $\mathcal{D}_i$, relative to the original task, without retraining.*

Across multiple $\mathcal{D}_i$, we have relative performance $\tau_p(\mathcal{D}_0, \mathcal{D}_i)$, from which we can establish statistical measures for transportability performance and variation.

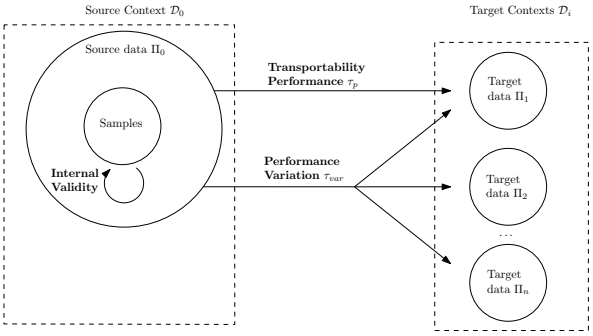


Figure 1: Schematic representation of the definitions

Transfer learning is a specific way of achieving transportability (between populations or domains) or generalisability (including between tasks). Singleton et al. (2014) state that "transport encompasses transfer learning in attempting to use statistical evidence from a source on a target, but differs by incorporating causal assumptions derived from a combination of empirical source data and outside domain knowledge.". Note that this is different to *generalisation* in the Machine Learning sense, which is akin to internal validity (Marsland, 2011). Figure 1 shows the definitions associated with transportability discussed in this paper.

Table 1 summarises terminology, of how the target differs from source $(\Psi_0, T_0, \mathcal{D}_0(\Pi_0))$.

Term	Ψ	**T**	$\mathcal{D}$	Π
Cross-validation	0	0	0	i
New modeling	i	0	0	0
Transportability	0	0	i	i
Transferability	0	0,i	i	i
Generalisability	0	0,i	0,i	0,i

Table 1: Terminology through variation from a source. Table body is subscripts.

Chance, *bias* and *confounding* are the three broad categories of "threat to validity". Broadly, chance and bias can be assessed by cross-validity, as it applies a model to the same task in the same domain on different data population. Confounding, error in interpretation of what is being measured, is more difficult to assess. Transportability is concerned with the transfer of learned information, with particular advances in the transport of causal knowledge.

Generalisability is the catch-all term for how externally valid a result or model is. Any combination of task, domain and data can be used.

3.2 Transportability performance

We define transportability performance τ_p as the gradient of the change in the performance metric's score from one domain to another. This measure does not take into account the underlying probability distributions, only the change in resulting performance measure.

$$\tau_p(\mathcal{D}_0, \mathcal{D}_i) = \frac{\mathrm{p}(\Psi, T, \mathcal{D}_i)}{\mathrm{p}(\Psi, T, \mathcal{D}_0)} \tag{1}$$

The measure uses a ratio in order to allow comparison between different systems. To generalise this measure across different settings, we can take an average to give Equation 2. Note that this is the average percentage change in performance, not an aggregated performance measure.

$$\tau_p(\mathcal{D}_0) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathrm{p}(\Psi, T, \mathcal{D}_i)}{\mathrm{p}(\Psi, T, \mathcal{D}_0)} \tag{2}$$

An analogous definition holds for different tasks over the same domain, $\tau_p(T)$.

3.3 Performance variation

Performance variation reflects how stable performance is across different contexts and can include, for example, to what extent the sampling method from the source data effects the performance metric of the algorithm. Part of this is data representativeness, the extent to which the source data representation also represents the target data.

More formally, performance variation $\tau_s(\Psi, T, \mathcal{D})$ is the change in performance of $(\Psi, T, \mathcal{D})$ across different contexts. This is useful in order to gain specific insight into external validity and generalisability. Indeed, we can assess the change in performance between source context $\mathcal{D}_0$ and target context $\mathcal{D}_i$. The source context has a privileged position, in that it is this space which the "learning" takes place, and the proposed metric for performance variation to multiple different domains is based on τ_p to reflect this. Through repeated measurement in different contexts, we can go further.

Definition 3. *Performance Variation: For a model trained on domain $\mathcal{D}_0$ and applied on n new domains $\mathcal{D}_i$, we define the performance variation as the coefficient of variation of performance across this set of domains so that:*

$$\tau_{var}(\mathcal{D}_0) = \left(1 + \frac{1}{4n}\right) \frac{\sqrt{\frac{\sum_{i=1}^{n}(\tau_p(\mathcal{D}_0, \mathcal{D}_i) - \tau_p(\mathcal{D}_0))^2}{n-1}}}{\tau_p(\mathcal{D}_0)} \tag{3}$$

The $1 + \frac{1}{4n}$ term corrects for bias. In order to be meaningful, the target contexts must to have a good coverage of different domains. Enumerating these would be a task of ontological proportions, but can be pragmatically approximated by using the available Gold Standard datasets.

We can also assess ability to generalise not just over different domains, but also different tasks, provided they can be meaningfully assessed by the same performance measure. We can consider n different domain-task combinations, and with $\overline{\tau_p} = \sum_{i,j=0}^{n} \tau_p(\Psi, T_i, \mathcal{D}_j)/n$, this gives a more general form for Equation 3, with n large:

$$\tau_{var} = \frac{\sqrt{\frac{\sum_{i\geqslant 0, j\geqslant 0}(\tau_p(\Psi, T_i, \mathcal{D}_j) - \overline{\tau_p})^2}{n-1}}}{\overline{\tau_p}} \tag{4}$$

In the case where different tasks cannot be assessed by the same measure, we are still able to compare different systems by looking at how the respective measures change.

3.3.1 Performance variation properties

For a purely random system, the transportability should be related to how similar the distributions of "answers" in the test dataset are. A random system should really be transportable by our measures. Similarly, we can consider trivial systems, such as identity and constant functions, which are necessarily entirely transportable. That is, for a system that is an identity function $\Psi = I$, $\tau_p = f(P)$, and $\tau_{var}(I, T, \mathcal{D}_i) = \tau_{var}(I, T, \mathcal{D}_j) = 0$, $\forall i, j$. Note that we would not expect the same performance of these functions on different tasks.

A stable system will have $\tau_{var}(\Psi, T, \mathcal{D}_0) \approx \tau_{var}(\Psi, T, \mathcal{D}_i) \forall i$, reflecting that it is resilient to the domain on which it is trained.

3.3.2 Factors influencing performance variation

Through repeated measurement, we can quantify how F_1-score changes with respect to different measures A (e.g. dataset complexity), $\frac{\partial F_1}{\partial A}$, with other properties held constant.

NLP system performance is dependent on A. This list may include gold standard feature distribution (in terms of representativeness of the semantic

or linguistic phenomena), and task difficulty or sensitivity.

Users of NLP systems would benefit from being able to estimate the performance of an existing NLP system on a new domain, without performing the full implementation. Important for the performance of an NLP system, especially for few or zero shot learning, is having a common set of features (or phenomena) across domains. We proceed to propose three measures of increasing complexity, in order to attempt to understand how "similar" two domains are.

Lexical feature difference: A measure grounded on lexical features (i.e. bag of words). The intuition behind this measure is for treating the set of lexical features as a representation. Linguistic space is observed as materialised tokens, which in turn are in some higher-dimensional semantic space, which enable interpretation. The measure considers the overlap of these linguistic spaces, and indeed the extent to which the linguistic space is covered by the data. Due to the simplicity of this measure, correlation between this and actual transportability performance is likely to be weaker than other measures but is simpler to calculate.

$$\text{Lexical Feature Difference} = 1 - \frac{|\mathcal{D}_i \cap \mathcal{D}_0|}{|\mathcal{D}_i|}, i > 0 \quad (5)$$

Where $|\mathcal{D}_i|$ is the number of features in the target domain $\mathcal{D}_i$, and $|\mathcal{D}_i \cap \mathcal{D}_0|$ is the number of features overlapping. This measure is then the proportion of unseen features in the new dataset. If all features of $\mathcal{D}_i$ are found in $\mathcal{D}_0$, then the feature difference is 0. If no features of $\mathcal{D}_i$ are found in $\mathcal{D}_0$, then the feature difference is 1. The feature overlap is task specific, and therefore appropriate to consider for transportability, but not generalisability.

In the simplest case, the transported performance of a bag of words model should be precisely the lexical feature difference combined with distributions of the source and target domains. The feature set can range from binary lexical features to latent vector spaces. For different models, which target different aspects of semantic phenomena, different semantic and syntactic features will matter more. For this reason, considering a set of measures for domain complexity is warranted. In the context of this work, two measures are used over more complex feature spaces.

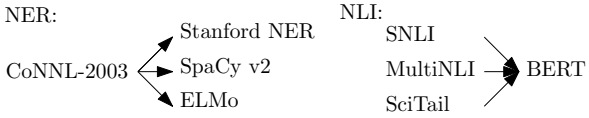


Figure 2: Overview of the experiments undertaken, indicating the models being applied to each dataset

Cosine distance: Specifically, we use Doc2Vec (Le and Mikolov, 2014) to embed the documents from each domain in a 300-dimensional feature-vector space, normalise, and calculate cosine distance to compare source and target domains.

Kullback–Leibler divergence: Considering each domain as a distribution of features, we can use relative entropy to understand the difference between the source and target domains. Similar to cosine distance, we convert the corpus to a vector using Doc2Vec and normalize. We treat these values as discrete probability distributions to calculate the KL divergence.

The usefulness of any of these domain similarity measures depends on the semantic phenomena and supporting corpora underlying the system, for example if the system requires a large training dataset, it may be more appropriate to use a measure which considers the underlying probability distributions in each feature. In this case, we can restrict to the case of the same task in order to keep the essential features reasonably consistent across domains. This makes this a measure of transportability (rather than generalisability).

There are additional dimensions of transportability potentially worthy of further investigation and quantification: (i) domain similarity (e.g. missing features), (ii) data efficiency (redundant/repeated features), (iii) data preparation (initial setup and formatting) and (iv) data manipulation required (data pipeline).

4 Experiments

4.1 Setup

The experiments aim to evaluate the consistency of the proposed transportability measures in the context of two standard tasks: named entity recognition and natural language inference. For reproducibility purposes the code and supporting data are available online[1].

We calculated the F1 score of multiple models on

[1] https://github.com/ai-systems/transportability

Dataset		Model		
		Stanford	SpaCy	ELMo
CoNLL-2003	Train	98.69	99.32	99.97
	Dev	93.22	81.56	98.17
	Test	88.78	88.11	93.79
Wiki		66.31	52.14	79.4
WNUT	Train	51.63	27.03	36.3
	Dev	53.59	32.23	48.8
	Test	47.11	26.28	58.1

Table 2: NER F1 scores for different models trained on CoNLL dataset transported across different corpora

multiple datasets (Figure 2). Note that in general the applicability of the proposed transportability measures are not limited to the use of F1 score, but this is simpler as the same measure applies for both tasks. All models and datasets are standard. For NER, the datasets were chosen as they have the consistent tags: Location, Person and Organisation. Stanford NER (Finkel et al., 2005) is a CRF classifier, SpaCy v2 is a CNN, ELMo (Peters et al., 2018) is a vector embedding model which outperforms GloVe and word2vec. Each of the three models used are trained on the CoNLL-2003 dataset (Sang and De Meulder, 2003). We evaluated these models on CoNLL-2003, Wikipedia NER (Ghaddar and Langlais, 2017) (Wiki) and WNUT datasets (Baldwin et al., 2015) for NER in twitter microposts.

For NLI, we chose to use standard datasets. SNLI (Bowman et al., 2015) is well established with a limited range of NLI statements, MultiNLI (Williams et al., 2018) is multigenre with a more diverse range of texts, and SciTail (Khot et al., 2018) is based on scientific exam questions. We applied BERT (Devlin et al., 2018), a state of the art embedding model, to these datasets.

4.2 Results

NER: Table 2 shows results for the NER task, trained on CoNLL. Unsurprisingly, all models performed better when the target was in the CoNLL domain. The reduced performance on Wiki was more extreme than expected, particularly for ELMo, which was expected to be resilient to domain change (i.e. transportable). Table 6 and Table 4 illustrate the transportability and domain similarity scores for different NER models respectively.

NLI: Table 3 shows results for the NLI task, using BERT. We find that, despite the vast training data, BERT's performance is substantially higher when it has been trained on data from that domain. BERT trained on SciTail performs poorly when transported to SNLI or MultiNLI. Table 7 and Table 5 illustrates the transportability and domain similarity scores for different NLI corpora.

4.3 Analysis

Every model had $\tau_p \ll 1$, meaning they performed worse on the new domain. This is as expected, though this would not be true in general.

NER: Examining the F1 scores (88.11 vs. 88.78) of SpaCy and Stanford they appear almost comparable. However, the latter transports much more effectively, with τ_p score difference (0.671 Vs 0.524 when transporting to Wiki) (refer Table 6).

ELMo is one of the state of the art approaches for NER, as evidenced by the high F1 scores for the source corpus. However, Stanford NER transports equally well, and when transported outperforms ELMo for twitter domain. While the absolute F1 score difference between them is 5, the τ_p scores are almost identical, with a difference of 0.003. In terms of transportability, it is notable that an approach that employs CRF tagger with linguistic features outperforms significantly the CNN-based SpaCy approach and stands in comparison to a computationally expensive model like ELMo.

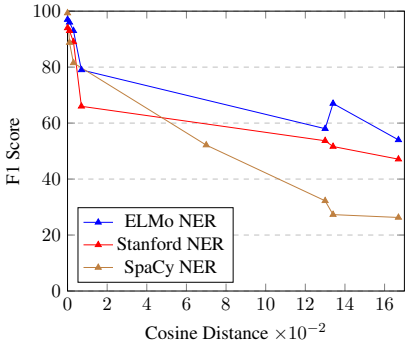
Stanford NER also has the lowest τ_{var}. This indicates this to be the most robust model out of the three. This conclusion was facilitated by the τ_p and τ_{var} measures.

NER for English is assumed to be an accomplished task as supported by the traditional F1 scores. By using τ_p we argue that there is a need for more robust models, with better transportability performance.

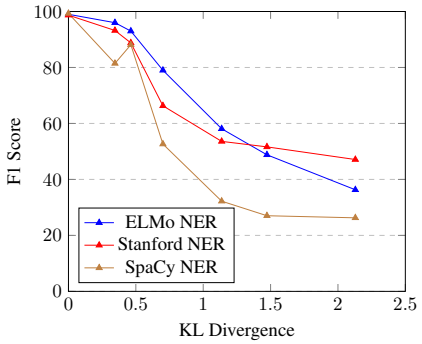
Figure 3a and Figure 3b illustrates the decrease in F1 scores as cosine distance and KL divergence increase. A simple 3 parameter non-linear regression model on KL Divergence and Cosine distance is able to predict the F1 score with an mean error of 3.33 and 2.66 respectively. Considering the lexical difference has similar results (Table 4). This implies that by using these measures we may be able to anticipate the accuracy of a model in a new domain based on easy to compute domain similarity, which is straightforward to compute.

Source Dataset	Target Dataset							
	SNLI Dataset			MultiNLI Dataset		SciTail Dataset		
	Train	Dev	Test	Train	Dev	Train	Dev	Test
SNLI (Train)	96.81	90.83	90.40	72.51	72.29	54.04	61.34	52.72
Multi NLI (Train)	77.13	79.05	79.31	97.78	83.50	66.52	67.79	67.26
SciTail (Train)	42.68	44.36	44.20	47.49	44.49	99.88	94.78	93.08

Table 3: NLI accuracy scores for BERT model trained on one dataset transported to a different dataset



(a) NER F1 scores Vs Doc2Vec cosine distance from training (CoNLL) corpus



(b) NER F1 scores Vs KL Divergence from training (CoNLL) corpus

Figure 3: NER F1 score plotted against different measures of corpus similarity

Dataset		Lexical	Cosine	KL Divergence
	Train	0.000	0.000	0.000
CoNLL	Dev	0.121	0.001	0.345
	Test	0.197	0.003	0.463
Wiki		0.290	0.007	0.701
	Train	0.421	0.134	2.129
WNUT	Dev	0.511	0.167	1.473
	Test	0.481	0.130	1.137

Table 4: Domain similarity scores between the training corpus (CoNLL-2003) across other NER datasets

NLI: Applying BERT to different domains was not as resilient to domain transport as we expected. The average τ_p is 0.612 over transported domains, despite these being standard corpora from the domains. We found MultiNLI(Train) to be more transportable than the others, since its performance in new domains is not much worse than new data from the same domain. This is as expected, since MultiNLI has been built to have good domain coverage. Specifically, MultiNLI has $\tau_p = 0.744$ and $\tau_{var} = 8.582$, whilst SNLI has $\tau_p = 0.646$ and $\tau_{var} = 15.22$ and SciTail has $\tau_p = 0.446$ and $\tau_{var} = 3.921$. SciTail transports poorly, and does so reliably! SNLI transports in between, but variably, being quite "hit or miss" with different samples of SciTail. These results suggest a threshold for τ_p of perhaps 0.8 as being "appropriate" for transportability performance. A threshold for τ_{var} is more difficult to establish and would benefit from further investigation. Clearly, these measures depend on the domains chosen.

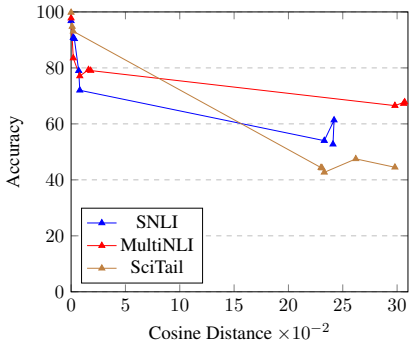
As with NER, we found lexical difference indicative of transported performance, and that for NLI, accuracy scores decrease with increasing lexical difference, cosine distance and KL divergence (Tables 3 and 5, and Figures 4a and 4b). A simple 3 parameter non-linear regression model on KL Divergence and Cosine distance is able to predict the accuracy score with an mean error of 3.98 and 1.95 respectively.

4.4 Discussion

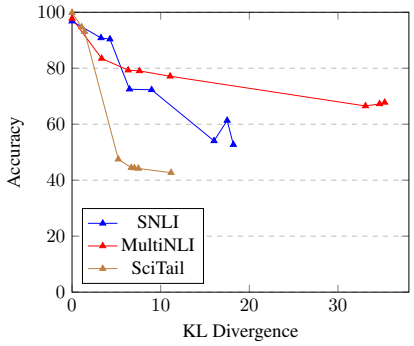
τ_p and τ_{var} as complementary to traditional measures. We are not breaking new ground in terms of evaluation methodology, but the experiments demonstrate that traditional F1 and accuracy measures do not capture a complete picture. Transportability measure are not only simple enough to calculate and convey but also evaluates a model

Dataset	Measurement	SNLI			MultiNLI		SciTail		
		Train	Dev	Test	Train	Dev	Train	Dev	Test
SNLI (Train)	Lexical	0.000	0.003	0.003	0.086	0.088	0.136	0.115	0.119
	Cosine	0.000	0.002	0.002	0.008	0.007	0.233	0.242	0.242
	KL Divergence	0.000	3.277	4.283	6.489	8.982	16.02	17.50	18.20
MultiNLI (Train)	Lexical	0.008	0.008	0.008	0.000	0.008	0.063	0.063	0.047
	Cosine	0.008	0.018	0.016	0.000	0.002	0.298	0.307	0.306
	KL Divergence	11.07	7.613	6.333	0.000	3.342	33.10	35.27	34.69
SciTail (Train)	Lexical	0.282	0.282	0.282	0.277	0.278	0.000	0.028	0.025
	Cosine	0.233	0.230	0.231	0.262	0.298	0.000	0.001	0.002
	KL Divergence	11.17	7.04	7.492	5.220	6.682	0.000	1.097	1.424

Table 5: Domain similarity scores between the source training corpus and target corpora



(a) NLI accuracy Vs Doc2Vec cosine distance from source corpus



(b) NLI accuracy Vs Lexical Divergence from training corpus

Figure 4: NLI accuracy score plotted against different measures of corpus similarity

	Stanford	SpaCy	ELMo
$\tau_p(wiki)$	0.671	0.524	0.794
$\tau_p(wnut)$	0.514	0.287	0.477
$\tau_p(wnut\ \&\ wiki)$	0.553	0.346	0.556
τ_{var}	15.051	35.171	32.666

Table 6: Transportability measures for NER models

	SNLI	MultiNLI	SciTail
τ_p	0.646	0.744	0.446
τ_{var}	15.22	8.582	3.921

Table 7: Transportability measures for NLI corpora

with regards to generalisability and robustness.

Low cost ways of anticipating performance for a new task or domain. Most of the state of the art models are computationally expensive. With the transportability and domain similarity measures we are able to predict performance in a new domain with reasonable accuracy. These similarity measures are relatively simpler to run.

5 Conclusion

We have presented a model of transportability for NLP tasks, together with metrics to allow for the quantification in the change in performance. We have shown that the proposed transportability measure allows for direct comparison of NLP systems' performance in new contexts. Further, we demonstrated domain similiarity as a measure to model corpus and domain complexity, and predict NLP system performance in unseen domains. This paper lays the foundations for further work in more complex transportability measures and estimation of NLP system performance in new contexts.

References

Fernando Antonanzas, Roberto Rodríguez-Ibeas, Carmelo Juárez, Florencia Hutter, Reyes Lorente, and Mariola Pinillos. 2009. Transferability indices

for health economic evaluations: methods and applications. *Health economics*, 18(6):629–643.

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

Elias Bareinboim and Judea Pearl. 2013. A general algorithm for deciding transportability of experimental results. *Journal of causal Inference*, 1(1):107–134.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

Stephanie Boulenger, John Nixon, Michael Drummond, Philippe Ulmann, Stephen Rice, and Gerard de Pouvourville. 2005. Can economic evaluations be made more transferable? *The European Journal of Health Economics*, 6(4):334–346.

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 (EMNLP)*. Association for Computational Linguistics.

Donald T Campbell and Julian C Stanley. 2015. *Experimental and quasi-experimental designs for research*. Ravenio Books.

Robert J Carroll, Will K Thompson, Anne E Eyler, Arthur M Mandelin, Tianxi Cai, Raquel M Zink, Jennifer A Pacheco, Chad S Boomershine, Thomas A Lasko, Hua Xu, et al. 2012. Portability of an algorithm to identify rheumatoid arthritis in electronic health records. *Journal of the American Medical Informatics Association*, 19(e1):e162–e169.

D Demner-Fushman and Noemie Elhadad. 2016. Aspiring to unintended consequences of natural language processing: a review of recent developments in clinical and consumer-generated text processing. *Yearbook of medical informatics*, 25(01):224–233.

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*.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Óscar Ferrández, Brett R South, Shuying Shen, F Jeff Friedlin, Matthew H Samore, and Stéphane M Meystre. 2012. Generalizability and comparison of automatic clinical text de-identification methods and resources. In *AMIA Annual Symposium Proceedings*, volume 2012, page 199. American Medical Informatics Association.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Abbas Ghaddar and Phillippe Langlais. 2017. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2786–2791.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Thomas M Maddox and Michael A Matheny. 2015. Natural language processing and the promise of big data: Small step forward, but many miles to go. *Circulation. Cardiovascular quality and outcomes*, 8(5):463.

Stephen Marsland. 2011. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. 2010. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60.

Judea Pearl and Elias Bareinboim. 2011. Transportability of causal and statistical relations: A formal approach. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Judea Pearl and Elias Bareinboim. 2014. External validity: From do-calculus to transportability across populations. *Statistical Science*, 29(4):579–595.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Seyed Mahdi Rezaeinia, Rouhollah Rahmani, Ali Ghodsi, and Hadi Veisi. 2019. Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117:139–147.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Natalie Schluter and Daniel Varab. 2018. When data permutations are pathological: the case of neural natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4935–4939.

Kyle W Singleton, Alex AT Bui, and William Hsu. 2014. Transfer and transport: incorporating causal methods for improving predictive models. *Journal of the American Medical Informatics Association*, 21(e2):e374–e375.

Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy, and Héctor Martínez Alonso. 2014. What's in a p-value in nlp? In *Proceedings of the eighteenth conference on computational natural language learning*, pages 1–10.

Damian G Walker, Yot Teerawattananon, Rob Anderson, and Gerry Richardson. 2010. Generalisability, transferability, complexity and relevance. In *Evidence-Based Decisions and Economics*, pages 56–66. Wiley and Sons.

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. Association for Computational Linguistics.

Applied Temporal Analysis: A Complete Run of the FraCaS Test Suite

Jean-Philippe Bernardy **Stergios Chatzikyriakidis**
Centre for Linguistic Theory and Studies in Probability
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
firstname.lastname@gu.se

Abstract

In this paper, we propose an implementation of temporal semantics that translates syntax trees to logical formulas, suitable for consumption by the Coq proof assistant. The analysis supports a wide range of phenomena including: temporal references, temporal adverbs, aspectual classes and progressives. The new semantics are built on top of a previous system handling all sections of the FraCaS test suite except the temporal reference section, and we obtain an accuracy of 81 percent overall and 73 percent for the problems explicitly marked as related to temporal reference. To the best of our knowledge, this is the best performance of a logical system on the whole of the FraCaS.

1 Introduction

The semantics of tense and aspect has been a long standing issue in the study of formal semantics since the early days of Montague Grammar and a number of different ideas have been put forth to deal with them throughout the years. Recent proposals include the works of the following authors: Dowty (2012); Prior and Hasle (2003); Steedman (2000); Higginbotham (2009); Fernando (2015). The semantics of tense and aspect have been also considered in the study of Natural Language Inference (NLI). The various datasets for NLI that have been proposed by the years contain examples that have some implicit or explicit reliance on inferences related to tense and aspect. One of the early datasets used to test logical approaches, the FraCaS test suite (Cooper et al., 1996) contains a whole section dedicated to temporal and aspectual inference (section 7 of the dataset). This part of the FraCaS test suite has been difficult to tackle. That is, so far, no computational system has been capable to deal with it in its entirety: when authors report accuracy over the FraCaS test suite they skip this section. In fact, they also often skip the anaphora and ellipsis sections, the exception being the system presented by Bernardy and Chatzikyriakidis (2017, 2019), which includes support for anaphora and ellipsis but still omit the temporal section.[1] In this paper, we take up the challenge of providing a computationally viable account of tense and aspect to deal with the section 7 of the FraCaS test suite. Our account is not meant to be a theoretically extensive account of tense and aspect, but rather an account that is driven by the need to cover the test suite in a way that is general enough to capture the test suite examples, *while still covering the rest of the FraCaS test suite*.

The account is evaluated on the entailment properties of various temporal and aspectual examples, as given by the test suite. As such, we are not getting into the discussion of how tense and aspect might affect grammaticality or infelicitousness of various sentences. We assume that the sentences of the FraCaS suite are syntactically and semantically correct, and strive to produce accurate logical representations given that assumption. We further assume that the entailment annotations of various problems are valid, and we use those to evaluate the correctness of the logical representations of sentences.

The paper is structured as follows: in Section 2, we give a brief summary of the computational frameworks whose various subsystems rely on. In particular, the Grammatical Framework is used to construct the syntactic parser, the Coq proof assistant checks all the reasoning and a monad-based dynamic semantics deals with Montague-style se-

[1] One can consider that MacCartney and Manning (2007) have made a run against the whole test suite. However, they do not deal with multi-premise cases. Consequently only 36/75 cases in the temporal section are attempted. The general accuracy of the system is .59, and .61 for the temporal section. Our system, as shown Table 1, presents considerable improvements in coverage and accuracy over that of MacCartney and Manning.

Proceedings of the 14th International Conference on Computational Semantics, pages 11–20
June 17–18, 2021.

mantics, and references (anaphora). We also provide some brief remarks on temporal semantics. In Section 3, we discuss the main aspects of the compositional semantics of our system, using various examples from the FraCaS suite to illustrate its effectiveness. In Section 5, we evaluate how our system performs with respect to the FraCaS suite. We ran the system across the whole suite: our system is thus the first which is capable of handling the complete FraCaS test suite. Yet, we are interested in particular in the performance on the temporal section. In Section 6, we conclude and discuss avenues for future work.

2 Temporal-Semantics in a Logic-based NLI System

Our temporal analysis places itself in the context of a complete NLI system – which is why we can test it on the FraCaS suite. In this section we give a brief overview of the phases of the system, referring the reader to published work for details.

GF The first phase of the system, parsing, is taken care of by the Grammatical Framework (GF, Ranta (2004)), which is a powerful parser generator for natural languages, based on type-theoretical abstract grammars. The present work leverages a syntactic representation of the FraCaS test suite in GF abstract syntax, in effect a GF FraCaS treebank (Ljunglöf and Siverbo, 2011). Thanks to this, we skip the parsing phase and avoid any syntactic ambiguity.

For the purpose of this paper, the important feature of GF syntax is that it aims at a balance of sufficient abstraction to provide a semantically-relevant structure, but at the same time it embeds sufficiently many syntactic features to be able to reconstruct natural-language text. That is, the parse trees generally satisfy the homomorphism requirement of Montague (1970, 1974), and we can focus on the translation of syntactic trees to logical forms. Consequently, the system presented here does not aim at textual natural language understanding, but rather provides a testable, systematic formal semantics of temporal phenomena. Example (1) shows an example abstract syntax tree and its realisation in English.

Dynamic Semantics Parse trees are then processed by a dynamic semantic component. Its role is essentially to support (non-temporal) anaphora, using a monadic-based dynamic semantics, generally following the state of the art in this matter (Unger, 2011; Charlow, 2015, 2017). Our particular implementation has weaknesses in certain areas (including group readings and counting; see Bernardy et al. (2020) for details) but non-temporal anaphoroi in the testsuite are generally resolved as they should be: on the whole accuracy is not affected significantly by issues in this subsystem.

As it is the case for other basic phenomena, there is not much interaction between our treatment of time and non-temporal anaphora. Critical exceptions are discussed in Section 3 and Section 5.

Montagovian Semantics Non-withstanding special support for anaphora, the core of the translation of syntax trees to logical form follows a standard montagovian semantics. In brief, sentences are interpreted as propositions, verbs and noun-phrases as predicates. We use type-raising of noun-phrases, to support quantifiers (Montague, 1974).

We support additionally the basic constructions and phenomena present in the testsuite, including adjectives, adverbs, nouns, verbs, anaphora, etc. The method is outlined by Montague (1970, 1973), but we direct the reader to our previous work for details Bernardy and Chatzikyriakidis (2017, 2019) , but the particular treatment of such phenomena is essentially independent from our treatment of time: in this paper we simply ignore these aspects beyond the fact that they are handled correctly in the FraCaS testsuite, except in a few pathological cases.

Inference using Coq Logical forms are then fed to the Coq interactive theorem prover (proof assistant). Coq is based on the calculus of co-inductive constructions (Werner, 1994) We do not use any co-induction (or even induction) in this paper, relying on the pure lambda-calculus inner core of Coq. Coq is a very powerful reasoning engine that makes it fit for implementing natural language semantics. Coq also supports dependent typing and subtyping. Both concepts are instrumental in expressing NL semantics (Chatzikyriakidis and Luo, 2014). Besides, on a more practical side, it works well for the the task of NLI, when the latter is formalised as a theorem proving task: its many tactics mean that many tasks in theorem proving are trivialised. In particular, all problems of time-intervals inclusion, which occur in every temporal problems, are solved with Coq's linear arithmetic tactic.

3 Our Treatment of Time

In montagovian semantics, (intransitive) verbs are one-place predicates; in types, they are functions from entities to propositions ($e \to t$). Our basic approach is to generalise the interpretation of verbs, so that it takes two additional time parameters, one corresponding to the starting time of the action and one corresponding to its stopping time ($(e \times time \times time) \to t$). For example, if John walked between t_0 and t_1, we would have: $walk(john, t_0, t_1)$. From now on we will call an interval of time points $[t_0, t_1]$ a timespan, where t_0 and t_1 are elements of the $time$ type, which is represented in Coq as an abstract ordered ring. Every timespan $[t_0, t_1]$ has the property $t_0 \leq t_1$: it starts no later than it stops. (We are thus using a simple Newtonian model of time, corresponding to a layman intuition of a linear constant flow of time.)

In principle, common nouns and adjectives should undergo the same procedure. For simplicity we will however only consider verbs from now on. (In fact, even in our implementation we chose not to extend nouns nor adjectives with timespan parameters. This choice limits the increase in complexity of the formulas compared to non-temporal semantics, at the expense of inaccuracy for a couple of problems in the FraCaS test suite: problems 271 and 272 use a an adjective as a copula which is subject to temporal reasoning.)

(271) **A** unknown
P1 Smith was present.
P2 Jones was present.
P3 Smith was present after Jones was present.
H Jones was present before Smith was present.

Temporal Context We adjust the montagovian semantics so that the interpretation of every category (propositions, verb phrases, etc.) takes a *temporal context* as an additional parameter, which serves as a time reference for the interpretation of all time-dependent semantics within the phrase. (While some categories do not need this temporal context, we pass it everywhere for consistency.) This context propagates through the compositional interpretation down to lexical items with atomic representation (verbs). By default, every interpretation passes the temporal context down to its components without changing it. However some key elements will act on it on nontrivial ways, which we proceed to detail below.

This temporal context is an *optional* timespan. That is, it can be a timespan or an explicitly unspecified context. The timespan in the context is optional because, in certain situations, the semantics is different depending on whether a timespan has been specified externally or not, as we explain below. A non-present timespan will be represented as $-$. If a semantic function does not depend on the temporal context at all, we will write $*$ instead.

Tenses The principal non-trivial manipulators of timespans are tense markers. In our syntax, inherited from GF, tenses are represented syntactically as an attribute of clauses. An illustration of a past-tense clause and its interpretation follows in Example (1). Notice in particular the $past$ argument to the $useCl$ constructor.

(1) A scandinavian won the nobel prize.

$$\begin{array}{l} useCl\, past\, pPos \\ (predVP\, (detCN\, (detQuant\, indefArt\, numSg) \\ \qquad scandinavian\_CN) \\ \quad (complSlash\, (slashV2a\, win\_V2) \\ \quad (detCN\, (detQuant\, indefArt\, numSg) \\ \qquad nobel\_prize\_CN))) \end{array}$$

In our semantics we deal only with present and past tenses (simple and continuous). Indeed we find that FraCas does not exercise additional specific tenses. (When a more complicated tense is used, the additional information is also carried by adverbs or adverbial phrases, in a more specific way). While we believe that many other tenses can be captured under the same general framework, we leave a detailed study to further work.

Even though we discuss a refinement to handle the past continuous at the end of this section, the procedure to handle tense annotations is as follows:

- If the tense is the past, and the temporal context is unspecified, then we locally quantify over a time interval $[t_0, t_1]$, such that $t_1 < now$, where now is a logical constant representing the current timepoint. The temporal context then becomes this interval.
- If the tense is the present and the temporal context is unspecified, then the temporal context becomes the simple (now, now) interval.
- If the temporal context is specified (for example due to the presence of an adverb or an adverbial clause, such as "before James swam"),

then the tense does not create a new interval, but it may constrain it. Typically, a past tense adds the constraint that the temporal context ends before the timepoint now.

Temporal Adverbs The other single most important source of interesting timespans are adverbs. Most of the temporal adverbs fall in either of the following categories:

exact For such adverbs, an exact interval is provided. In fact, such adverbs typically specify a single point in time (so the start and the end of the interval coincide).

$$[\![\text{at 5 pm}, s]\!](*) = [\![s]\!](5pm, 5pm)$$

existentially quantifying The majority of temporal adverbs existentially quantify over a timespan. Examples include "since 1991", "in 1996", "for two years", etc. The common theme is to introduce the interval and then restrict its bounds or its duration in some way. Sometimes the restriction is an equality, as in "for exactly two hours". In the following example we show the inclusion constraint, for "in 1992".

$$[\![\text{in 1992}, s]\!](*) = \exists t_1, t_2.[t_1, t_2] \subseteq 1992, [\![s]\!](t_1, t_2)$$

In the FraCaS test suite, we normally do not find several time-modifying adverbs modifying a single verb phrase. Indeed, sentences such as "in 1992, in 1991 john wrote a novel" are infelicitous. This justifies ignoring the input timespan in the above interpretation – we are in particular not interested in modelling felicity with our semantics, only giving an accurate semantics when the input is felicitous.

universally quantifying A few adverbs introduce intervals via a universal quantification (sometimes with a constraint). Examples include "always" and "never".

If there is no explicit time context, then "always" has no constraint on the interval, otherwise the quantified interval must be included in it:

$$[\![\text{always } s]\!](t_0, t_1) = \forall t'_0, t'_1.[t'_0, t'_1] \subseteq [t_0, t_1], [\![s]\!](t'_0, t'_1)$$

Note that here we *do* use the input interval, resulting in a correct interpretation for phrases such as "In 1994, Itel was always on time." .

Aside: aspectual classes in the literature In this paper we borrow several notions from classical temporal semantics such as "stative", "achievement", "activity", etc., even though our definitions do not perfectly match the classical ones. We explain our precise meaning for these terms in the body of the paper. Nevertheless, we refer the reader to Steedman (2000) for an extensive review of formal temporal semantics.

For the *cognoscenti*, we can already point out some differences in terminology: we use the term activity as a general term which encompasses the three classical notions of activites, achievements and accomplishments. Indeed, insofar as the test suite is concerned, we find that these three categories can be collapsed into a single one (they are subject to Eq. (1)). That is, it is sufficient for the testsuite to distinguish between events and states. (In this paper, we always assume that the problems in the FraCaS testsuite are correctly annotated.)

Time references and aspectual classes A common theme in the testsuite is the reference to previous occurrences of an event:

(262) **P1** Smith left after Jones left.
P2 Jones left after Anderson left.
H Did Smith leave after Anderson left?

To be able to conclude that there is entailment, as the testsuite expects, we have to make sure that the two occurrences of "Jones left" (in **P1** and **P2**) refer to the same time intervals. For this purpose we postulate *unicity of action* for certain time-dependent propositions:

$$\mathsf{unicity}_P : P(t_1, t_2) \to P(t_3, t_4) \to (t_1 = t_3) \wedge (t_2 = t_4) \quad (1)$$

Unicity of action holds only if the aspectual class of the proposition P is *activity* (Steedman, 2000) (which, for our purposes, includes *achievements* and *accomplishments* as well).

(The difference between activity and accomplishments on the one hand and achievement on the other hand is that for the latter, time intervals can be assumed to be of nil duration. In reality, this is an oversimplification as achievements are usually of short duration, but not nil. However, this plays little role in our analysis. As far as we can tell the FraCaS test suite does exercise temporal semantics to such a level of precision.)

Unicity of action plays the role of event coreference in (neo-)Davidsonian accounts (Parsons,

1990). It is also a fine-grained principle, allowing coreference to take into account certain arguments when referencing. As we detail below, taking arguments into account yields is critical to handle repeatability of achievements.

Unicity of action appears to be a non-logical principle. Indeed, it is quite possible that "Jones left" several times. However, it seems that this principle is never contradicted by the testsuite. As such, even though unicity of action is only a pragmatic rule, it can be taken as a valid one *by default*: it is only when we have a sufficiently constrained situation that one should reject it. Consider the following discourse:

(1) Smith left at 1pm.
(2) Smith went to his appointment with the lawyer.
(3) Smith left at 4pm.

One would normally not say that there is contradiction. However if the middle sentence were not present, a contradiction should be flagged. We leave such discourse analysis as future work, and simply apply unicity of action everywhere: it is valid uniformly in the FraCaS test suite for activity aspect classes.

Statives *A contrario*, if P is *stative*, then we get a time-interval subsumption property:

$$\mathsf{subsumption}_P : [t_3, t_4] \subseteq [t_1, t_2] \rightarrow P(t_1, t_2) \rightarrow P(t_3, t_4)$$

This principle is used to reason about problem (314), below (note that "Smith" is used as a surname in the FraCaS and can take both feminine and masculine values):

(314) **P1** Smith arrived in Paris on the 5th of May, 1995.
P2 Today is the 15th of May, 1995.
P3 She is still in Paris.
H Smith was in Paris on the 7th of May, 1995.

Indeed, from **P3** we get that Smith was in Paris between May 5th and May 15th. Because "being in Paris" is stative, we also get that Smith was in Paris in any sub-interval. Contrary to unicity of action, subsumption is always valid.

Class-modifying adverbs It should be noted that some adverbs can locally disable the application of subsumption. For example, problem 299 features the sentence "Smith lived in Birmingham for exactly a year". Even though "live" is normally stative, one can no longer apply subsumption in the context of "exactly a year" — this can be done by propagating another context flag in the montagovian semantics (in addition to the temporal context).

(Un)repeatable Achievements The principle of using unicity of action interacts well with the usual interpretation of existential quantifiers (and anaphora). Indeed, using it, we can refute problem (279), as expected by the testsuite:

(279) **P1** Smith wrote a novel in 1991.
H Smith wrote it in 1992.

Indeed, following our account, the above (contradictory) inference problem is to be interpreted as

$$\begin{array}{l}\forall x.novel(x) \wedge \\ \exists t_1, t_2.[t_1, t_2] \subseteq 1991 \wedge write(smith, x, t_1, t_2) \wedge \\ \exists t_3, t_4.[t_3, t_4] \subseteq 1992 \wedge write(smith, x, t_3, t_4) \\ \longrightarrow \bot\end{array} \quad (2)$$

Note here that the scope for the existential is extended beyond the scope of **P1**, and its polarity switched (to universal). This extension can follow the account of Unger (2011), and our implemented analysis of anaphora(Bernardy et al., 2020; Bernardy and Chatzikyriakidis, 2019).

Thanks to the unicity of action of $write(smith, x, ...)$ (the subject and direct object are fixed) we find $[t_1, t_2] = [t_3, t_4]$, and due to the years 1991 and 1992 being disjoint we obtain contradiction. In sum, no special notion of accomplishment is needs to be invoked: we only need the principle of unicity of action.

Yet, the testsuite instructs that we should *not* be able to refute problem (280), with the justification that "wrote a novel" is a repeatable accomplishment:

(280) **P1** Smith wrote a novel in 1991.
H Smith wrote a novel in 1992.

Here our interpretation is:

$$(\exists x.novel(x) \wedge$$
$$\exists t_1, t_2.[t_1, t_2] \subseteq 1991 \wedge write(smith, x, t_1, t_2)) \wedge$$
$$(\exists y.novel(y) \wedge$$
$$\exists t_3, t_4.[t_3, t_4] \subseteq 1992 \wedge write(smith, y, t_3, t_4))$$
$$\longrightarrow \bot$$

Our analysis does not need to treat this last case specially. Indeed, even if $write(smith, x, ., .)$ is an activity and thus subject to unicity of action, in (280), x is quantified existentially; we have two *different* actions: $write(smith, x, t_1, t_2)$ and $write(smith, y, t_3, t_4)$, because $x \neq y$, and thus we cannot deduce equality of the intervals t_1, t_2 and t_3, t_4. In turn, the hypothesis cannot be refuted.

Action-modification Verbs The final class of lexemes carrying a temporal-dependent semantics are verbs taking a proposition as argument, like "finish", "start", etc. These verbs modify the temporal context in non-trivial ways. Consider for example "finish to ...". The timespan of the argument of "finish" should end within the timespan of the finishing action:

$$[\![\text{finish to } s]\!](t_0, t_1) = \exists (t'_0, t'_1).t'_1 \in [t_0, t_1] \wedge [\![s]\!](t'_0, t'_1)$$

Progressive Aspect We treat verbs in the progressive form as different semantically from the non-progressive form. For example, "John was writing a book" is encoded as $\exists (t_1, t_2).t_1 \leq t_2, t_2 \leq now, PROG\_write(John, book, t_1, t_2)$, while "John wrote a book" is encoded as $\exists (t_1, t_2).t_1 \leq t_2, t_2 \leq now, write(John, book, t_1, t_2)$. This distinction is necessary because in our analysis the progressive form ($PROG\_write$) is subject to subsumption. That is, if John is writing in the interval $[t_1, t_2]$ then he is writing in any sub-interval of $[t_1, t_2]$. This interpretation corresponds to the idea that the action takes place continuously over the whole interval. However, the same cannot be said of the non-continuous form ($write$): the end-points of the interval indicate the time needed to complete the achievement. (For example, "John wrote a book in 1993" neither entails "John wrote a book in January 1993" nor "John wrote a book in December 1993".) (In fact, *write*, in the non-progressive from, is on the contrary subject to unicity.) Finally, we also have $write(x, y, t_1, t_2) \rightarrow PROG\_write(x, y, t_1, t_2)$. That is, the achievement (or *activity* in our terminology) variant implies the stative variant, for the same interval. Consequently we get the entailment from "John wrote a book in 1993" to "John was writing a book in 1993", but not the other way around.

We note however that this interpretation differs only slightly from the usual accounts of the progressive in the literature. Ogihara (2007) summarises the position of Bennett and Partee (1978) as follows: a progressive sentence is true at an interval $[t_0, t_1]$ iff there is an interval $[t'_0, t'_1]$ such that $[t_0, t_1]$ is a non-final subinterval of $[t'_0, t'_1]$ and the progressive sentence is true at $[t'_0, t'_1]$. This is very similar to our approach (subsumption for the progressive form only), but there is a difference regarding final intervals. Yet in our view this difference is hard to justify: we cannot see why "John was writing a book in 1993" entails that he was writing it January, February, etc. but not in December.

Ogihara (2007) argues that this simple account of the progressive fails to reject a sentence such as "Lee is resembling Terri." while "Lee is walking" is acceptable. We argue instead that the latter should be rejected for pragmatic reasons. Indeed, when a predicate holds for a very long interval, one typically uses the simple present tense in English. Therefore the continuous form pragmatically implies that the predicate holds for a limited interval. But, without further context, the predicate "resemble Terri" does not vary over time (while "walk" generally does). Therefore the continuous form "Lee is resembling Terri" is confusing: one implies a limited interval, but the semantics of resembling normally yield an unlimited interval. Because we do not account for pragmatics, we prefer to retain the simplest account based on the subinterval property (which we call subsumption here).

Finally we stress that not all verbs are subject to the stative/achievement distinction induced by the progressive. For example, the phrases "John ran" and "John was running" appear to be logically equivalent, for entailment purposes.

4 Worked out example

To give a sense of the additional details necessary to deal with the precision demanded by a proof-assistant such as Coq we show how problem (279) is worked out in full details.

We start with input trees in GF format, given by Ljunglöf and Siverbo (2011). They can be rendered

as follows:

```
s_279_1_p=
sentence
(useCl past pPos
 (predVP
   (usePN (lexemePN "smith_PN"))
   (advVP
     (complSlash
      (slashV2a (lexemeV2 "write_V2")))
     (detCN (detQuant indefArt numSg)
       (useN (lexemeN "novel_N"))))
     (lexemeAdv "in_1991_Adv")))
s_279_3_h=
sentence
(useCl past pPos
 (predVP (usePN
   (lexemePN "smith_PN"))
   (advVP
     (complSlash
      (slashV2a (lexemeV2 "write_V2"))
      (usePron it_Pron))
     (lexemeAdv "in_1992_Adv"))))
```

Of particular note is the use of the pronoun "it", and the fact that adverbial expressions such that "in 1992" are lexicalized. We also follow the GF convention to postfix lexical items with the name of their category. Most of the other categories follow usual naming conventions. We remind the reader that "slash" categories are used to swap the order of arguments (compared to non-slashed categories of similar names).

Our dynamic and temporal semantics gives the following interpretation for `s_279_1_p` implies `s_279_3_h`.

```
FORALL (fun a=>novel_N a)
(fun a=>(exists (b: Time),
((exists (c: Time),
(IS_INTERVAL Date_19910101 b /\
 IS_INTERVAL c Date_19911231 /\
 IS_INTERVAL b c /\
 appTime b c (write_V2 a)
   (PN2object smith_PN))))) ->
Not (exists (f: Time),
((exists (g: Time),
 (IS_INTERVAL Date_19920101 f /\
  IS_INTERVAL g Date_19921231 /\
  IS_INTERVAL f g /\
  appTime f g (write_V2 a)
    (PN2object smith_PN)))))).
```

In the above, one should remark the top-level quantification over the novel (as explained in Section 3), the quantification over time intervals as individual timepoints, and the use of custom operators for several constructions (`FORALL`, `Not`, `IS_INTERVAL`, `appTime`). This use of custom operators is useful for several generalisations (for example, we have quantifiers such as `MOST` in addition to `FORALL` — see Bernardy and Chatzikyriakidis (2017))

Unfolding the definitions for these operators yield the following proposition:

```
forall x : object,
novel_N x ->
(exists b c : Z,
   Date_19910101 <= b /\
   c <= Date_19911231 /\
   b <= c /\ write_V2 x SMITH b c) ->
(exists f g : Z,
   Date_19920101 <= f /\
   g <= Date_19921231 /\
   f <= g /\ write_V2 x SMITH f g) ->
False
```

This is very close to our idealised representation of the problem Eq. (2). One difference is the use of abstract Coq integers for timepoints. Using a discrete time allows us to use predefined Coq tactics. The discrete nature of integers does not interfere with the reasoning.

Finally, we can show a Coq proof for the above proposition:

```
Theorem  problem279 : Problem279aFalse.
cbv.
intros novel isSmithsNovel P1 H.
destruct P1 as
   [t0 [t1 [ct1 [ct2 [ct3 P1]]]]].
destruct H as
   [u0 [u1 [cu1 [cu2 [cu3 H]]]]].
specialize writeUnique
  with (x := novel)(y := SMITH) as A.
unfold UniqueActivity in A.
specialize (A _ _ _ _ P1 H) as B.
lia.
Qed.
```

The intros and destruct tactics serve bookkeeping purposes. The critical part is the use of the `writeUnique` axiom, which witnesses the aspectual class of the predicate `write_V2`. The proof is completed by the use of the `lia` tactic, which is embeds a decision procedure for linear arithmetic problems[2]. Fortunately, `lia` can take care of all the problems which arise in the FraCaS testsuite.

5 Results and Evaluation

Our target is the FraCaS testsuite, which aims at covering a wide range of common natural-language phenomena. The suite is structured according to the semantic phenomena involved in the inference process for each example, and contains nine sections: Quantifiers, Plurals, Anaphora, Ellipsis, Adjectives, Comparatives, Temporal, Verbs and Attitudes. The system described here focuses on the Temporal section. However, it also supports the other eight sections. To our knowledge this is the first system which attempts to target the temporal section in full. But in fact, our system even provides support for all the other sections. Thus, a couple of decades

[2]It solves linear goals over rings by searching for linear refutations and cutting planes

Section	#FraCaS	This	FC2	FC	MINE	Nut	LP
Quantifiers	75	.93 74	.96 74	.96	.77	.53	.93 44
Plurals	33	.79	.82	.76	.67	.52	.73 24
Anaphora	28	.79	.86	-	-	-	-
Ellipsis	52	.81	.87	-	-	-	-
Adjectives	22	.95 20	.95 20	.95	.68	.32	.73 12
Comparatives	31	.65	.87	.56	.48	.45	-
Temporal	75	.73	-	-	-	-	-
Verbs	8	.75	.75	-	-	-	-
Attitudes	13	.85	.92	.85	.77	.46	.92 9
Total	337	.81 329	.89 259	.83 174	.69 174	.50 174	.85 89

Table 1: Accuracy of our system compared to others. "This" refers to the approach presented in this paper. When a system does not handle the nominal number of test cases (shown in the second column), the actual number of test cases attempted is shown below the accuracy figure, in smaller font. "FC" refers to the work of Bernardy and Chatzikyriakidis (2017), and "FC2" its followup (Bernardy and Chatzikyriakidis, 2019). "MINE" refers to the approach of Mineshima et al. (2015), "NUT" to the CCG system that utilises the first-order automated theorem prover *nutcracker* (Bos, 2008), and "LP" to the system presented by Abzianidze (2015). A dash indicates that no attempt was made for the section.

after its formulation, we propose a first attempt at covering the whole suite. As such, there it is no other system to compare our system with, in all aspects. We can however compare with systems which target parts of the FraCaS testsuite, as shown in Table 1.

Interaction with anaphora One reason explaining the lower performance of our system on some sections of the testsuite is that our interpretation of time interacts imperfectly with anaphora and ellipsis. Consider the following example:

(232) **P1** ITEL won more orders than APCOM did.
P2 APCOM won ten orders.
H ITEL won at least eleven orders.

In the first premise, our system essentially resolves the ellipsis to get the following reading: "ITEL won X orders and APCOM won Y orders and $X > Y$.". One would need each of the verb phrases "won X orders" and "won Y orders" to introduce their own timespans with existential quantifiers. However, the organisation of our system is such that the existentials are introduced before the ellipsis is expanded. Consequently we get a wrong interpretation and the inference cannot be made.

6 Conclusions and Future Work

We have presented a first attempt for a computational approach dealing with the temporal section of the FraCaS test suite. To do this, we have provided a simplified taxonomy of aspectual classes for verb phrases, guided by the applicability of the unicity of action and temporal subsumption properties. While part of this simplification is accidental (conflation of activity and accomplishment), we find that other parts (the automatic distinction between repeatable and unrepeatable achievements) constitute theoretical improvements.

Besides inference, formal interpretation of tense is found in natural-language interfaces to databases. Of note is the work of Androutsopoulos et al. (1998), which handles many of the time-aware adverbial clauses that we address. However, we cover many more logical aspects of inference, such as coreference via unity of action and interaction with quantifiers.

Bernardy and Chatzikyriakidis (2019) presented a logical system for handling 8 of the 9 sections of the FraCaS test suite, but excluded section 7, suggesting that it requires many examples that need an *ad hoc* treatment. Here, we took up this challenge and have shown that a system similar to theirs can be extended to cover the remaining section of the test suite, without considerably decreasing the performance of the rest of the sections. This is indeed a common problem with logical approaches, namely the fact that one can have theoretically motivated implementations of individual phenomena, e.g. anaphora, ellipsis, quantifiers, temporal reference etc., but when one tries to put all these together into a unified system, this proves to be a daunting task. We believe that this paper presents an exception, and provides a system that can deal with all these different semantic phenomena under a unified system with very good results. We use the same combination of a number of well-studied tools as Bernardy and Chatzikyriakidis (2019) : type theory, parsing using the Grammatical Framework (GF), Monadic Dynamic Semantics and proof assistant technology (Coq). The system achieves an accuracy of 0.73 on the Temporal Section and 0.81 overall. The whole system, including data sets, is available at the following url: `https://github.com/GU-CLASP/FraCoq/tree/iwcs2021`.

One of the things to be looked at is fixing the issues associated with parts of the test suite that "broke" when the temporal analysis was introduced. Some of these have been already mentioned: interaction of the temporal variables with anaphora.

Another extension of this work is to reflect more temporal semantic inference properties in an extended test suite. Indeed, there as properties which are not captured in the FraCaS test suite, such as fine-grained examples of lexical and grammatical aspect, as well as the interaction between those two, for example cases where one needs to actually distinguish between achievements and accomplishments on the basis of their inferential properties:

(∗1) **P1** John found his keys.
H John was finding his keys (UNK).

(∗2) **P1** John wrote a book.
H John was writing a book (YES).

In the first of the two examples involving an achievement verb, the inference is UNK, since there is no guarantee that the action is non-instantaneous. To the contrary, for accomplishment verbs, the inference follows.

Further cases to be included in an extended FraCaS future suite involve examples where the interaction between different tenses needs to be captured:[3]

(∗3) **P1** When the phone rang, John had entered the house.
H John entered the house before the phone rang (YES).

Finally it would be desirable to improve automation of the system, and evaluate it on a larger test set. As it stands Coq fully *checks* the proof of entailment for each (provable) problem. However, the construction of such proofs has demanded human intervention. It would be desirable to fully automate the proof construction step. For this to make sense however we need a much larger test suite, properly separated into a development and a (secret) test set. Otherwise, only the limited power of the logic prevents us (or any followup work) to fine-tune the rules of the system until one gets full coverage. This kind of observation holds in general of any rule-based system, and thus applies not only to the proof-construction phase, but also to the underlying dynamic semantics and parsing phase (which is limited only by the power of the language and frameworks used for its implementation). In sum, contrary to statistical approaches to language understanding, the value of the present work lies not in the bare accuracy numbers which we are able to achieve, but in the details of *how* we do so: the of set of rules which we use, which is described in detail here and in the work which we base ourselves upon (Bernardy et al., 2020; Bernardy and Chatzikyriakidis, 2019).

[3]While this work was completed, the work by (Vashishtha et al., 2020) was published. The authors present a five datasets to be used for the training of neural models' ability to capture temporal reasoning. It would be interesting to check the amount of data covered, most specifiaclly the level of fine-grainedness of temporal reasoning needed to capture those examples, as compared to what we have been discussing in this paper. We thank an anonymous reviewer for bringing this work to our attention.

Acknowledgements

The research reported in this paper was supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg. We are grateful to our colleagues in CLASP for helpful discussion of some of the ideas presented here. We also thank anonymous reviewers for their useful comments on an earlier draft of the paper.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of EMNLP15*.

Ion Androutsopoulos, Graeme D Ritchie, and Peter Thanisch. 1998. Time, tense and aspect in natural language database interfaces. *arXiv preprint cmp-lg/9803002*.

Michael Bennett and Barbara Hall Partee. 1978. *Toward the logic of tense and aspect in English*, volume 84. Indiana University Linguistics Club Bloomington.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2017. A type-theoretical system for the fracas test suite: Grammatical framework meets coq. In *IWCS 2017-12th International Conference on Computational Semantics-Long papers*.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2019. A wide-coverage symbolic natural language inference system. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. ACL.

Jean-Philippe Bernardy, Stergios Chatzikyriakidis, and Aleksandre Maskharashvili. 2020. A computational

treatment of anaphora and its algorithmic implementation: Extended version. Available on the first author's homepage: https://jyp.github.io/pdf/phoroi.pdf or online https://bit.ly/2xQ4G2M.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In *Semantics in text processing. Step 2008 conference proceedings*, pages 277–286.

Simon Charlow. 2015. Monadic dynamic semantics for anaphora. *Ohio State Dynamic Semantics Workshop*.

Simon Charlow. 2017. A modular theory of pronouns and binding. In *Logic and Engineering of Natural Language Semantics (LENLS) 14*. Springer.

Stergios Chatzikyriakidis and Zhaohui Luo. 2014. Natural language inference in coq. *Journal of Logic, Language and Information*, 23(4):441–480.

R. Cooper, D. Crouch, J. van Eijck, C. Fox, J. van Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, and S. Pulman. 1996. Using the framework. Technical report LRE 62-051r, The FraCaS consortium. ftp://ftp.cogsci.ed.ac.uk/pub/FRACAS/del16.ps.gz.

David R Dowty. 2012. *Word meaning and Montague grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ*, volume 7. Springer Science & Business Media.

Tim Fernando. 2015. The semantics of tense and aspect. *The Handbook of Contemporary Semantic Theory*, pages 203–236.

James Higginbotham. 2009. *Tense, aspect, and indexicality*, volume 26. OUP Oxford.

P. Ljunglöf and M. Siverbo. 2011. A bilingual treebank for the FraCas test suite. Clt project report, University of Gothenburg.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200. Association for Computational Linguistics.

Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of EMNLP*.

Richard Montague. 1970. English as a formal language. In *Linguaggi nella Societa e nella Tecnica*.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In *Approaches to natural language*, pages 221–242. Springer.

Richard Montague. 1974. The proper treatment of quantification in ordinary english. In Richmond Thomason, editor, *Formal Philosophy*. Yale UP, New Haven.

Toshiyuki Ogihara. 2007. Tense and aspect in truth-conditional semantics. *Lingua*, 117(2):392–418.

Terence Parsons. 1990. *Events in the Semantics of English*, volume 5. MIT press Cambridge, MA.

Arthur N Prior and Per FV Hasle. 2003. *Papers on time and tense*. Oxford University Press on Demand.

Aarne Ranta. 2004. Grammatical framework. *Journal of Functional Programming*, 14(2):145–189.

Mark Steedman. 2000. The productions of time. *Draft. Available at http://www. cogsci. ed. ac. uk/steedman/papers. html*.

Christina Unger. 2011. Dynamic semantics as monadic computation. In *JSAI International Symposium on Artificial Intelligence*, pages 68–81. Springer.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Benjamin Werner. 1994. *Une théorie des constructions inductives*. PhD thesis, Université de Paris 7.

CO-NNECT: A Framework for Revealing Commonsense Knowledge Paths as Explicitations of Implicit Knowledge in Texts

Maria Becker, Katharina Korfhage, Debjit Paul, Anette Frank
Department of Computational Linguistics, Heidelberg University
mbecker|korfhage|paul|frank@cl.uni-heidelberg.de

Abstract

In this work we leverage commonsense knowledge in the form of knowledge paths to establish connections between sentences, as a form of *explicitation of implicit knowledge*. Such connections can be direct (singlehop paths) or require intermediate concepts (multihop paths). To construct such paths we combine two model types in a joint framework we call CO-NNECT: a *relation classifier* that predicts direct connections between concepts; and a *target prediction model* that generates target or intermediate concepts given a source concept and a relation, which we use to construct multihop paths. Unlike prior work that relies exclusively on *static* knowledge sources, we leverage language models finetuned on knowledge stored in ConceptNet, to *dynamically* generate knowledge paths, as explanations of implicit knowledge that connects sentences in texts. As a central contribution we design manual and automatic evaluation settings for assessing the *quality* of the generated paths. We conduct evaluations on two argumentative datasets and show that a combination of the two model types generates meaningful, high-quality knowledge paths between sentences that reveal implicit knowledge conveyed in text.

1 Introduction

Commonsense knowledge covers simple facts about the world, people and everyday life, e.g., *Birds can fly* or *Cars are used for driving*. It is increasingly used for many NLP tasks, e.g. for question answering (Mihaylov et al., 2018), textual entailment (Weissenborn et al., 2018), or classifying argumentative functions (Paul et al., 2020). In this work, we leverage commonsense knowledge in the form of single- *and* multihop knowledge paths for establishing connections between concepts from different sentences in texts, and show that these paths can *explicate* implicit information conveyed by the text. Connections can either be direct, e.g. given the sentences *The car was too old* and *The engine broke down*, the concepts *car* and *engine* can be linked with a direct relation (singlehop path) *car* → HASA → *engine*; or indirect – here intermediate concepts are required to establish the link, as between *Berliners produce too much waste* and *Environmental protection should play a more important role*, where the link between *waste* and *environmental protection* requires a multihop reasoning path: *waste* → RECEIVESACTION → *recycle* → PARTOF → *environmental protection*.

We show that two complementary model types can be combined to solve the two subtasks: (i) for predicting singlehop paths between concepts, we propose a *relation classification* model that is very precise, but restricted to direct connections between concepts; (ii) for constructing longer paths we rely on a *target prediction* model that can generate intermediate concepts and is thus able to generate multihop paths. However, the intermediate concepts can be irrelevant or misleading. To our knowledge, prior work has applied *either* relation classification *or* target prediction models. We propose CO-NNECT, a framework that establishes **CO**mmonsense knowledge paths for **CONNECT**ing sentences by *combining* relation classification and target prediction models, leveraging their strengths and minimizing their weaknesses. With CO-NNECT, we obtain *high-quality knowledge paths* that explicate implicit knowledge conveyed by the text.

We focus on commonsense knowledge in ConceptNet (Speer et al., 2017), a knowledge graph (KG) that represents concepts (words or phrases) as nodes, and relations between them as edges, e.g., ⟨*oven*,USEDFOR, *baking*⟩. As instances of the model types we use COREC (Becker et al., 2019), a multi-label relation classifier that predicts *relation*

Proceedings of the 14th International Conference on Computational Semantics, pages 21–32
June 17–18, 2021.

types and that we enhance with a pretrained language model; and COMET (Bosselut et al., 2019), a pretrained transformer model that learns to generate *target concepts* given a source concept and a relation. In contrast to models that retrieve knowledge from static KGs (Mihaylov et al., 2018; Lin et al., 2019), both models are fine-tuned on ConceptNet and applied *on the fly*, to dynamically generate knowledge paths that generalize beyond the static knowledge, allowing us to predict unseen knowledge paths. We compare our models to a baseline model that solely relies on static KGs.

We evaluate our framework on two English argumentative datasets, IKAT (Becker et al., 2020) and ARC (Habernal et al., 2018), which offer annotations that explain implicit connections between sentences. While knowledge paths have been widely used in NLP downstream tasks, a careful evaluation of these paths has not received much attention. As a central contribution of our work, we address this shortcoming by designing manual and automatic settings for path evaluation: we evaluate the relevance and quality of the paths and their ability to represent implicit knowledge in an annotation experiment; and we compare the paths to the annotations of implicit knowledge in IKAT and ARC, using automatic similarity metrics.

Our main contributions are: i) we propose CONNECT, a framework that combines two complementary types of knowledge path prediction models that have previously only been applied separately;[1] ii) we show that commonsense knowledge paths generated with CONNECT effectively represent implied knowledge between sentences; iii) we propose an evaluation scheme that measures the quality of the knowledge paths, going beyond many approaches that use knowledge paths for downstream applications without analyzing their quality.

2 Related Work

In this work we combine relation classification and target prediction for generating commonsense knowledge representations over text. **Relation classification** covers a range of methods and learning paradigms for representing relations. A variety of neural architectures such as RNNs (Zhang et al., 2018), CNNs (Guo et al., 2019), sequence-to-sequence models (Trisedya et al., 2019) or language models (Wu and He, 2019) achieved state-of-the-art results. To our knowledge, Becker et al. (2019) is the only work that proposed a relation classification model specifically for ConceptNet relations, which we adapt for our work. Besides COMET (Bosselut et al., 2019), the model used in our approach, Saito et al. (2018) perform **target prediction** on ConceptNet using an attentional encoder-decoder model. They improve the KB completion model of Li et al. (2016) by jointly scoring triples and predicting target concepts.

Utilizing commonsense knowledge paths. When using commonsense knowledge for question answering (Mihaylov et al., 2018), commonsense reasoning (Lin et al., 2019) or NLI (Kapanipathi et al., 2020), most approaches rely on paths retrieved from *static* knowledge resources. In contrast, we propose a framework that in addition makes use of *dynamic* knowledge provided by language models. Few other models have used knowledge paths **dynamically**, e.g. Paul et al. (2020), who enrich ConceptNet on the fly when classifying argumentative functions.
Wang et al. (2020) make use of **language models** for question answering. They generate multi-hop paths by sampling random walks from ConceptNet and finetune a language model on these paths to connect question and answers, improving accuracy on two question answering benchmarks. Bosselut et al. (2021) generate knowledge paths using a language model for zero-shot question answering, which they use to select the answer to a question, surpassing performance of pretrained language models on SocialIQA (a multiple-choice question answering dataset for probing machine's emotional and social intelligence in a variety of everyday situations). Similarly, Chang et al. (2020) incorporate knowledge from ConceptNet in pretrained language models for SocialIQA. They extract keywords from question and answers, query ConceptNet for relevant triples, and incorporate them in their language models via attention. Their evaluation shows that their knowledge-enhanced model outperforms knowledge-agnostic baselines. Finally, Paul and Frank (2020) propose an attention model that encodes commonsense inference rules and incorporates them in a transformer based reasoning cell, taking advantage of pretrained language models and structured knowledge. Their evaluation on two reasoning tasks shows that their model improves performance over models that lack external knowledge. Hence, none of these sys-

[1] The code for our framework can be found here: https://github.com/Heidelberg-NLP/CO-NNECT.

tems *directly* evaluates the **quality** of the generated paths, but measure the effectiveness of commonsense knowledge *indirectly* by evaluation on downstream tasks. We will address this shortcoming in our work by carefully evaluating the quality of the generated paths.

3 Enriching Texts with Commonsense Knowledge Paths

This section describes CO-NNECT, the framework we propose for enriching texts with commonsense knowledge, by establishing relations or paths between concepts from different sentences. Towards this aim, we apply relation classification and target prediction models in combination. We first characterize differences between the two model types and their instantiations, COREC-LM and COMET, describe how we adapt them to our task and evaluate them on ConceptNet to assess their performance (§3.1). We then show how we utilize the models to establish connections between concepts in texts (§3.2) and present a baseline model that uses ConceptNet as a static KG to establish commonsense knowledge paths (§3.3).

3.1 Comparing and Evaluating Model Types

Relation classification and target prediction both aim at representing relational commonsense knowledge, but the respective task settings are fundamentally different. We choose two models that have been developed for representing commonsense knowledge in CN: COREC, a relation classification and COMET, a target prediction model.

Relation Classification with COREC-LM. A relation classifier is ideally suited to predict *direct* relations between concepts, hence we can apply COREC (Becker et al., 2019), an open-world multi-label relation classification system, for this task. Given a pair of concepts c_s, c_t from sentences, it predicts one or several relations r_i from a set of relation types R_{CN} that hold between c_s and c_t. We enhance the original neural model with the pretrained language model DistilBERT (Sanh et al., 2019) to construct a classifier we call COREC-LM. We finetune this model on ConceptNet by masking the relations and use sigmoid as output layer to model the probability of each relation independently, accounting for ambiguous relations in CN.

Target Prediction with COMET. To generate multihop paths that include (possibly novel) intermediate concepts, we apply COMET (Bosselut et al., 2019), a transformer encoder-decoder based on GPT-2 (Radford et al., 2019). Input to the model is a source concept c_s and a relation r_i. Then the pretrained language model is finetuned using ConceptNet as labelled train set for the task of generating new concepts. Depending on the beam size, COMET can propose multiple targets per input instance.

Datasets. To compare model performances, we evaluate COREC-LM and COMET on the **CN-100k** benchmark dataset of Li et al. (2016), which is based on the OMCS subpart of ConceptNet. The dataset comprises 37 relation types such as ISA, PARTOF or CAUSES and contains 100k relation triples in the train set and 1200 in the development and the test set, respectively. CN-100k contains a lot of infrequent relations which are hard to learn and often overspecific (e.g. HASFIRSTSUBEVENT), and hence not useful for establishing high quality relations and paths between concepts. We therefore extract a subset that contains all triples of the 13 most frequent relations (**CN-13**).[2] CN-13 covers 90,600 triples for training, 1080 triples for development, and 1080 triples for testing.

Since our application task requires that the relation classifier also learns to detect that a given concept pair is *not* related, we extend the data for training and testing COREC-LM with a RANDOM class that contains unrelated concept pairs, which we add to CN-100k and CN-13.[3]

PoS Sequence Filtering. We apply a type-based PoS sequence filtering for COREC-LM and COMET, where the type is dependent on the predicted relation. The relation ISA, for example, is supposed to connect two noun phrases; in contrast, HASPREREQUISITE typically relates two verb phrases. We determine frequent PoS sequence patterns for specific argument types from the ConceptNet resource and use them to filter relation and path predictions.

Metrics. We evaluate COREC-LM in terms of weighted F1-scores, precision and recall, which is its genuine evaluation setting. For COMET we report precision scores for the first prediction with highest confidence score (hits@1); we further report hits@10 which gives information if the correct

[2] These are: ATLOCATION, CAUSES, CAPABLEOF, ISA, HASPREREQUISITE, HASPROPERTY, HASSUBEVENT, USEDFOR, CAUSESDESIRE, DESIRES, HASA, MOTIVATEDBYGOAL and RECEIVESACTION.

[3] For details about the construction of the RANDOM class, cf. Appendix.

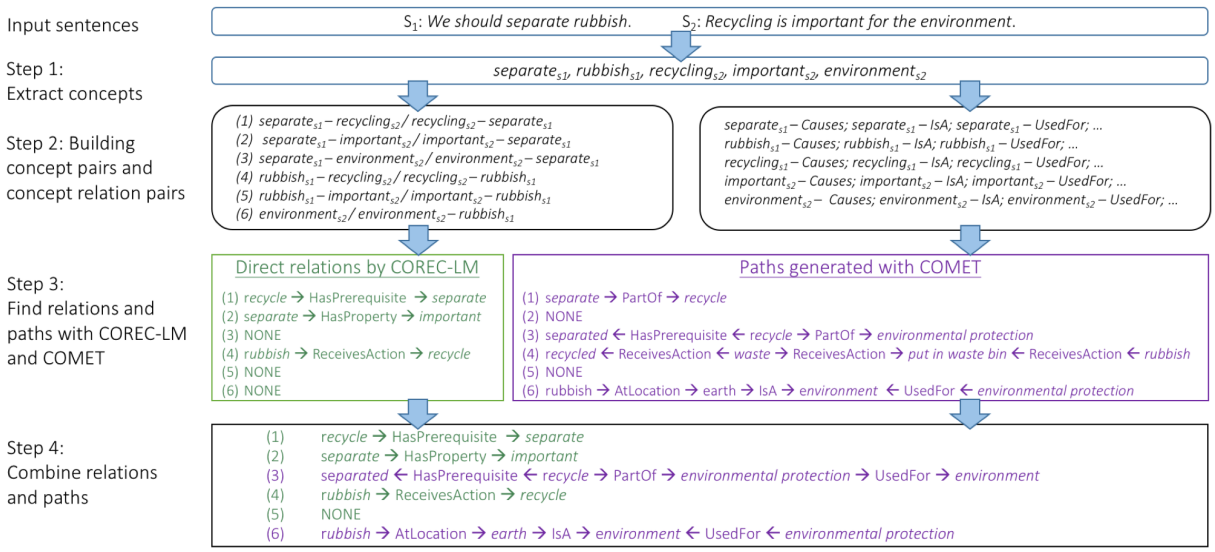


Figure 1: Our framework CO-NNECT: It finds single- and multihop paths between concepts, as explicitations of implicit knowledge that connects sentences.

triple is included in the first ten predictions (which will be important since we later use a beamsize of 10 for generating paths). In addition, we report accuracy using the Bilinear AVG model of Li et al. (2016) (COMET's genuine evaluation setting), which is trained on CN-100k and produces a probability for a generated relation triple to be correct. Following Bosselut et al. (2019), we apply a beamsize of 1 and a threshold at 0.5 for judging a triple as correct.

Model Performances. COREC-LM achieves high F1-scores on CN-100k (76.5) and CN-13 (86.0).[4] Scores are significantly lower when adding the RANDOM class (-7pp on CN-100k&CN-13), indicating that detecting unrelated concept pairs is not trivial. The results show that a strength of COREC-LM is its precision (90.1/CN-100k; 88.2/CN-13) – which we will leverage when combining models. **COMET** achieves high accuracy scores (92.3/CN-100k; 96.3/CN-13) according to the bi-score. For the much stricter metric hits@1 which judges a triple as correct only if it matches the respective triple in the testset, much lower scores are achieved (25/CN-100k; 23.5/CN-13), which is evident given the wide range of possible target generations. Higher scores for hits@10 (65.3/CN-100k; 65.9/CN-13) show that the chance for correct predictions significantly rises with increasing beam size.

In sum, COREC-LM and COMET both aim at learning commonsense knowledge representations, but tackle different tasks and have different strengths and weaknesses. COREC-LM is very precise in its predictions, but is restricted to predicting *direct* relations between two given concepts. COMET is more powerful since it can genuinely generate *novel* target concepts and thus can generate multihop paths. However, it tends to be more imprecise, and bears the risk of generating *irrelevant or noisy* concepts. Hence, a combination of models seems beneficial, to predict high-quality single- *and* multihop paths between concepts.

3.2 Establishing Connections Using Relation Classification and Target Prediction

In the following we describe how we combine and apply COREC-LM and COMET in a joint framework, CO-NNECT, to establish high-quality knowledge paths between sentences. An overview is given in Fig.1. In the first step we **extract relevant concepts** from the text. For this we integrate the concept extraction tool CO-CO-EX (Becker et al., 2021a), which extracts meaningful concepts from texts and maps them to concept nodes in CN, considering all surface forms.

Linking Concepts with *Direct Relations*. We construct all possible pairs of concepts extracted from S_1 and S_2 by taking the cross product $c_s \times c_t$, where c_s is a concept from S_1, and c_t a concept from S_2 (Fig. 1, Step 2, left). We then apply COREC-LM trained on CN-13+RANDOM with a tuned threshold of 0.9 for predicting which rela-

[4]The original version of COREC (Becker et al., 2019) achieves F1 of 53.31/CN-100k; 72.33/CN-13.

tion $r_i \in R_{CN13}$ holds between the concept pairs, or whether no relation holds (RANDOM) (cf. Fig. 1, Step 3 (left) for examples).

Linking Concepts with *Multihop Paths.* COMET requires as input a source concept and a relation. For each concept pair c_s, c_t we build such inputs by combining c_s with each relation $r_i \in R_{CN13}$, yielding 13 pairs c_s, r_i which we input for target prediction (Step 2, right). To discover relation chains starting from S_2, we apply the same process to c_t, using c_s as target concept. We also include *inverse relations*, which gives us greater flexibility for connecting entities, i.e., paths are allowed to contain inverted triplets (e.g. *baking* ← USEDFOR ← *oven* → ATLOCATION → *kitchen*). To this end, we switch the order of concept pairs within a given relation r_i, relabel the relation as r_i^{-1}, and add the inverted relation pair to COMET's training set.

Forward Chaining. For all pairs in the cross-product $c_s \times c_t$, for each input c_s, r_i and c_s, r_i^{-1} we generate the 10 most confident concepts c_{t_i} with COMET (beamsize 10) trained on CN-13 including inverse triples. We continue with all paths where the generated concept c_{t_i} has minimum cosine distance of 0.7 to the respective target concept c_t. We generate the next hop by using each c_{t_i} as a new source concept, combine it with each of the 13 original and inverse relations, generate novel target predictions, and again compare to the target concept. This similarity comparison guides the forward chaining process towards the chosen target concepts and helps detecting contextually relevant paths. We use ConceptNet numberbatch embeddings for the encoding of concepts; for multiword concepts we average the embeddings of all non-stopwords.

Terminating Paths. We terminate path generation as soon as the similarity between c_{t_i} and c_t is higher than 0.95 – here we expect the two concepts to express the same meaning. We restrict the path length to 3 hops and consider only completed paths for evaluation (Step 3, right in Fig. 1).

Combining Approaches. With our framework CO-NNECT we leverage the potential of the complementarity of the two model types by combining COREC-LM and COMET in a straightforward way. Our hypothesis is that a system that admits both single and multihop connections for establishing links between concepts offers the greatest flexibility. We further hypothesize that direct relations should be preferred over indirect multihop paths, since the latter could include irrelevant or misleading intermediate nodes. Thus, we discard all multihop paths for each concept pair for which COREC-LM predicted a direct connection (Fig. 1, Step 4, pair 4). If COMET *and* COREC-LM produce a singlehop path, we also prefer COREC-LM's prediction, relying on the model's high precision (pair 1 in Step 4). We keep the paths generated by COMET for concept pairs for which no direct relation could be established (i.e., COREC-LM predicted RANDOM or no prediction above its threshold, pair 3&6), assuming that in such cases intermediate concepts are required to establish a link. If only one of the models establishes a link, we keep this connection (pair 2), and if none of the models finds a link, we assume that the concepts are not (closely) connected (pair 5).

3.3 Static Baseline Model

We compare COREC-LM and COMET against the model of Paul and Frank (2019) that uses ConceptNet as a static KG. The system extracts paths between pairs of concepts from sentence pairs, hence conforms well to our setting. Following Paul and Frank (2019), starting from concepts in a sentence pair (§3.2), we construct a subgraph $G' = (V', E')$ of the ConceptNet graph, where V' comprises all concepts c_i in $\langle S_1, S_2 \rangle$. The system then finds all shortest paths p' from ConceptNet that connect any concept pairs in V', and includes them in G'. It then includes, for any concepts in G', all directly connected concepts from ConceptNet together with their edges. This yields a small sub-graph from ConceptNet that contains concepts and relations relevant for capturing conceptual links across the sentence pair. To select *relevant* paths, G' is filtered by computing scores for vertices and paths using PageRank and Closeness Centrality score, and we constrain path lengths to 3 hops.

4 Revealing Implicit Knowledge through Knowledge Paths: Experiments and Evaluation

In this section we evaluate the paths generated by our proposed models. We first present our datasets and statistics on established connections (§4.1), and then evaluate the quality of the paths manually (§4.2) and automatically (§4.3).

Sentence 1	Sentence 2	COREC-LM	COMET	Ranked Subgraphs	Gold Implicit Information
The morning-after **pill** ought to be available over-the-counter in pharmacies.	Even though with **contraceptives there** can be mistakes and problems.	*pill* IsA *contraceptive*	*pill* IsA *contraceptive*	*pill* IsA *contraceptive*	The morning-after pill is a contraceptive.
I as an employee find it very practical to be able to **shop** at least on weekends.	Plus, the state wants me to **spend** my money.	NONE	*spend* HasSubevent *go to mall* HasSubevent *shop*	*spend* DerivedFrom *spend money* RelatedTo *shop*	When someone is shopping, he is spending money.
Today I will delete my **Facebook** account.	I'm tired of my **data** boosting the market value of that corporation.	*Facebook* HasA *data*	NONE	*Facebook* RelatedTo *book* RelatedTo *data*	With a facebook account you make your data available.
Felons should be allowed to vote.	A person who **stole** a car at 17 should bot be barred from being a full citizen for life.	NONE	*felon* CapableOf *steal*	NONE	Grand theft auto is a felony.
Comment sections have not failed.	With good moderation, useful and interesting **discussions** ca be has.	*comment* Causes *discussion*	*comment* AtLocation *debate* IsA *discussion*	NONE	Moderation is usually good.

Figure 2: Example generations from our three model types (first three instances from IKAT, last two from ARC).

4.1 Datasets and Statistics

The **IKAT** dataset (Becker et al., 2020) is based on the English Microtexts Corpus of short argumentative texts (Peldszus and Stede, 2016). For all sentence pairs that are adjacent or argumentatively related, annotators added the implicit knowledge that connects them, using short sentences. IKAT contains 719 such sentence pairs, from which we extracted 60,294 concept pairs. The **ARC** dataset (Habernal et al., 2018) contains arguments taken from online discussions in English, consisting of a claim and a premise, and an annotated implicit warrant that explains why the claim follows from the premise. We evaluate our models on the ARC test set that comprises 444 argument pairs, from which we extracted 21,898 concept pairs; and the corresponding warrants.

Example generations for both datasets from our three model types – COREC-LM, COREC, and ranked CN-graphs – appear in Fig. 2, where the first three sentence pairs come from IKAT, and the last two from ARC.

Number of links and hops. Table 1 gives statistics of the paths generated between concepts from sentence pairs from IKAT and ARC using our different models. We find that COREC-LM finds relations between around 22k from 66k concept pairs in **IKAT**, while COMET only generates paths between 3,660 pairs. This can be explained by the very high similarity threshold we imposed for guiding the forward chaining process towards the target concept, since our motivation was not to generate as *many* paths as possible, but paths that are *meaningful* and contextually *relevant*. When combining paths from COMET and COREC-LM, we find links for more than 24k concept pairs in IKAT. The highest number of links is established by ranking CN-subgraphs (50k linked concept pairs). For **ARC**, which contains 22k concept pairs, COREC-LM finds links between around 10k and COMET around 2k concept pairs, while almost 15k pairs can be connected using ranked CN-graphs. In both datasets, the ranked CN-graphs contain on average 2.1 **hops** (relations) per path, while the paths generated by COMET are shorter (1.4 on IKAT/1.5 on ARC). In fact, COMET establishes many direct relations (69% of all paths are single hops), whereas the ranked CN-graphs are mostly two- (49%) or three-hop paths (37%).

Replacing Vague Relations in CN-Graphs. We find that in contrast to COREC-LM and COMET, the ranked CN-graphs are constructed using mostly the very general relation RELATEDTO (71%/IKAT; 72%/ARC), followed by the likewise vague relation HASCONTEXT (8% in both datasets).[5] For determining the impact of vague relations on path quality, we replace all RELATEDTO and HASCONTEXT relations in the ranked CN-graphs with relations predicted by COREC-LM (trained on CN-13, threshold 0.9). For IKAT, we replace 43.4% of all RELATEDTO and 46.2% of all HASCONTEXT instances, in ARC we replace 70.7% of all RELATEDTO and 37% of all HASCONTEXT relations. We use this version when evaluating paths, in addition to the original ranked CN-graphs.

4.2 Manual Evaluation of Path Quality

Our statistics showed that most links between concepts can be revealed using knowledge paths retrieved from ConceptNet as a static KG, whereby these paths tend to contain multiple hops and a high amount of vague relations. Fewer links are established using the dynamic models COREC-LM and COMET, which produce shorter paths using

[5]For details on relation distributions cf. Appendix.

		COR	COM	CONN	CN
IKAT	linked pairs	21,934	3,660	24,063	50,003
	avg. hops	1	1.4	1.1	2.1
ARC	linked pairs	9,844	1,826	10,828	14,940
	avg. hops	1	1.5	1.1	2.1

Table 1: Statistics of paths generated by COREC-LM, COMET, their combination (CO-NNECT), and ranking CN-graphs (CN): number of concepts pairs between which a link was found, and average number of hops per path.

		IKAT				ARC			
		COR	COM	CN	CN-r	COR	COM	CN	CN-r
Predictions		74	64	**88**	**88**	**78**	60	76	76
Relev.	+2	**70**	50	36	40	**63**	49	30	34
	+1	19	**27**	22	24	25	28	28	**32**
	0	8	18	27	21	8	9	29	22
	-1	3	5	10	10	2	6	4	3
	-2	0	0	5	5	2	8	9	9
Impl.	yes	**80**	78	57	67	**87**	81	57	62
Knowl.	no	20	22	43	33	13	19	43	38
Best Link		**65**	64	28	34	**76**	70	7	14

Table 2: Manual evaluation of paths from COREC-LM, COMET, ranked CN-graphs (CN), and CN-graphs with replaced vague relations (CN-r); all numbers in %.: How many concept pairs could be linked (line 1), are the links relevant and meaningful (2-6), do the links represent implicit knowledge (7-8), how often a link was chosen to be most helpful for understanding the connection (9).

only specific relation types from CN-13. Since our aim is to construct high-quality, meaningful knowledge paths that help to explain implicit information (rather than establishing as many links as possible), we now examine the quality and relevance of the knowledge paths. We set up an annotation experiment, providing annotators with 100 sentence pairs from each dataset, with marked concepts (one from S_1 and one from S_2) and the path generated between these concepts by (i) COREC-LM, (ii) COMET, (iii) ranked paths from CN, and (iv) ranked paths with replaced vague relations (CN-r).

Annotation Setup. For each sentence pair, our annotators evaluated if 1) the path is a meaningful and relevant explanation for the connection between the two sentences (very relevant/relevant/neutral/not relevant/misleading); if 2) the path represents implicit information not explicitly expressed in the sentences (yes/no); and 3) which model generates the path that is most helpful and expressive for understanding the connection between the sentences. 4) To evaluate the *combination* of COREC-LM and COMET in CO-NNECT, we generate a subset for each dataset that includes all sentence/concept pairs for which COREC-LM predicted a singlehop path *and* COMET generated a multihop path (10 pairs per subset). For these instances we ask in addition whether the multihop paths include unrelated, unnecessary or uninformative intermediate nodes (yes/no), misleading intermediate nodes (yes/no); or intermediate nodes that are important for explaining the connection and missing in the direct relation predicted by COREC-LM (yes/no).[6] Annotations were performed by two annotators with a linguistic background. We measure IAA using Cohen's Kappa and achieve an agreement of 81%. Remaining conflicts were resolved by an expert annotator.

Results. Table 2 shows the results of our annotation experiment. On **IKAT**, 89% of the paths established by COREC-LM and 77% of the relations predicted by COMET were annotated as very relevant (+2) or relevant connections (+1), which only applies for 58% of the ranked CN-paths. 15% of the ranked CN-paths were annotated as not relevant (-1) or misleading (-2), which can be explained by noisy intermediate nodes; and 27% as vague (0), which can be explained by the large amount of unspecific relations. When replacing RELATEDTO and HASCONTEXT (CN-r), the amount of paths annotated as vague slightly decreases, and the amount of paths labelled as relevant and very relevant increases.

Moreover, paths generated by COREC-LM and COMET were found to yield better implicit knowledge representations than ranked CN-paths (line 8-9, Table 2), while we find that replacing vague relations in the CN-paths improves their ability of representing implicit knowledge. Finally, 65% of relations predicted by COREC-LM and 64% of paths generated by COMET were chosen as explaining the connections between sentences best, which is only the case for 28% of the CN-paths, and slightly better for the replaced version of the CN-paths (34%).

On **ARC**, the high amount of CN-paths annotated as vague (29%) again indicates uninformative connections and can be reduced when replacing vague with more specific relations. Relations predicted by COREC-LM were found to be less relevant for connecting sentences in ARC than in IKAT, but 87% of them were evaluated as appropriate expressions of implicit knowledge. 76% of the

[6]The annotation manual together with example annotations can be found here: https://github.com/Heidelberg-NLP/CO-NNECT/blob/main/manual.pdf

relations predicted by COREC-LM were evaluated as best connections, which applies only for 7% of CN-paths and 14% of CN-paths with replaced relations. For COMET we find overall comparable results between IKAT and ARC.

Regarding the **combination** of COREC-LM and COMET addressed with question 4, according to our annotators 50% of the multihop paths in the IKAT subset include misleading nodes and *all* of them include irrelevant or uninformative nodes. Still, compared against the direct relations predicted by COREC-LM, annotators state for 30% of the multihop paths from COMET that they contain intermediate concepts that are important for explaining the connection. On the ARC subset, 40% of the multihop paths include misleading and 60% include irrelevant nodes, and only 20% contain important intermediate concepts that are missing in the direct relation. For each subset, annotators preferred the shorter path over the multihop path in 90% of the given sentence pairs. Comparing singlehop paths generated by COMET to direct relations predicted by COREC-LM for the same concept pairs, our annotators preferred the relation predicted by COREC-LM in 64% of the cases, in 29% the link was annotated as equally good, and only in 7% COMET's generation was preferred.

To summarize, according to our manual evaluation, the dynamic models COMET and COREC-LM are better suited for generating meaningful knowledge paths that express implicit knowledge between sentences than ranked paths from the static CN knowledge graph, even though replacing vague by more specific relations slightly improves results. When comparing multihop paths to direct relations established between the same concept pairs, we find that longer paths tend to contain irrelevant or even misleading nodes, and that direct relations are preferred by human annotators. These findings support our proposed joint framework CO-NNECT, which gives preference to direct relations and utilizes multihop paths only if no direct connection between concepts can be revealed.

4.3 Automatic Evaluation Against Gold

Our goal is to generate meaningful paths that convey implicit knowledge between sentences. In our automatic evaluation we compare the set of model-generated paths between all concept pairs from two related sentences to the implicit knowledge annotated in IKAT and ARC for these sentences, using similarity metrics.

Since the generated relation and path representations differ from the annotated natural sentences, we approximate a common representation as follows: We **encode the golden annotations of implicit knowledge** – usually short sentences – using three settings: (i) **Silver Paths:** we encode their relational knowledge, by extracting all concepts from each golden implicit knowledge sentence (*My dog has a bone* → *dog, bone*) using the CN-extraction tool CoCo-Ex (Becker et al., 2021a), and predict the relations between them using COREC-LM, trained on CN-13 (*dog*, HASA, *bone*). If a sentence contains more than two relations, we concatenate the predicted relation triples. (ii) IKAT provides manual annotations of ConceptNet relations for the golden implicit knowledge sentences, which we use as **Gold Paths** (*The tree is in the garden* → *tree* ATLOCATION *garden*). (iii) **Gold-NL**: Here we use the implicit knowledge (in natural language) as provided in the datasets: IKAT's implicit knowledge sentences and ARC's implicit warrants.

For **encoding the generated paths** we apply two settings: (i) we concatenate all concepts and relations within the paths; (**Generated Paths**) and (ii) we translate the relation triples and paths to (pseudo) natural language using templates provided by ConceptNet (e.g. c_s CAUSES c_t → *The effect of* c_s *is* c_t; **Generated Paths-NL**).

We apply two **automatic similarity metrics**, comparing (a) Generated vs. Silver Paths, (b) Generated Paths-NL vs. Gold-NL, and (c) Generated vs. Gold Paths (only IKAT). (i) We encode each representation as described above using ConceptNet numberbatch embeddings (Speer et al., 2017) (for multiword concepts we average the embeddings of all non-stopwords), and compute cosine similarity between them, and (ii) we use BERTScore F1 (Zhang et al., 2020) to compare representations, which computes string similarity using contextualized embeddings. Both metrics lie in $[-1, 1]$.

Results. Table 3 shows that the paths generated by combining COREC-LM and COMET in our framework CO-NNECT achieve the highest similarity scores according to Numberbatch-Cosim on **IKAT** in setting (a) and (b), while for (c) we get the highest Cosim scores for ranked CN-graphs with replaced vague relations. According to BERTScore, either COREC-LM (setting a) or COMET (setting b) applied separately, or both applied in combination (setting c) achieve highest

	COR	COM	CONN	CN	CN-r
(a) Generated Paths vs. Silver Paths					
IKAT	.61/**.85**	.54/.82	**.62**/.84	.57/.78	.58/.80
ARC	.41/.84	.39/.82	**.42**/**.86**	.40/.77	.40/.78
(b) Generated Paths-NL vs. Gold-NL					
IKAT	.69/.81	.65/**.83**	**.70**/.81	.65/.75	.69/.76
ARC	.72/.81	.66/**.82**	.72/.81	.71/.75	**.77**/.76
(c) Generated Paths vs. Gold Paths					
IKAT	.57/.78	.49/.78	.58/**.79**	.66/.73	**.67**/.74

Table 3: Comparing generated paths to implicit knowledge annotations on IKAT and ARC, measured by Cosim/BERTScore (F1).

results on IKAT. On **ARC**, CO-NNECT achieves both highest Cosim and BERTScores in setting (a), while in (b) we get the best scores for CN-r according to Cosim, and the best scores for COMET according to BERTScore.

Summarizing our insights from automatic evaluations, we find that **COREC-LM** achieves high scores when applied separately *or* in combination with COMET (CO-NNECT). **COMET** applied in isolation does not yield the highest scores, but helps to boost COREC-LM's performance in the joint CO-NNECT framework. **Ranked CN-graphs** achieve highest Cosim in two settings/datasets (ARC–b; IKAT–c), but we do not find significant improvements when replacing vague relations in CN-graphs (expect for Cosim in setting b). This can be explained by the fact that even though many RELATEDTO and HASCONTEXT instances could be replaced, for both datasets a large amount of vague relations still remain (56.6% of RELATEDTO/53.2% of HASCONTEXT in IKAT; 29.3% RELATEDTO/63% HASCONTEXT in ARC). Therefore, the vague relation types in the CN-graphs still remain problematic when representing implicit knowledge.

When comparing our **manual** evaluation results to the **automatic** scores, we find that the generations that were manually evaluated as most relevant and meaningful explanations of implicit knowledge are not always highest-ranked by automatic metrics, which points to two limitations of our automatic evaluation: Besides well-known issues regarding the reliability, interpretability, and biases of automatic metrics (Callison-Burch et al., 2006), we evaluate the generated paths against an annotated *reference* – paths or sentences – which is often only one among several valid options for expressing the implicit knowledge (cf. Becker et al. 2017). This means that a generated path may still be a relevant explicitation of implicit information, even if *not* similar to the reference. Hence, automatic scores are to be considered with caution.

5 Conclusion

Our work aims to leverage commonsense knowledge in the form of single and multihop paths, to establish knowledge connections between concepts from different sentences, as a form of explicitation of implicitly conveyed information. We combine existing relation classification and target prediction models in a dynamic knowledge prediction framework, CO-NNECT, utilizing language models finetuned on knowledge relations from ConceptNet. We compare against a path ranking system that employs static knowledge from ConceptNet as a baseline and evaluate the quality of the obtained paths (i) through manual evaluation and (ii) using automatic similarity metrics, by comparing generated paths to annotations of implicit knowledge in two argumentative datasets. Our evaluations show that we obtain the highest number of connections from the static ConceptNet graph, however, they are often noisy due to unrelated intermediate nodes, and – even after replacements – still contain many unspecific relations. Our framework CO-NNECT, instead, combines relation classification and target prediction, leveraging the *high precision* of the former, and the *ability to perform forward chaining* of the latter, and obtain high-quality, meaningful and relevant knowledge paths that reveal implicit knowledge conveyed by the text, as shown in a profound manual evaluation experiment.

We believe that CO-NNECT is a useful framework which can be applied for different tasks, such as enriching texts with commonsense knowledge relations and paths, for dynamically enriching knowledge bases, or for building knowledge constraints for language generation. In Becker et al. (2021b) for example we inject single- and multihop commonsense knowledge paths predicted by CO-NNECT as constraints into language models and show that this improves the model's ability of generating sentences that explicate implicit knowledge which connects sentences in texts. We furthermore believe that the paths established with CO-NNECT, which can provide explicitations of implicit knowledge, can be useful to enhance many other NLP downstream tasks, such as argument classification, stance detection, or commonsense reasoning.

Acknowledgements

This work has been funded by the DFG within the project ExpLAIN as part of the Priority Program "Robust Argumentation Machines" (SPP-1999). We thank our annotators for their contribution.

References

Maria Becker, Katharina Korfhage, and Anette Frank. 2020. Implicit Knowledge in Argumentative Texts: An Annotated Corpus. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 2316–2324, Online.

Maria Becker, Katharina Korfhage, and Anette Frank. 2021a. COCO-EX: A Tool for Linking Concepts from Texts to ConceptNet. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL), Demo Papers*, Online.

Maria Becker, Siting Liang, and Anette Frank. 2021b. Reconstructing Implicit Knowledge with Language Models. *Accepted at: Deep Learning Inside Out (DeeLIO): Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*.

Maria Becker, Michael Staniek, Vivi Nastase, and Anette Frank. 2017. Enriching Argumentative Texts with Implicit Knowledge. In *Applications of Natural Language to Data Bases (NLDB) - Natural Language Processing and Information Systems*, Lecture Notes in Computer Science, pages 84–96. Springer.

Maria Becker, Michael Staniek, Vivi Nastase, and Anette Frank. 2019. Assessing the Difficulty of Classifying ConceptNet Relations in a Multi-Label Classification Setting. In *Proceedings of RELATIONS - Workshop on Meaning Relations between Phrases and Sentences*, pages 1–15, Gothenburg, Sweden.

Antoine Bosselut, Ronan Le Bras, , and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Ting-Yun Chang, Yang Liu, Karthik Gopalakrishnan, Behnam Hedayatnia, Pei Zhou, and Dilek Hakkani-Tur. 2020. Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 74–79, Online. Association for Computational Linguistics.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. 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 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.

Pavan Kapanipathi, Veronika Thost, Siva Patel, Spencer Whitehead, Ibrahim Abdelaziz, Avinash Balakrishnan, Maria Chang, Kshitij Fadnis, Chulaka Gunasekara, Bassem Makni, Nicholas Mattei, Kartik Talamadupula, and Achille Fokoue. 2020. Infusing knowledge into the textual entailment task using graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8074–8081.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Debjit Paul and Anette Frank. 2019. Ranking and selecting multi-hop knowledge paths to better predict human needs. 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 3671–3681, Minneapolis, Minnesota. Association for Computational Linguistics.

Debjit Paul and Anette Frank. 2020. Social commonsense reasoning with multi-head knowledge attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2969–2980, Online. Association for Computational Linguistics.

Debjit Paul, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank. 2020. Argumentative Relation Classification with Background Knowledge. In *Proceedings of the International Conference on Computational Models of Argument (COMMA)*, pages 319–330, Online.

Andreas Peldszus and Manfred Stede. 2016. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon 2015 / Vol. 2*, pages 801–815, London. College Publications.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2018. Commonsense knowledge base completion and generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 141–150, Brussels, Belgium. 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. *EMC$\hat{2}$: 5th Edition. Co-located with NeurIPS'19*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy. Association for Computational Linguistics.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany. Association for Computational Linguistics.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. *EMNLP Findings*, pages 4129–4140.

Dirk Weissenborn, Tomas Kocisky, and Chris Dyer. 2018. Dynamic Integration of Background Knowledge in Neural NLU Systems. *ICLR 2018*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2361–2364, New York, NY, USA. Association for Computing Machinery.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *ICLR*.

Xiaobin Zhang, Fucai Chen, and Ruiyang Huang. 2018. A Combination of RNN and CNN for Attention-based Relation Classification. In *Procedia Computer Science*, volume 131, pages 911 – 917.

APPENDIX

A Constructing the RANDOM Class for Training COREC-LM in an Open World Setting

Our downstream application task – finding connections between concepts – requires that our relation classifier also learns to detect that no direct relation holds between a given pair of concepts. We thus extend the data for training and testing COREC-LM with a RANDOM class which contains concept pairs that are not related which we add to CN-100k and CN-13 Instances for this class are generated similarly to Vylomova et al. (2016): 50% of them are opposite pairs which we obtain by switching the order of concept pairs within the same relation, and 50% are corrupt pairs, obtained by replacing one concept in a pair with a random concept from the same relation. Corrupt pairs ensure that COREC-LM learns to encode relation instances rather than simply learning properties of the word classes. We add these instances (2070 for training and 260 for development and testing, respectively) to CN-100k and CN-13 when training and evaluating in an open world setting.

	COREC-LM	COMET	CONNECT	CN Subgraphs
IKAT	AtLocation(25%) HasProperty(20%) IsA(17%)	IsA(19%) HasA(18%) Causes(17%)	AtLocation(22%) IsA(17%) HasProperty(16%)	RelatedTo(71%) HasContext(8%) IsA(7%)
ARC	AtLocation(31%) IsA(18%) HasProperty(14%)	AtLocation(22%) Causes(20%) HasA(18%)	AtLocation(27%) IsA(15%) HasProperty(10%)	RelatedTo(72%) HasContext(8%) IsA(7%)

Table 4: Most frequently used relations when constructing single and multihop knowledge paths using COMET, COREC-LM, their combination, and ranked subgraphs from CN.

B Relations Used for Constructing Single- and Multihop Paths

Table 4 lists the three most frequently used relations when constructing single and multihop knowledge paths using COMET, COREC-LM, their combination, and ranked subgraphs, respectively for the two datasets IKAT and ARC. The top three relations used by **COREC-LM** within both datasets are AtLocation, HasProperty, and IsA. Interestingly, besides IsA and HasA, **COMET** frequently uses the only causal relation in the CN inventory Causes. In contrast to COREC-LM and COMET, the ranked CN-graphs are constructed using mostly the very general relation RelatedTo, followed by the likewise vague relation HasContext. When excluding paths that contain RelatedTo, only 2,551 connected concept pairs remain in IKAT and 6,858 in ARC.

Computing All Quantifier Scopes with CCG

Miloš Stanojević
School of Informatics
University of Edinburgh
m.stanojevic@ed.ac.uk

Mark Steedman
School of Informatics
University of Edinburgh
steedman@inf.ed.ac.uk

Abstract

We present a method for computing all quantifer scopes that can be extracted from a single CCG derivation. To do that we build on the proposal of Steedman (1999, 2011) where all existential quantifiers are treated as Skolem functions. We extend the approach by introducing a better packed representation of all possible specifications that also includes node addresses where the specifications happen. These addresses are necessary for recovering all, and only, possible readings.

1 Introduction

Quantifiers often introduce a peculiar type of semantic ambiguity. Take for instance the following sentence: Every farmer owns a donkey. This sentence has two readings: a *wide reading* where there is one donkey that all farmers share and *narrow reading* where each farmer has a different donkey. If we express these readings as first-order logic they would look as follows:

Wide:
$\exists a\,[donkey'(a) \wedge \forall b\,[farmer'(b) \Rightarrow own'(b,a)]]$
Narrow:
$\forall b\,[\exists a\,[donkey'(a) \wedge (farmer'(b) \Rightarrow own'(b,a))]]$

From these formulas it is clear where the name for different readings come from. In the *wide* reading the existential quantifier takes the wide scope i.e. it contains the universal quantifier. In the *narrow* reading the existential quantifier's scope does not cover the universal quantifier.

Any theory of the syntax-semantics interface needs to account for the fact that quantifiers can introduce scope ambiguity. Early approaches to this problem involved either representing the two meanings with distinct logical forms like the above, obtained from the surface string either by treating *every farmer* and *a donkey* as generalized quantifiers or "quantifying in" in either order to a proposition containing distinguished variables (Montague, 1973), or via equivalent structure-changing operations of "quantifier raising" (May, 1985). Later approaches decoupled scope from syntactic derivation by the use of "storage" to pass scope information (Cooper, 1983; Keller, 1988). However, all of these approaches overgenerate unattested readings for certain examples involving coordination, first noted by (Geach, 1970) and considered in section 3 below. The approach of (Steedman, 2011) can be thought of as reuniting a storage-like account with surface-compositional syntactic derivation.

2 Computing Scope with CCG

Steedman (1999, 2011) introduces a different view of existential quantifiers, according to which the only true quantifiers are universal quantifiers and that existential quantifiers can be treated as generalized Skolem terms in the following way:

$$\textit{Wide}: \quad \forall b\left[farmer'(b) \Rightarrow own'(b, \mathrm{sk}^{\{\}}_{donkey'})\right]$$
$$\textit{Narrow}: \quad \forall b\left[farmer'(b) \Rightarrow own'(b, \mathrm{sk}^{\{b\}}_{donkey'})\right]$$

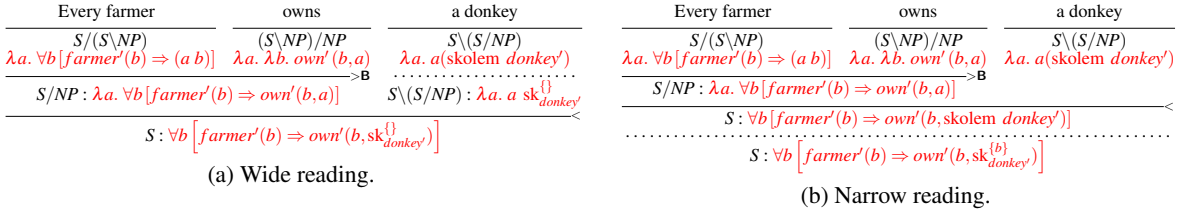
Here, $\mathrm{sk}^{\alpha}_{\beta}$ represents the Skolem function whose arguments are variables of type α and whose result is of type β.

In the wide scope reading $\mathrm{sk}^{\{\}}_{donkey'}$ is a Skolem constant (Skolem function with no arguments). This means that it will produce only a unique value of type *donkey*′, somewhat like a proper name. In the narrow scope reading $\mathrm{sk}^{\{b\}}_{donkey'}$ is a Skolem function that has the variable b bound by the universal as its argument. This function will produce a different value for each b, in other words there will be a different *donkey*′ for each *farmer*′.

Other non-universal generalized quantifiers are also treated as Skolem terms. Steedman (2011) also discusses negation which we do not present here, but our approach naturally extends to it. We do not deal with intentionality.

This view of quantifiers allows for a simple

Proceedings of the 14th International Conference on Computational Semantics, pages 33–37
June 17–18, 2021.



(a) Wide reading.

(b) Narrow reading.

Figure 1: Two different readings.

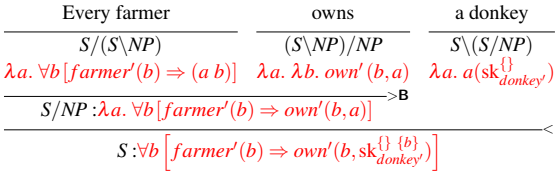


Figure 2: Packed representation.

syntax-semantics interface. CCG derivations for wide and narrow readings are presented in Figure 1, for one of the two derivation trees allowed by CCG. Syntactic component of these two trees is the same, only the semantics differ. Semantic entry for all words are the usual lambda expressions except for the indefinite articles whose entry is $\lambda a.\lambda b.b(\text{skolem } a)$. Here *skolem a* is a *underspecified* Skolem term of type a. An underspecified Skolem term becomes a Skolem function/constant when it is *specified*. Skolem specification is marked in the derivation tree with a dotted underline, and influences only the logical form, converting an underspecified Skolem terms by giving it as arguments all universally bound variables into whose scope it has been brought by the derivation so far. In Figure 1a that set is empty, so the result of Skolem specification is a Skolem constant, yielding the wide scope reading. In Figure 1b, that set includes the single variable b. By choosing to specify at a different point in the derivation, we get a different narrow-scope reading for the sentence.

In order to prevent overgeneration of unattested readings, we must impose a further rather natural constraint on Skolem specification requiring that any embedded unspecified Skolem terms are specified at the same time in the same environment. Thus we get the following readings for "every farmer owns a donkey that ate a hat":

$\forall b[farmer'(b) \Rightarrow own'(b, \text{sk}^{\{\}}_{\lambda a.donkey'(a) \wedge ate'(a,\text{sk}^{\{\}}_{hat'})})]$

$\forall b[farmer'(b) \Rightarrow own'(b, \text{sk}^{\{b\}}_{\lambda a.donkey'(a) \wedge ate'(a,\text{sk}^{\{b\}}_{hat'})})]$

$\forall b[farmer'(b) \Rightarrow own'(b, \text{sk}^{\{b\}}_{\lambda a.donkey'(a) \wedge ate'(a,\text{sk}^{\{\}}_{hat'})})]$

However, we exclude a fourth reading with a wide-scope Skolem constant donkey eating multiple farmer-dependent hats:[1]

$\#\forall b[farmer(b) \Rightarrow own'(b, \text{sk}^{\{\}}_{\lambda a.donkey'(a) \wedge ate'(a,\text{sk}^{\{b\}}_{hat'})})]$

To ensure that all available readings are obtained, it is inefficient to choose all possible specification points in the derivation, because most of them yield duplicate results where there has been no change in the set of scoping variables. To eliminate such redundancy, Steedman (2011) proposed a *packed representation* presented in Figure 2 where the Skolem term is associated with multiple bindings. At points in the derivation where the binding environment of the function changes, a new argument combination is introduced.

3 Problems with Taking Scope over Coordination

The proposal of Steedman (2011) was implemented by Kartsaklis (2010) and it works quite well for examples that we have seen so far. However, coordination poses some challenges for the packed representation. Consider coordination of two universal quantifiers in Figure 3a. Here $NP^{\uparrow}$ is a shorthand for a type-raised NP. For a moment ignore additional annotations in the arguments of the Skolem functions. In this example, the specification of *an apple* will either happen before it is combined with the universals, or after. This means that either it will be in the scope of both or none. The only two readings are given in Figure 3b. However, if we were to unpack the packed formula by computing all combinations of Skolem arguments we would get four readings, including the impossible reading of *an apple* being within scope of one universal quantifier but not the other. We stress that this is a problem arising from the packed representation, not the theory of scope itself.

It may look like the solution to this problem is simple: take all Skolems stemming from the

[1] This condition was inadvertently omitted from the original proposal.

same noun phrase and combine their arguments in order i.e. first arguments of both Skolems go together and second arguments of both Skolems go together. While this solves the example in Figure 3, it does not work on that in Figure 4 where we coordinate one universal and one existential quantifier. Here there is no clear correspondence between arguments of two Skolem functions: one of them has two different arguments ($\{\}$ and $\{a\}$) while the other one has only one possible argument ($\{\}$). Of course, in principle the difference in the number of possible combinations could be significantly larger and it may not be clear how to combine them. To solve this problem we need a more principled solution that directly reflects the mechanism of the non-packed derivations.

4 Proposed Solution

The packed representation can be seen as a dynamic programming approach to computing all possible orders of specifications of Skolem terms. However, the packed representation of Steedman (2011) that we considered so far is incomplete: from a given packed representation we cannot reconstruct the non-packed representations that are encoded in it. That is caused by the missing information of the location in the tree where the specification was done. We extend the packed representation with this information: whenever a new argument combination is added, together with it we add the Gorn address of the current node. For instance $\mathrm{sk}_{apple'}^{\{\}^{trrl}\ \{a\}^{t}}$ from Figure 4a signifies that there are two possible arguments for this Skolem function: an empty argument list specified at Gorn address $trrrl$ ($top \rightarrow right \rightarrow right \rightarrow right \rightarrow left$) and a non-empty argument $\{a\}$ at address t (top). We know that all the Gorn addresses for a given Skolem function will be on a single path from the root of the tree to the determiner that introduced it into derivation. This means that, for a given function, we can sort all addresses by their height in the tree.

Assume we have a Skolem function with k possible argument sets $e_1, e_2, \ldots, e_k$ sorted by the height of their Gorn addresses $g_1, g_2, \ldots, g_k$ such that g_k is closest to the root of the tree. We can say that every argument set e_i corresponds to the specializations done on any node g for which it holds $g_i \leq g < g_{i+1}$.[2] In other words g can be any node between g_i and g_{i+1}, including g_i but excluding g_{i+1}. If we take again $\mathrm{sk}_{apple'}^{\{\}^{trrl}\ \{a\}^{t}}$ as an example we can say that the argument $\{\}$ corresponds to specialization of Skolem function for nodes $trrl$, trr and tr.

Additional important point is that we know for certain that g_1 is the address of the leaf of the tree because that is the first point in the derivation where the specification can be done. This is important because in the cases of coordination the logical formula can have copies of the Skolem term that comes from the same noun phrase. We can use the Gorn address of the leaf to identify the Skolem terms that originate in the same noun phrase. Steedman (2011) uses a special index to keep track of this information, but that index is not necessary in our representation due to the existance of Gorn addresses.

Now we can define unpacking of the new version of the packed representation. We will illustrate it with the example packed formula from the top node of Figure 4a: $\forall a \left[man'(a) \Rightarrow eat'\left(a, \mathrm{sk}_{apple'}^{\{\}^{trrl}\ \{a\}^{t}}\right)\right] \wedge eat'\left(\mathrm{sk}_{woman'}^{\{\}^{tlrrl}}, \mathrm{sk}_{apple'}^{\{\}^{trrl}}\right)$

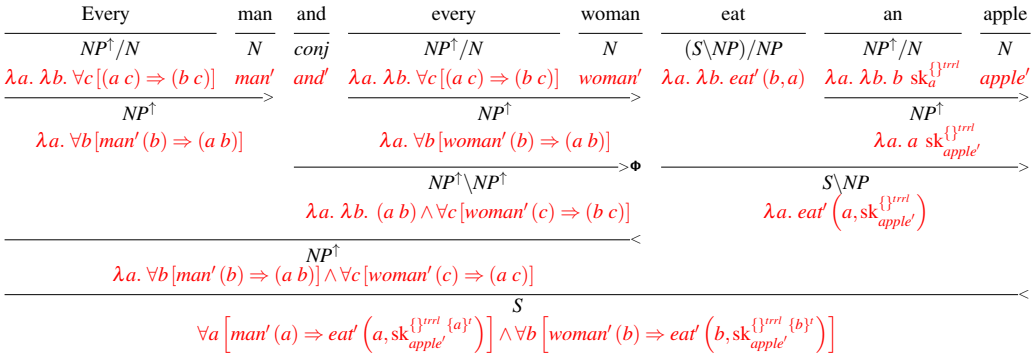
step 1 Group Skolem terms by the NP they belong to. For that we can use the first Gorn address that specifies the leaf node. In the example that would give $\{\mathrm{sk}_{woman'}^{\{\}^{tlrrl}}\}$ for the first NP and $\{\mathrm{sk}_{apple'}^{\{\}^{trrl}\ \{a\}^{t}}, \mathrm{sk}_{apple'}^{\{\}^{trrl}}\}$ for the second.

step 2 For each group of the Skolems extract the unique Gorn addresses where specification changes. In this example that would be $\{tlrrl\}$ for the first noun phrase and $\{trrl, t\}$ for the second.

step 3 Compute the Cartesian product of the sets of Gorn addresses. That will give all possible combinations of specification points. Each combination will correspond to one possible reading of the sentence. In the example that will give $\{(tlrrl, trrl), (tlrrl, t)\}$.

step 4 To transform each entry to a reading we filter the Skolem arguments by the Gorn address. Let us consider how we extract the reading for entry $(tlrrl, t)$. Filtering arguments for the first noun phrase Skolem term $\{\mathrm{sk}_{woman'}^{\{\}^{tlrrl}}\}$ with $tlrrl$ is easy because there is only one entry that matches it exactly. Filtering arguments for the second noun phrase is more interesting because there are two copies of it. We need to

[2] For simplicity, when $g_k \neq t$ we can consider $g_{k+1} = t$ in order to have a complete coverage to the root of the tree.



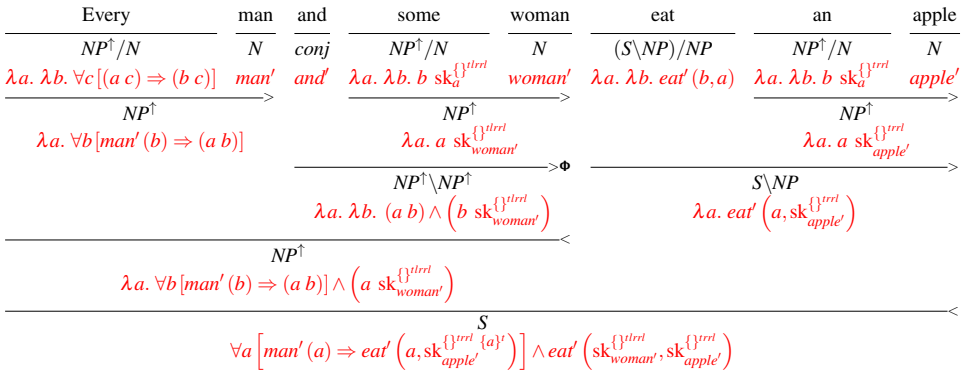
(a) Packed derivation.

$$\forall a \left[man'(a) \Rightarrow eat' \left(a, \mathrm{sk}_{apple'}^{\{\}^{trrl}} \right) \right] \wedge \forall b \left[woman'(b) \Rightarrow eat' \left(b, \mathrm{sk}_{apple'}^{\{\}^{trrl}} \right) \right]$$

$$\forall a \left[man'(a) \Rightarrow eat' \left(a, \mathrm{sk}_{apple'}^{\{a\}^{t}} \right) \right] \wedge \forall b \left[woman'(b) \Rightarrow eat' \left(b, \mathrm{sk}_{apple'}^{\{b\}^{t}} \right) \right]$$

(b) Readings.

Figure 3: Coordination with two universal quantifiers.



(a) Packed derivation.

$$\forall a \left[man'(a) \Rightarrow eat' \left(a, \mathrm{sk}_{apple'}^{\{\}^{trrl}} \right) \right] \wedge eat' \left(\mathrm{sk}_{woman'}^{\{\}^{tlrrl}}, \mathrm{sk}_{apple'}^{\{\}^{trrl}} \right)$$

$$\forall a \left[man'(a) \Rightarrow eat' \left(a, \mathrm{sk}_{apple'}^{\{a\}^{t}} \right) \right] \wedge eat' \left(\mathrm{sk}_{woman'}^{\{\}^{tlrrl}}, \mathrm{sk}_{apple'}^{\{\}^{trrl}} \right)$$

(b) Readings.

Figure 4: Coordination with one universal and one existential quantifier.

select for specification on node t. In the first copy $\mathrm{sk}_{apple'}^{\{\}^{trrl}\ \{a\}^{t}}$ we just select argument $\{a\}$ since it corresponds to node t. In the second copy $\mathrm{sk}_{apple'}^{\{\}^{trrl}}$ we select for $\{\}$ because it covers all nodes from $trrl$ to the root including t.

5 Conclusion

This approach is really just a full dynamic programming representation of the unpacked representations that could easily be extracted from this representation. We do not have to explicitly encode all the nodes where specification happens, but only for the places where that specification changes the existing result and we also encode exactly at which places in the tree this happens.

Here we have described how to get all possible readings from a single CCG derivation. However, in some cases there can be alternative CCG derivations that can provide additional readings. To get those readings we can apply the same method on all alternative derivations either by chart parsing, as described in (Steedman, 2011), or by recovering alternative derivations with the *tree-rotation* operation (Niv, 1994; Stanojević and Steedman, 2019)

Evang and Bos (2013) show that there is a strong preference for subject to take scope over object.

Our representation of Skolem terms could be extended to encode the information of the type of noun phrase they originate from. With this extension we could rank the extracted readings by *subject* > *object* preference.

The implementation of our approach is available at https://github.com/stanojevic/CCG-Quantifiers.

Acknowledgments

This work was supported by ERC H2020 Advanced Fellowship GA 742137 SEMANTAX grant.

References

Robin Cooper. 1983. *Quantification and Syntactic Theory*. Reidel, Dordrecht.

Donald Davidson and Gilbert Harman, editors. 1972. *Semantics of Natural Language*. Reidel, Dordrecht.

Kilian Evang and Johan Bos. 2013. Scope Disambiguation as a Tagging Task. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 314–320, Potsdam, Germany. Association for Computational Linguistics.

Peter Geach. 1970. A program for syntax. *Synthèse*, 22:3–17. Reprinted as Davidson and Harman 1972:483–497.

Dimitrios Kartsaklis. 2010. Wide-coverage CCG parsing with quantifier scope. Master's thesis, University of Edinburgh.

William Keller. 1988. Nested Cooper storage. In Uwe Reyle and Christian Rohrer, editors, *Natural Language Parsing and Linguistic Theory*, pages 432–447. Reidel, Dordrecht.

Robert May. 1985. *Logical Form*. MIT Press, Cambridge, MA.

Richard Montague. 1973. The Proper Treatment of Quantification in Ordinary English. In K. J. J. Hintikka, J. M. E. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, pages 221–242. Springer Netherlands, Dordrecht.

Michael Niv. 1994. A Psycholinguistically Motivated Parser for CCG. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 125–132, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Miloš Stanojević and Mark Steedman. 2019. CCG Parsing Algorithm with Incremental Tree Rotation. 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 228–239, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Steedman. 1999. Alternating Quantifier Scope in CCG. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 301–308, USA. Association for Computational Linguistics.

Mark Steedman. 2011. *Taking Scope: The Natural Semantics of Quantifiers*. The MIT Press.

Encoding Explanatory Knowledge for Zero-shot Science Question Answering

Zili Zhou[1,2], Marco Valentino[1], Dónal Landers[2], André Freitas[1,2,3]
[1]Department of Computer Science, University of Manchester, United Kingdom
[2]Digital Experimental Cancer Medicine Team
Cancer Research UK Manchester Institute, United Kingdom
[3]Idiap Research Institute, Switzerland
{zili.zhou, marco.valentino}@manchester.ac.uk
donal.landers@digitalecmt.org
andre.freitas@idiap.ch

Abstract

This paper describes N-XKT (Neural encoding based on eXplanatory Knowledge Transfer), a novel method for the automatic transfer of explanatory knowledge through neural encoding mechanisms. We demonstrate that N-XKT is able to improve accuracy and generalization on science Question Answering (QA). Specifically, by leveraging facts from background explanatory knowledge corpora, the N-XKT model shows a clear improvement on zero-shot QA. Furthermore, we show that N-XKT can be fine-tuned on a target QA dataset, enabling faster convergence and more accurate results. A systematic analysis is conducted to quantitatively analyze the performance of the N-XKT model and the impact of different categories of knowledge on the zero-shot generalization task.

1 Introduction

Contemporary Question Answering (QA) is evolving in the direction of addressing more abstractive reasoning tasks (Thayaparan et al., 2020; Dua et al., 2019; Clark et al., 2018; Mihaylov et al., 2018), supported by multi-hop inference (Khot et al., 2020; Yang et al., 2018) and explanatory scientific facts (Jansen and Ustalov, 2019; Jansen et al., 2018, 2016).

This trend of aiming to address more complex, multi-evidence and chained inference is pushing the envelope for novel representation and architectural patterns (Ding et al., 2019; Qiu et al., 2019; Asai et al., 2020; Thayaparan et al., 2019; Kundu et al., 2019; Valentino et al., 2021), which are moving from modelling meaning from immediate distributional semantics patterns into deeper abstractive capabilities. This poses a paradigmatic challenge on the design of QA architectures, which need to operate over high-level semantic patterns and acquire the necessary knowledge to perform abstraction (Clark et al., 2018). At the same time, the design of new strategies to incorporate explanatory knowledge into neural representation has the potential to address fundamental data efficiency problems and promote zero-shot generalisation on out-of-distribution examples.

Explanation-based Science QA (Jansen et al., 2018) provides a rich framework to evaluate these emerging requirements, as the task typically requires multi-hop reasoning through the composition of explanatory facts. While existing approaches in the field mainly focus on the construction of natural language explanations (Jansen et al., 2018; Jansen and Ustalov, 2019), this work aims to explore the impact of explanatory knowledge on zero-shot generalisation.

In this paper, we argue that explanation-centred corpora can serve as a resource to boost zero-shot capabilities on Question Answering tasks which demand deeper inference. To this end, we explore the adoption of latent knowledge representations for supporting generalisation on downstream QA tasks requiring multi-hop inference.

Our hypothesis is that explanatory scientific knowledge expressed in natural language can be transferred into neural network representations, and subsequently used to achieve knowledge based inference on scientific QA tasks. To validate this hypothesis, this paper proposes a *unified* approach that frames Question Answering as an explanatory knowledge reasoning problem. The unification between the two tasks allows us to explore the adoption of pre-training strategies over explanatory knowledge bases, and subsequently leverage the same paradigm to generalise on the Question Answering task.

An empirical evaluation is performed on Transformers-based architectures adopting the WorldTree corpus as a knowledge base (Xie et al., 2020; Jansen et al., 2018) and measuring generalisation on ARC (Clark et al., 2018) and OpenbookQA

Proceedings of the 14th International Conference on Computational Semantics, pages 38–50
June 17–18, 2021.

(Mihaylov et al., 2018). The main contributions of this paper are as follows:

- We propose N-XKT, a neural mechanism for encoding and transferring explanatory knowledge for science QA. To the best of our knowledge, N-XKT is the first work tackling science QA tasks through the transfer of external explanatory knowledge via neural encoding mechanisms.
- We introduce the explanatory knowledge transfer task on explanation-centred knowledge bases, describing the methodology to implement N-XKT for knowledge acquisition and downstream Question Answering using Transformer-based models as neural encoders.
- We conduct a systematic empirical analysis to demonstrate the effectiveness of N-XKT on improving downstream QA accuracy and overall convergence speed in the training phase. An ablation analysis on different types of knowledge facts is performed to measure the impact of different knowledge categories.

2 Related Work

In this section we describe several works related to knowledge-based scientific QA.

Explanation Bank Explanation Bank[1] is a core component of the WorldTree corpus (Jansen et al., 2018; Xie et al., 2020). The dataset provides explanations for multiple-choice science questions in the form of graphs connecting questions and correct answers, where multiple sentences from a knowledge base (KB) are aggregated through lexical overlap between terms. The background knowledge used for the explanations is grouped in semi-structured tables, whose facts range from common-sense to core scientific statements. Explanation Bank has been proposed for the task of explanation regeneration (Jansen and Ustalov, 2019) – i.e. given a multiple-choice science question, regenerate the gold explanation supporting the correct answer. The explanation regeneration task has been framed as an Information Retrieval (IR) problem (Valentino et al., 2021). In this paper, we aim to leverage the knowledge expressed in the explanations to enhance generalisation and zero-shot capability on multiple-choice scientific question answering.

[1] http://cognitiveai.org/explanationbank/

Bidirectional Encoder Representations from Transformers BERT represents the foundation which defines the state-of-the-art in several NLP tasks (Devlin et al., 2019). This model adopts a Transformer-based architecture composed of several layers of attention (Vaswani et al., 2017) that are used to learn a deep bidirectional representation of language. BERT-based models have demonstrated remarkable results in Question Answering when directly fine-tuned on the answer prediction task or additionally pre-trained using domain specific knowledge (Clark et al., 2020; Beltagy et al., 2019). A recent line of research attempts to enrich the input of BERT with background knowledge in the form of explanations in order to boost generalisation and accuracy for challenging QA settings. Here, the explanations are explicitly constructed through the adoption of language models (Rajani et al., 2019) or information retrieval (IR) approaches (Valentino et al., 2021; Yadav et al., 2019). Conversely, this paper explores mechanisms to implicitly encode explanatory knowledge in the neural representation to improve the capability of performing downstream inference. Specifically, in this work, we adopt Transformers as text neural encoders.

Leveraging External Knowledge for Scientific QA Recently, many solutions have been proposed for science QA that leverage either external reference corpora (Khot et al., 2017; Khashabi et al., 2018; Zhang et al., 2018) or existing knowledge graphs (Li and Clark, 2015; Sachan et al., 2016; Wang et al., 2018; Musa et al., 2019; Zhong et al., 2019). Generally, previous works rely on Information Retrieval models or on structural embeddings for Knowledge Bases, while our work focuses on directly encoding explanatory knowledge, evaluating it in a downstream scientific QA setting.

3 Methodology

Scientific Question Answering has the distinctive property of requiring the articulation of multi-hop and explanatory reasoning. This can be contrasted with the lexical-retrieval style of factoid Question Answering. Additionally, the explanatory chains required to arrive at the correct answer typically operate at an abstract level, through the combination of definitions and scientific laws (Thayaparan et al., 2020). This characteristic makes the generalisation process more challenging, as the answer

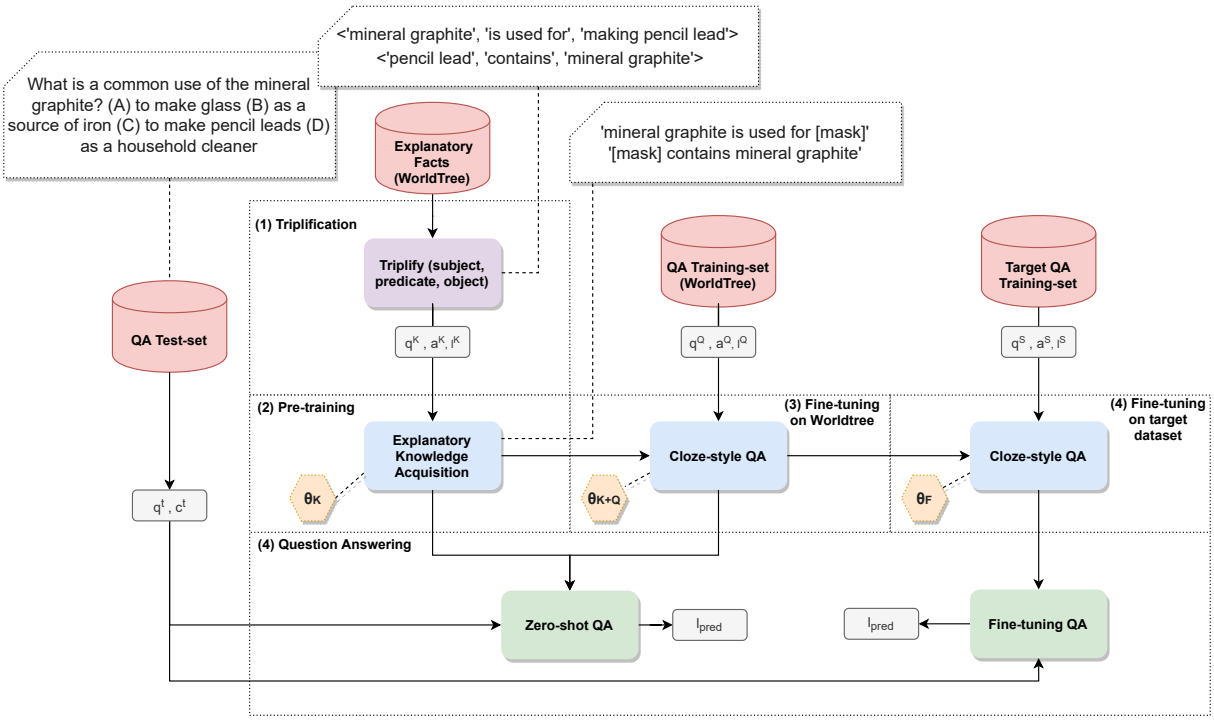


Figure 1: Outline of the proposed approach.

prediction model needs to acquire the ability to perform abstraction from the specific context in the question.

This paper hypothesises that it is possible to automatically transfer abstractive knowledge from explanatory facts into neural encoding representation for more accurate scientific QA, and for enabling zero-shot generalization. To this end, we propose N-XKT (Neural encoding based on eXplanatory Knowledge Transfer) which encodes abstractive knowledge into neural representation to improve the effectiveness in both zero-shot QA task and fine-tuning based QA task. The general neural encoding mechanism is evaluated adopting the following training tasks:

1. **Explanatory Knowledge Acquisition**: In this pre-training task, the N-XKT model encodes the explanatory textual knowledge from a set of explanatory facts into supporting embeddings. This process aims to acquire the necessary explanatory knowledge to test generalization on downstream science QA. We frame this problem as a knowledge base completion task. Specifically, after casting each explanatory fact in the knowledge base into a tuple composed of subject, object, and predicate, the model is trained on completing each fact by alternatively masking each element in the tuple (additional details can be found in section 3.1).

2. **Cloze-style Question Answering**: To keep the encoding mechanism consistent with the pre-training explanatory knowledge acquisition task, we cast Multiple-choice Question Answering into a cloze-style QA problem. Specifically, we train the N-XKT model to complete the question with the expected candidate answer. This task aims to acquire additional knowledge for addressing downstream science QA since the patterns in the questions are typically more complex than the background explanatory facts (additional details can be found in section 3.2).

The training tasks defined above can be used to encode different types and levels of knowledge into the N-XKT model, allowing us to perform a detailed evaluation on both zero-shot and fine-tuning-based Question Answering tasks.

Figure 1 shows a schematic representation of the proposed approach.

3.1 Explanatory Knowledge Acquisition

The WorldTree corpus (Jansen et al., 2018) contains natural language explanatory facts, which are stored in semi-structured tables whose columns correspond to semantic roles. The knowledge base

contains a total of 82 tables, where each table represents a different knowledge type, with different arity and argument types. N-XKT can be used as a unified approach for transferring knowledge from heterogeneous explanatory facts via a neural encoding mechanism.

To acquire the explanatory knowledge in a unified way for subsequent transfer learning, we normalize the semi-structured facts using a binary predicate-argument structure as typical practice in standard knowledge-base completion tasks (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015). Specifically, for each table, we map the columns into three main components: subject, predicate, and object. After performing the mapping for each table in the knowledge base, we generate triples for all the facts in the knowledge base.

By framing the explanatory knowledge acquisition task as a knowledge base completion problem, we alternatively mask subjects and objects from the triples and train the model to predict the missing component in the triple by giving in input the remaining ones. Specifically, we simulate a question answering problem adopting either subject or object as an answer, and the other two components in the triple as a question.

The neural encoder of N-XKT learns an embedding representation for each pair in input. A softmax layer is added on top of the embedding to predict the probability of the missing component in the triple. The configuration adopted for the N-XKT model is described in equation 1;.

$$\theta_K \leftarrow \text{argmin}_\theta \mathcal{L}(\textbf{N-XKT}_\theta(q^K, a^K), l^K) \quad (1)$$

Here, q^K and a^K represent the simulated question-answer pair generated from a generic explanatory fact triple, while l^K represents the target labels (i.e. 1 if a is the correct component for completing the triple, 0 otherwise). θ_K is the set of parameters optimised during the explanatory knowledge acquisition stage. The negative samples are generated by replacing each correct answer with a random component extracted from different explanatory facts in the knowledge base.

The transformer neural network is used as a textual neural encoder component of N-XKT, where each question-answer pair is compiled into the input token sequence:

$$[CLS][question][SEP][answer][SEP] \quad (2)$$

The final hidden vector $C \in \mathbb{R}^H$ of the Transformer neural network that corresponds to the first input token ([CLS]) is used as an embedding to perform the final classification.

3.2 Cloze-style Question Answering

Normally, the explanatory knowledge patterns do not contain the complete information to address downstream Question Answering. However, the questions in WorldTree can be used as additional knowledge to deal with complex structured science questions, allowing N-XKT to learn to recognize more complex patterns.

To acquire additional knowledge while keeping the encoding mechanism consistent with the pre-training explanatory knowledge acquisition task, we cast Multiple-choice Question Answering into a cloze-style QA problem. The particular encoding configuration of the N-XKT model can be used in fact to address this type of question answering problems, where the model is trained to complete the question with the expected candidate answer. The detailed parameters and inputs adopted for cloze-style QA are described in equation 3:

$$\theta_{K+Q} \leftarrow \text{argmin}_\theta \mathcal{L}(\textbf{N-XKT}_{\theta_K}(q^Q, a^Q), l^Q) \quad (3)$$

The setting adopted for cloze-style QA is similar to the one adopted for explanatory knowledge acquisition, but with two main differences: 1) In this case, the question q^Q, the answer a^Q, and the target label l^K are generated from the WorldTree multiple-choice question answering set, where the right candidate answer of each question acts as a positive sample, and the incorrect candidate answers act as the negative samples. 2) The initial parameters are initially set with θ_K, that is, we adopt the parameters that have been optimised during the explanatory knowledge acquisition stage.

3.3 Zero-shot and Fine-tuning Settings

Given a multiple-choice science question, N-XKT can perform question answering by framing it as a sequence classification problem, where the question is paired with each candidate answer to compute a probability score. The candidate choice with highest score can then be selected as the predicted answer. We evaluate N-XKT in two different settings: zero-shot and fine-tuning-based QA.

Regarding the **zero-shot setting**, the N-XKT is trained only on the explanatory knowledge acquisition task and then directly tested on downstream Question Answering. We also evaluate the model

trained jointly on explanatory knowledge and science questions in WorldTree, evaluating its generalization capabilities on different multiple-choice Question Answering datasets, such as ARC[2] (Clark et al., 2018) and OpenBook QA[3] (Mihaylov et al., 2018). For each pair of question and candidate answer, the scores are computed as described in equation 4. Here, (q^T, c^T) represent the test question and a candidate answer, while l^T_{pred} is the score predicted by the model.

$$l^T_{pred} = \textbf{N-XKT}_{\theta_{K+Q}}(q^T, c^T) \quad (4)$$

In the **fine-tuning setting**, the N-XKT model is additionally fine-tuned on each target QA dataset as in equation 6. Here, (q^S, a^S) represents a question-answer pair from the target QA training set, while l^S is the label indicating whether the answer is correct or not.

$$\theta_F \leftarrow \text{argmin}_\theta \mathcal{L}(\textbf{N-XKT}_{\theta_{K+Q}}(q^S, a^S), l^S) \quad (5)$$

As shown in equation 6, we adopt the same configuration as in the zero-shot setting, where the only difference is represented by the fine-tuned parameters set θ_F:

$$l^T_{pred} = \textbf{N-XKT}_{\theta_F}(q^T, c^T) \quad (6)$$

4 Empirical Evaluation

We conduct our experiments on four widely used science QA datasets, WorldTree V2.0 (Xie et al., 2020), ARC Easy and Challenge (Clark et al., 2018), and Openbook QA (Mihaylov et al., 2018). The results tend to confirm our research hypothesis that explanatory knowledge encoding can improve generalization in downstream science Question Answering (QA) tasks. Furthermore, we systematically analyze several factors which may have an impact on the final results, including the use of Transformer-based models with a larger number of parameters (BERT-large), testing the model on QA tasks using different types of explanatory background knowledge, and measuring training and test performance by further fine-tuning the model on other datasets.

4.1 Experimental Setup

QA dataset size. In order to conduct a thorough quantitative analysis, we use four science QA datasets, WorldTree V2.0 (Xie et al., 2020), ARC Easy and Challenge (Clark et al., 2018), and Openbook QA (Mihaylov et al., 2018). The number of question-answer pairs in each dataset is listed in Table. 1.

Table 1: QA datasets size.

Dataset	#Train	#Dev	#Test
WorldTree V2.0	3,947	1,019	4,165
ARC Easy	2,251	570	2,376
ARC Challenge	1,119	299	1,172
Openbook QA	4,957	500	500

Table 2: Number of instances in each explanatory knowledge category.

Type	Size
All	9,701
Retrieval	7,006
Inference-supporting	1,670
Complex Inference	1,025

Explanatory knowledge dataset size. We encode different types of explanatory knowledge in the WorldTree corpus into Transformer neural networks. The statistics of the adopted explanatory facts are reported in Table 2. Because we further analyze the impact of different types of knowledge, the number of each knowledge type is also given in the table.

Hyperparameters configuration. We adjust two major hyperparameters for the training of the model, namely batch size and learning rate. We optimize the parameters considering the following combinations: we adopt training batch sizes in $\{16, 32\}$, and learning rate in $\{1e-5, 3e-5, 5e-5\}$. The best results are obtained with batch size 32 and learning rate $3e-5$ for the BERT-base model, and batch size 16 and learning rate $1e-5$ for BERT-large (Devlin et al., 2019).

Information Retrieval baseline. We adopt an Information Retrieval (IR) baseline similar to the one described in Clark et al. (2018). Given a question q, for each candidate answer $c_i \in C = \{c_1, \ldots, c_n\}$, the IR solver uses BM25 vectors and cosine similarity to retrieve the top K sentences in the WorldTree corpus that are most similar to the concatenation of q and c_i. The score of a candidate answer c_i is then obtained by considering the sum of the BM25 relevance scores associated to

[2] https://allenai.org/data/arc
[3] https://allenai.org/data/open-book-qa

Table 3: N-XKT Question Answering accuracy results.

Config	Explanation Bank Dev	Explanation Bank Test	ARC Easy Dev	ARC Easy Test	ARC Challenge Dev	ARC Challenge Test	Openbook QA Dev	Openbook QA Test
IR BM25 (K = 5)	50.29%	44.55%	54.56%	50.00%	37.46%	31.14%	24.80%	26.80%
K base	49.30%	44.74%	50.18%	50.89%	34.38%	33.17%	30.96%	32.72%
Q base	44.86%	40.34%	50.81%	47.43%	24.41%	26.86%	27.92%	33.12%
K+Q base	58.14%	50.42%	58.53%	57.98%	37.46%	35.87%	35.32%	37.60%
K large	51.62%	45.85%	52.81%	52.58%	37.53%	33.07%	31.72%	34.12%
Q large	47.54%	43.47%	53.61%	51.41%	27.09%	28.63%	28.24%	36.04%
K+Q large	**60.16%**	**50.98%**	**61.19%**	58.24%	**39.00%**	37.63%	35.64%	38.20%
base FT	-	-	53.61%	53.82%	36.72%	32.71%	53.64%	53.16%
K base FT	-	-	53.61%	52.81%	35.79%	34.90%	53.60%	54.60%
Q base FT	-	-	59.05%	58.44%	33.65%	35.09%	56.04%	57.08%
K+Q base FT			59.33%	**58.79%**	38.13%	**38.09%**	**56.12%**	**56.56%**

Table 4: Question Answering accuracy results using different explanatory knowledge categories.

Knowledge	Config	Explanation Bank Dev	Explanation Bank Test	ARC Easy Dev	ARC Easy Test	ARC Challenge Dev	ARC Challenge Test	Openbook QA Dev	Openbook QA Test
None	Q base	44.86%	40.34%	50.81%	47.43%	24.41%	26.86%	27.92%	33.12%
Retrieval	K base	39.05%	38.72%	44.42%	45.25%	23.75%	26.25%	27.12%	29.96%
	K+Q base	51.00%	46.08%	51.79%	53.22%	34.65%	33.00%	31.96%	32.96%
Inference-supporting	K base	41.60%	38.24%	45.96%	44.77%	26.09%	26.02%	27.40%	30.88%
	K+Q base	52.72%	47.33%	54.35%	54.32%	34.85%	34.40%	33.64%	37.16%
Complex inference	K base	41.01%	38.58%	46.32%	45.98%	24.95%	23.75%	26.96%	29.76%
	K+Q base	52.99%	46.12%	55.30%	52.74%	34.78%	34.51%	32.08%	35.08%
All	K base	49.30%	44.74%	50.18%	50.89%	34.38%	33.17%	30.96%	32.72%
	K+Q base	**58.14%**	**50.42%**	**58.53%**	**57.98%**	**37.46%**	**35.87%**	**35.32%**	**37.60%**

the retrieved sentences. The predicted answer corresponds to the candidate choice with the highest score. To test the generalisation of this approach on ARC and OpenbookQA, we keep the same background knowledge throughout the experiments.

Configuration Setting. We adopt different configurations in the experiments to control for training data, Transformer model, and target QA test dataset fine-tuning. We report the different configurations in the "Config" column of Table 6 and Table 7. The label "K" indicates that the model is trained only on the explanatory knowledge acquisition task, "Q" means that the model is trained only on the cloze-style QA task using WorldTree as reference dataset, "K+Q" means that the model is pre-trained for explanatory knowledge acquisition and then further fine-tuned on cloze-style QA (again using only WorldTree as training dataset). Moreover, "base" means using BERT-base as Transformer model, while "large" means using BERT-large. Finally, "FT" means that the model is additionally fine-tuned on the target QA dataset's training data.

4.2 Overall Results on Zero-shot Science Question Answering

In Table 6, we report the performance of N-XKT under different configurations along with the accuracy of the BM25 baseline with $K = 5$ number of facts. The models are tested across multiple QA datasets including WorldTree, ARC, and OpenbookQA.

From the results, we derive the following conclusions. First, the proposed N-XKT model can clearly achieve better accuracy than the BM25 baseline since N-XKT uses Transformer-based neural mechanisms to acquire and encode external knowledge. Second, using BERT-large instead of BERT-base as initial Transformer can improve the performance since BERT-large contains more parameters than BERT-base. However, we found that the advantage of using BERT-large is not significant since more parameters implies more resources needed for training. Third, we observe than N-XKT obtains better performance than pre-trained BERT when fine-tuning on the target datasets.

Table 5: Accuracy comparison between N-XKT and othe approaches. External KB adopted by the models: 1.ARC-corpus (Clark et al., 2018), 2.ConceptNet (Speer et al., 2017), 3.Wikipedia (https://www.wikipedia.org/), 4.SciTail (Khot et al., 2018), 5.SNLI (Bowman et al., 2015), 6.MultiNLI (Williams et al., 2018), 7.RACE (Lai et al., 2017), 8.MCScript (Ostermann et al., 2018), 9.WorldTree (Jansen et al., 2018).

	ARC Easy	ARC Challenge	Openbook QA	External KB	IR-based	Fine-tuned
IR BM25 (K = 5)	50.00%	31.14%	26.80%	9	yes	no
Clark et al. (2018)	62.60%	20.30%	-	1	yes	yes
Mihaylov et al. (2018)	-	-	50.20%	2, 3	yes	yes
Khot et al. (2018)	59.00%	27.10%	24.40%	4	yes	yes
Zhang et al. (2018)	-	31.70%	-	1	no	yes
Yadav et al. (2018)	58.40%	26.60%	-	none	no	yes
Musa et al. (2019)	52.20%	33.20%	-	1	yes	yes
Zhong et al. (2019)	-	33.40%	-	2	no	yes
Pîrtoacă et al. (2019)	61.10%	26.90%	-	4, 5, 6	no	yes
Ni et al. (2019)	-	36.60%	-	7, 8	no	yes
GPT^{II}(Radford, 2018)	57.00%	38.20%	52.00%	7	no	yes
RS^{II}(Sun et al., 2019)	66.60%	40.70%	55.20%	7	no	yes
N-XKT K+Q base (ours)	**57.98%**	**35.87%**	**37.60%**	9	no	no

4.3 Ablation Analysis on Impact of Different Explanatory Knowledge Types

To understand the impact of different types of explanation on the final accuracy, we breakdown the facts stored in the knowledge base using three different categories (i.e., retrieval, inference-supporting and complex inference) and rerun the training of the N-XKT model using only one category per time.

The adopted categories are provided in the WorldTree corpus and can be described as follows:

- *Retrieval*: facts expressing knowledge about taxonomic relations and/or properties.
- *Inference-Supporting*: Facts expressing knowledge about actions, affordances, requirements.
- *Complex Inference*: Facts expressing knowledge about causality, processes, and if/then relationships.

The obtained accuracy is showed in Table 7. The results highlight the importance of using all the explanation categories to achieve the final accuracy for the combined approach. However, the retrieval category seems to have a higher impact on the generalisation. We believe that this result is due to the taxonomic knowledge encoded in the retrieval category (i.e. *"x is a kind of y"*), which facilitates the acquisition of the implicit explanatory capabilities necessary for answering science questions.

In Table 7, we compare the impact of different explanatory knowledge types and get the following conclusion. 1) All three types of explanatory knowledge are helpful for further science QA task. The results using all three types of knowledge are significantly better than the results obtained when using no explanatory knowledge at all (first rown in Table 7). 2) The model trained on all explanatory knowledge outperforms the models using each individual type of knowledge alone, confirming that different types of knowledge are complementary for achieving the final performance.

4.4 Evaluating Zero-shot N-XKT with Start-of-the-art baselines

In Table 5, we evaluate several start-of-the-art methods as baselines along with N-XKT trained only on the WorldTree. The table reports the accuracy results on ARC and OpenbookQA. In the "External KB" column, we list the external Knowledge Bases (KB) adopted by different models. The "IR-based" column indicates whether the model adopts Information Retrieval (IR) techniques, and the "Fine-tuned" column indicates whether the approach is fine-tuned on the target dataset.

Table 5 is intended to provide a general comparative analysis between N-XKT and the baseline models, most of them fine-tuned on the target datasets. N-XKT is able to achieve comparable performance under a transfer learning setting. The generalization performance of the proposed model is more noticeable for the ARC Challenge dataset, which requires the implicit encoding of more complex explanatory knowledge.

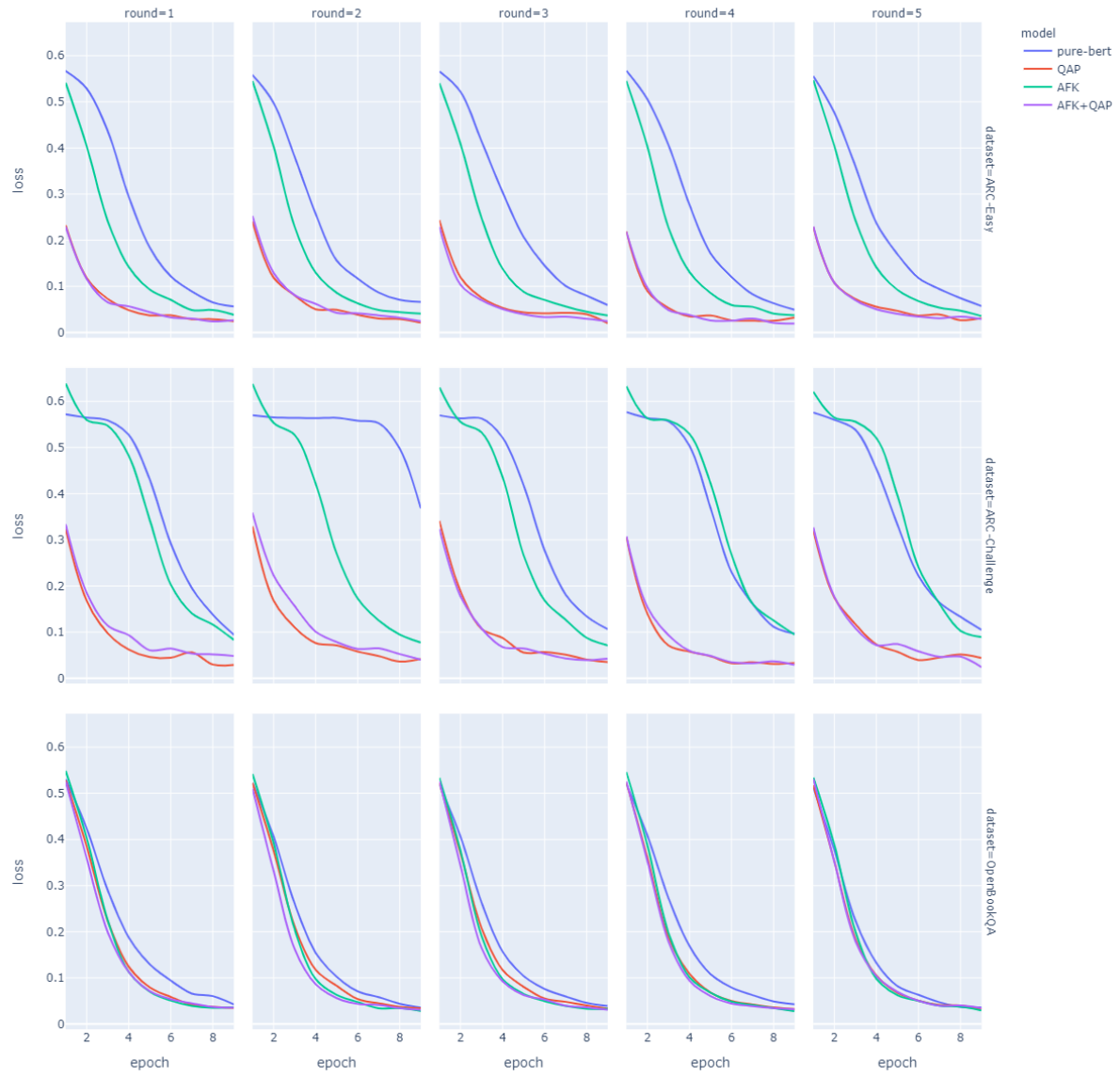


Figure 2: Convergence curve when fine-tuning different version of N-XTK on the target QA datasets.

4.5 Improvement on Fine-tuning Convergence

In Figure 2, we visualize the convergence curve for the fine-tuning over three science QA tasks (ARC Easy, ARC Challenge and OpenBookQA), comparing a pure BERT-based N-XKT model with a pre-trained N-XKT models using different configurations, AFK (pre-trained on explanatory knowledge acquisition), QAP (pre-trained on WorldTree cloze-style QA), AFK+QAP (pre-trained on both). It is noticeable that the encoding of explanatory knowledge impacts the convergence of the model for all three datasets, with a particular emphasis on the two ARC variants.

5 Conclusion

In this paper, we proposed a neural encoding mechanism for explanatory knowledge acquisition and transfer, N-XKT. We evaluated the impact of the encoding mechanism on downstream science QA. The proposed model delivers better generalisation and accuracy for QA tasks that require multi-hop and explanatory inference. The proposed encoding mechanism can be used to deliver zero-shot inference capabilities, providing comparable performance when compared to supervised models on QA. These results supports the hypothesis that pre-training tasks targeting abstract and explanatory knowledge acquisition can constitute and impor-

tant direction to improve inference capabilities and generalization of state-of-the-art neural models.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific 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 3615–3620, Hong Kong, China. 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.

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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, and Michael Schmitz. 2020. From 'f' to 'a' on the n.y. regents science exams: An overview of the aristo project. *AI Magazine*, 41(4):39–53.

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.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Jansen, Niranjan Balasubramanian, Mihai Surdeanu, and Peter Clark. 2016. What's in an explanation? characterizing knowledge and inference requirements for elementary science exams. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2956–2965, Osaka, Japan. The COLING 2016 Organizing Committee.

Peter Jansen and Dmitry Ustalov. 2019. TextGraphs 2019 shared task on multi-hop inference for explanation regeneration. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 63–77, Hong Kong. Association for Computational Linguistics.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and D. Roth. 2018. Question answering as global reasoning over semantic abstractions. In *AAAI*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 311–316, Vancouver, Canada. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Souvik Kundu, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2019. Exploiting explicit paths for multi-hop reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2737–2747, Florence, Italy. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Yang Li and Peter Clark. 2015. Answering elementary science questions by constructing coherent scenes using background knowledge. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2007–2012, Lisbon, Portugal. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Ryan Musa, Xiaoyan Wang, Achille Fokoue, Nicholas Mattei, Maria Chang, Pavan Kapanipathi, Bassem Makni, Kartik Talamadupula, and Michael Witbrock. 2019. Answering science exam questions using query reformulation with background knowledge. In *Automated Knowledge Base Construction (AKBC)*.

Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. 2019. Learning to attend on essential terms: An enhanced retriever-reader model for open-domain question answering. 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 335–344, Minneapolis, Minnesota. Association for Computational Linguistics.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. MCScript: A novel dataset for assessing machine comprehension using script knowledge.

George-Sebastian Pîrtoacă, Traian Rebedea, and Ștefan Rușeți. 2019. Improving retrieval-based question answering with deep inference models. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, Florence, Italy. Association for Computational Linguistics.

A. Radford. 2018. Improving language understanding by generative pre-training.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Mrinmaya Sachan, Kumar Dubey, and Eric Xing. 2016. Science question answering using instructional materials. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 467–473, Berlin, Germany. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving machine reading comprehension with general reading strategies. 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 2633–2643, Minneapolis, Minnesota. Association for Computational Linguistics.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A survey on explainability in machine reading comprehension.

Mokanarangan Thayaparan, Marco Valentino, Viktor Schlegel, and André Freitas. 2019. Identifying supporting facts for multi-hop question answering with document graph networks. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 42–51, Hong Kong. Association for Computational Linguistics.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021. Unification-based reconstruction of multi-hop explanations for science questions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 200–211, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018. Yuanfudao at SemEval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. pages 758–762.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press.

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.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5456–5473, Marseille, France. European Language Resources Association.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. 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 2578–2589, Hong Kong, China. Association for Computational Linguistics.

Vikas Yadav, Rebecca Sharp, and M. Surdeanu. 2018. Sanity check: A strong alignment and information retrieval baseline for question answering. *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Y. Zhang, H. Dai, Kamil Toraman, and L. Song. 2018. Kg2: Learning to reason science exam questions with contextual knowledge graph embeddings. *ArXiv*, abs/1805.12393.

Wanjun Zhong, Duyu Tang, Nan Duan, M. Zhou, Jiahai Wang, and J. Yin. 2019. Improving question answering by commonsense-based pre-training. In *NLPCC*.

A Hyperparameters tuning

The N-XKT mainly use a transformer network as natural language encoder component, the hyperparameters of transformer network training have been tuned manually for the optimisation is the maximisation of the accuracy in answer prediction. Specifically, 3 parameters should be set for training, train batch size β, learning rate α, and train epoch $\mathcal{N}$. The values used in pre-training on explanation knowledge base are as follows:

- β = 32
- α = 5e-5
- $\mathcal{N}$ = 5

The values used in fine-tuning on Question Answer are as follows:

- β = 32
- α = 5e-5
- $\mathcal{N}$ = 40

B Data

We use two versions of Explanation Bank Scientific Question Answer datasets in this paper. The version 1 of Explanation Bank dataset can be downloaded at the following URL: `http://cognitiveai.org/dist/worldtree_corpus_textgraphs2019sharedtask_withgraphvis.zip`. The version 2 of Explanation Bank dataset is available at the following URL: `https://github.com/cognitiveailab/tg2020task`.

C Computing Infrastructure

To accelerate the training process of the experiments, we adopt a NVIDIA Tesla V100 GPU.

D Accuracy Results Including Standard Deviation

We repeat the N-XKT model Question Answering training process on all the dataset for 5 times, each time with random parameters initialization. Addition to the tables provided in paper, we report the detailed results with standard deviation in following tables.

Table 6: N-XKT Question Answering accuracy result comparison

Config	Explanation Bank		ARC Easy		ARC Challenge		Openbook QA	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
IR BM25 (K = 5)	50.29%	44.55%	54.56%	50.00%	37.46%	31.14%	24.80%	26.80%
K base	49.30% ±0.0238	44.74% ±0.0166	50.18% ±0.0167	50.89% ±0.0198	34.38% ±0.0255	33.17% ±0.0165	30.96% ±0.0359	32.72% ±0.0273
Q base	44.86% ±0.0229	40.34% ±0.0087	50.81% ±0.0258	47.43% ±0.0136	24.41% ±0.0101	26.86% ±0.0049	27.92% ±0.0342	33.12% ±0.0176
K+Q base	58.14% ±0.0119	**50.42%** **±0.0039**	58.53% ±0.0047	**57.98%** **±0.0014**	37.46% ±0.0135	**35.87%** **±0.0149**	35.32% ±0.0124	**37.60%** **±0.0085**
K large	51.62% ±0.0159	45.85% ±0.0089	52.81% ±0.004	52.58% ±0.0136	37.53% ±0.0109	33.07% ±0.0129	31.72% ±0.0199	34.12% ±0.0232
Q large	47.54% ±0.0131	43.47% ±0.0061	53.61% ±0.0176	51.41% ±0.0073	27.09% ±0.012	28.63% ±0.0125	28.24% ±0.0118	36.04% ±0.0167
K+Q large	60.16% ±0.0168	50.98% ±0.0102	61.19% ±0.0108	58.24% ±0.0076	39.00% ±0.0268	37.63% ±0.0155	35.64% ±0.0076	38.20% ±0.0161
base FT	- -	- -	53.61% ±0.0168	53.82% ±0.0093	36.72% ±0.0104	32.71% ±0.0086	53.64% ±0.0182	53.16% ±0.0223
K base FT	- -	- -	53.61% ±0.0159	52.81% ±0.0241	35.79% ±0.0218	34.90% ±0.0239	53.60% ±0.0248	54.60% ±0.0281
Q base FT	- -	- -	59.05% ±0.0177	58.44% ±0.0070	33.65% ±0.0280	35.09% ±0.0065	56.04% ±0.0126	57.08% ±0.0178
K+Q base FT	-	-	59.33% ±0.0187	**58.79%** **±0.0087**	38.13% ±0.0224	**38.09%** **±0.0124**	56.12% ±0.0186	**56.56%** **±0.0111**

Table 7: Question Answering accuracy result in different abstractive knowledge categories

Knowledge	Config	Explanation Bank		ARC Easy		ARC Challenge		Openbook QA	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
None	Q base	44.86% ±0.0229	40.34% ±0.0087	50.81% ±0.0258	47.43% ±0.0136	24.41% ±0.0101	26.86% ±0.0049	27.92% ±0.0342	33.12% ±0.0176
RET	K base	39.05% ±0.0258	**38.72%** **±0.0106**	44.42% ±0.011	**45.25%** **±0.0139**	23.75% ±0.0165	**26.25%** **±0.0141**	27.12% ±0.0099	**29.96%** **±0.0202**
	K+Q base	51.00% ±0.0173	**46.08%** **±0.0135**	51.79% ±0.0178	**53.22%** **±0.0141**	34.65% ±0.0321	**33.00%** **±0.0128**	31.96% ±0.0192	**32.96%** **±0.0182**
INSUPP	K base	41.60% ±0.0149	**38.24%** **±0.0075**	45.96% ±0.0127	**44.77%** **±0.0118**	26.09% ±0.0164	**26.02%** **±0.0099**	27.40% ±0.0168	**30.88%** **±0.0122**
	K+Q base	52.72% ±0.0247	**47.33%** **±0.0062**	54.35% ±0.0206	**54.32%** **±0.0092**	34.85% ±0.031	**34.40%** **±0.0128**	33.64% ±0.0279	**37.16%** **±0.0306**
COMPLEX	K base	41.01% ±0.0132	**38.58%** **±0.0035**	46.32% ±0.0134	**45.98%** **±0.0091**	24.95% ±0.0263	**23.75%** **±0.0066**	26.96% ±0.012	**29.76%** **±0.0163**
	K+Q base	52.99% ±0.0098	**46.12%** **±0.0131**	55.30% ±0.0081	**52.74%** **±0.0087**	34.78% ±0.0112	**34.51%** **±0.0194**	32.08% ±0.018	**35.08%** **±0.0153**
All	K base	49.30% ±0.0238	44.74% ±0.0166	50.18% ±0.0167	50.89% ±0.0198	34.38% ±0.0255	33.17% ±0.0165	30.96% ±0.0359	32.72% ±0.0273
	K+Q base	58.14% ±0.0119	50.42% ±0.0039	58.53% ±0.0047	57.98% ±0.0014	37.46% ±0.0135	35.87% ±0.0149	35.32% ±0.0124	37.60% ±0.0085

Tab. 6 is for overall accuracy of N-XKT model on QA tasks, and Tab. 7 is for ablation analysis results, only use part of explanations in training process.

Predicate Representations and Polysemy in VerbNet Semantic Parsing

James Gung*
University of Colorado Boulder
james.gung@colorado.edu

Martha Palmer
University of Colorado Boulder
martha.palmer@colorado.edu

Abstract

Despite recent advances in semantic role labeling propelled by pre-trained text encoders like BERT, performance lags behind when applied to predicates observed infrequently during training or to sentences in new domains. In this work, we investigate how semantic role labeling performance on low-frequency predicates and out-of-domain data can be improved by using VerbNet, a verb lexicon that groups verbs into hierarchical classes based on shared syntactic and semantic behavior and defines semantic representations describing relations between arguments. We find that VerbNet classes provide an effective level of abstraction, improving generalization on low-frequency predicates by allowing them to learn from the training examples of other predicates belonging to the same class. We also find that joint training of VerbNet role labeling and predicate disambiguation of VerbNet classes for polysemous verbs leads to improvements in both tasks, naturally supporting the extraction of VerbNet's semantic representations.

1 Introduction

Semantic role labeling (SRL) is a form of shallow semantic parsing that involves the extraction of predicate arguments and their assignment to consistent roles with respect to the predicate, facilitating the labeling of e.g. *who* did *what* to *whom* (Gildea and Jurafsky, 2000). SRL systems have been broadly applied to applications such as question answering (Berant et al., 2014; Wang et al., 2015), machine translation (Liu and Gildea, 2010; Bazrafshan and Gildea, 2013), dialog systems (Tur and Hakkani-Tür, 2005; Chen et al., 2013), metaphor detection (Stowe et al., 2019), and clinical information extraction (Gung, 2013; MacAvaney et al., 2017). Recent approaches to SRL have achieved large gains in performance through the use of pre-trained text encoders like ELMo and BERT (Peters et al., 2018; Devlin et al., 2019). Despite these advances, performance on low-frequency predicates and out-of-domain data remains low relative to in-domain performance on higher frequency predicates.

	Billy	***consoled***	*the puppy*
PB	Arg0	console.01	Arg1
VN	Stimulus	amuse-31.1	Experiencer
	Billy	***walked***	*the puppy*
PB	Arg0	walk.01	Arg1
VN	Agent	run-51.3.2-2-1	Theme

Table 1: Comparison of PropBank (PB) and VerbNet (VN) roles for predicates *console* and *walk*. VerbNet's thematic role assignments (e.g. Stimulus vs. Agent and Experiencer vs. Theme) are more dependent on the predicate than PropBank's numbered arguments.

The assignment of role labels to a predicate's arguments is dependent upon the predicate's sense. PropBank (Palmer et al., 2005) divides each predicate into one or more *rolesets*, which are coarse-grained sense distinctions that each provide a set of core numbered arguments (A0-A5) and their corresponding definitions. VerbNet (VN) groups verbs into hierarchical *classes*, each class defining a set of valid syntactic frames that define a direct correspondence between thematic roles and syntactic realizations, e.g. *Agent REL Patient* (e.g. *John broke the vase*) or *Patient REL* (e.g. *The vase broke*) for *break-45.1* (Schuler, 2005).

Recent PropBank (PB) semantic role labeling models have largely eschewed explicit predicate disambiguation in favor of direct prediction of semantic roles in end-to-end trainable models (Zhou and Xu, 2015; He et al., 2017; Shi and Lin, 2019).

*Work done prior to joining Amazon.

Proceedings of the 14th International Conference on Computational Semantics, pages 51–62
June 17–18, 2021.

This is possible for several reasons: First, PropBank's core roles and modifiers are shared across all predicates, allowing a single classifier to be trained over tokens or spans. Second, although definitions of PB roles are specific to the different senses of each predicate, efforts are made when creating rolesets to ensure that A0 and A1 exhibit properties of Dowty's prototypical Agent and prototypical Patient respectively (1991). Finally, PB rolesets are defined based on VN class membership, with predicates in the same classes thus being assigned relatively consistent role definitions (Bonial et al., 2010).

Unlike PropBank, VerbNet's thematic roles are shared across predicates and classes with consistent definitions. However, VN roles are more dependent on the identity of the predicate (Zapirain et al., 2008; Merlo and Van Der Plas, 2009). Examples of PropBank and VerbNet roles illustrating this are given in Table 1. Consequently, VN role labeling models may benefit more from predicate features than PropBank. Furthermore, while it is possible to identify PB or VN roles without classifying predicate senses, linking the resulting roles to their definitions or to the syntactic frames and associated semantic primitives in VN does require explicit predicate disambiguation (Brown et al., 2019). Therefore, predicate disambiguation is often an essential step when applying SRL systems to real-world problems.

In this work, we evaluate alternative approaches for incorporating VerbNet classes in English VerbNet and PropBank role labeling. We propose a joint model for SRL and VN predicate disambiguation (VN classification), finding that joint training leads to improvements in VN classification and role labeling for out-of-domain predicates. We also evaluate VN classes as predicate-specific features. Using gold classes, we observe significant improvements in both PB and VN SRL. We also observe improvements in VN role labeling when using predicted classes and features that incorporate all valid classes for each predicate[1].

2 Background and Related Work

VerbNet VerbNet is a broad-coverage lexicon that groups verbs into hierarchical classes based on shared syntactic and semantic behavior (Schuler, 2005). Each VN class is assigned a set of thematic roles that, unlike PB numbered arguments, maintain consistent meanings across different verbs and classes. VN classes provide an enumeration of syntactic frames applicable to each member verb, describing how the thematic roles of a VN class may be realized in a sentence. Every syntactic frame entails a set of low-level semantic representations (primitives) that describe relations between thematic role arguments as well as changes throughout the course of the event (Brown et al., 2018). The close relationship between syntactic realizations and semantic representations facilitates straightforward extraction of VN semantic predicates given identification of a VN class and corresponding thematic roles. VN primitives have been applied to problems such as machine comprehension (Clark et al., 2018) and question generation (Dhole and Manning, 2020).

Comparing VerbNet with PropBank Yi et al. (2007) use VN role groupings to improve label consistency across verbs by reducing the overloading of PropBank's numbered arguments like A2. Comparing SRL models trained on PB and VN, Zapirain et al. (2008) find that their VerbNet model performs worse on infrequent predicates than their PB model, and suggest that VN is more reliant on the identity of the predicate than PB based on experiments removing predicate-specific features from their models. They suggest that the high consistency of A0 and A1 enables PB to generalize better without relying on predicate-specific information.

Merlo and Van Der Plas (2009) provide an information-theoretic perspective on the comparison of PropBank and VerbNet, demonstrating how the identity of the predicate is more important to VN SRL than for PB by comparing the conditional entropy of roles given verbs as well as the mutual information of roles and verbs. In multilingual BERT probing studies comparing several SRL formalisms, Kuznetsov and Gurevych (2020) find that layer utilization for predicates differs between PB and VN. PB emphasizes the same layers used for syntactic tasks, while VN uses layers associated with tasks used more prevalently in lexical tasks. These findings reinforce the importance of predicate representations to VerbNet.

SRL and Predicate Disambiguation Previous work has investigated the interplay between predicate sense disambiguation and SRL. Dang and Palmer (2005) improve verb sense disambiguation

[1]Our code is available at https://github.com/jgung/verbnet-parsing-iwcs-2021.

(VSD) using features based on semantic role labels. Moreda and Palomar (2006) find that explicit verb senses improve PB SRL for verb-specific roles like A2 and A3, but hurt on adjuncts. Yi (2007) find that using gold standard PB roleset IDs as features in an SRL model improves performance only on highly polysemous verbs. Dahlmeier et al. (2009) propose a joint probabilistic model for *preposition* disambiguation and SRL, finding an improvement over independent models.

Predicate disambiguation plays a critical role in FrameNet (Baker et al., 1998) parsing, in part because FrameNet's role inventory is more than an order of magnitude larger than that of PB and VN. This richer, more granular role inventory lends advantages to approaches that constrain role identification to the set of valid roles for the predicted frame (Das et al., 2014; Hermann et al., 2014), or that jointly encode argument and role representations given identified frames (FitzGerald et al., 2015).

LM Pre-training and SRL Language model (LM) pre-training has become ubiquitous in natural language processing tasks, with LM encoders like ELMo propelling forward the state of the art in SRL (Peters et al., 2018). We are interested in whether a strong baseline model using a LM encoder such as BERT can be further improved by incorporating external knowledge from lexical resources like VN.

BERT (Devlin et al., 2019) is a Transformer encoder (Vaswani et al., 2017) jointly trained using two objectives: a masked language modeling objective to predict the identity of randomly-masked tokens in the input, as well as a next sentence prediction task (NSP) intended to encourage the model to encode the relationship between sentence pairs (henceforth referred to as *Sent. A* and *Sent. B*). Sentences are tokenized using WordPiece (Wu et al., 2016). As a Transformer encoder, BERT applies multiple layers of a multi-headed self-attention mechanism to progressively build contextual token-level representations. In our experiments, we use encodings from the final layer.

3 Semantic Role Labeling with BERT

Our baseline SRL model closely follows Shi and Lin (2019). We thus approach SRL as a sequence tagging task, predicting per-word, IOB-encoded (**I**n, **O**ut, **B**egin) role labels independently for each predicate in a sentence. A predicate-aware encoding of a sentence is produced using the target predicate as the *Sent. B* input to BERT. For example, the sentence *I tried opening it* is processed as:

CLS I tried opening it SEP opening SEP

for the verb *open*. This enables BERT to incorporate the identity of the predicate in the encoding of each word while clearly delineating it from tokens in the original sentence.

To simplify notation, we'll treat $\mathbf{LM}(\mathbf{a}, \mathbf{b}) \in \mathbb{R}^{T_a \times D_{\text{LM}}}$ as shorthand for the final layer BERT encoding for a pair of sentences $\mathbf{a} = w_1, ..., w_{T_a}$ and $\mathbf{b} = w_1, ..., w_{T_b}$ with T_a and T_b words respectively, where D_{LM} gives BERT's hidden size. This is produced by applying WordPiece tokenization (**WP**) to each word in each sentence and concatenating the resulting sequences of token IDs with standard BERT-specific IDs:

$$\mathbf{w} = \left[\text{CLS}, \mathbf{WP}(\mathbf{a}), \text{SEP}, \mathbf{WP}(\mathbf{b}), \text{SEP}\right]$$

The resulting sequence of tokens $\mathbf{w}$ is encoded using BERT. We use the final layer outputs, taking vectors only for the first WordPiece token for each original word in *Sent. A* ($\mathbf{a}$), filtering out vectors corresponding to *Sent. B* ($\mathbf{b}$), SEP or CLS. The resulting matrix consists of a vector per word in *Sent. A*, avoiding any discrepancies between IOB-encoded word-level output labels and WordPiece tokens used as inputs.

Following previous work (Zhou and Xu, 2015; He et al., 2017), we use a marker feature as an indicator for the specific location of the predicate within the sentence. For a sentence, $w_1, ..., w_T$, with a predicate given by index $p \in 1...T$, we compute a predicate-aware, contextualized embedding $\mathbf{x}_{pt}$ of each word as

$$\mathbf{x}_{pt} = \left[\mathbf{LM}(w_{1...T}, w_p)_{(t)}; \mathbf{W}^{(mark)}_{(t=p)}\right] \quad (1)$$

with $\mathbf{W}^{(mark)} \in \mathbb{R}^{2 \times D_{mark}}$ and $\mathbf{x}_{pt} \in \mathbb{R}^{D_{\text{LM}}+D_{mark}}$, where D_{mark} provides the size of the predicate marker embedding.

The predicate's positional information from the marker is integrated using a bidirectional LSTM (Hochreiter and Schmidhuber, 1997), concatenating the hidden states for the forward and backward LSTMs at each timestep (omitting the p from $\mathbf{x}_{pt}$ for brevity):

$$\begin{aligned} \mathbf{h}^{(fw)}_t &= \text{LSTM}^{(fw)}(\mathbf{x}_{1...T})_{(t)} \\ \mathbf{h}^{(bw)}_t &= \text{LSTM}^{(bw)}(\mathbf{x}_{T...1})_{(T-t)} \\ \mathbf{h}^{(fb)}_t &= \left[\mathbf{h}^{(fw)}_t; \mathbf{h}^{(bw)}_t\right] \end{aligned} \quad (2)$$

The BiLSTM output at each timestep t is concatenated with that of the predicate's timestep and passed through a sequentially-applied linear transformation followed by a leaky ReLu ($\alpha = 0.1$):

$$\mathbf{x}_{pt}^{(mlp)} = \sigma\left(\mathbf{W}^{(mlp)}\left[\mathbf{h}_t^{(fb)}; \mathbf{h}_p^{(fb)}\right] + \mathbf{b}^{(mlp)}\right) \tag{3}$$

We apply a final linear projection from $\mathbf{x}_{pt}^{(mlp)}$ to IOB-encoded role labels:

$$\mathbf{s}_{pt}^{(srl)} = \mathbf{W}^{(srl)}\mathbf{x}_{pt}^{(mlp)} + \mathbf{b}^{(srl)} \tag{4}$$

where $\mathbf{s}_{pt}^{(srl)} \in \mathbb{R}^K$ provides the unnormalized scores for each of K possible role labels, with the probability of predicting a label for a given token t and predicate p given by:

$$P(y_{pt}^{(srl)}|w_{1...T}, w_p) = \text{softmax}(\mathbf{s}_{pt}^{(srl)}) \tag{5}$$

Like He et al. (2017), we apply constrained Viterbi decoding to restrict inferred label sequences to produce valid IOB sequences.

4 VerbNet Classes as Predicate Features

Verbs belonging to the same VN class share syntactic and semantic properties and the same set of thematic roles and syntactic frames. Replacing a predicate in a sentence with a different verb from the same class typically produces a syntactically coherent sentence and does not impact the proposition's thematic role labels. VN classes may thus provide an effective level of abstraction for predicates in SRL.

We hypothesize that using VN classes as predicate-specific features may help reduce sparsity issues for low-frequency and out-of-vocabulary (OOV) verbs. Intuitively, training examples for each member verb within a class contribute to the estimation of parameters associated with all other members of the same class, enabling the fine-tuning of predicate-level features even for OOV predicates. For example, a verb like *traipse* may rarely or never occur during training, but may belong to a class which appears hundreds of times in the form of more common verbs like *run* or *rush*. We investigate whether by sharing parameter updates across VN members, we can further improve generalization on infrequent verbs.

Methodology Intuitively, BERT's NSP pretraining task encourages some level of focus on *Sent. B* tokens from attention heads when processing tokens in *Sent. A*. The predicate feature presented by Shi and Lin (2019) and applied in our baseline model uses the predicate token as the *Sent. B* input to BERT and thus allows the encodings of tokens in a sentence to be conditioned directly on the predicate.

We propose to include tokens corresponding to the predicate's VN class as additional features as part of *Sent. B*. To realize this, we concatenate the corresponding VN class ID to *Sent. B* along with the predicate, updating the inputs given in Equation 1:

$$\mathbf{LM}(w_{1...T}, w_p w_s) \tag{6}$$

where w_s is a token corresponding to the VN class of the predicate w_p[2].

VerbNet Classification VN classes can be predicted automatically using a word sense disambiguation system. We propose a simple model for VerbNet classification: fine tune a pre-trained BERT encoder by applying a feedforward multilayer perceptron (MLP) classifier over all VN classes to the BERT encoding associated with the first WordPiece of the target predicate.

We again condition BERT on the target predicate by including it as a feature (w_p) in *Sent. B*:

$$\begin{aligned} \mathbf{x}_p &= \mathbf{LM}(w_{1...T}, w_p)_{(p)} \\ \mathbf{x}_p^{(mlp)} &= \sigma\left(\mathbf{W}^{(mlp)}\mathbf{x}_p + \mathbf{b}^{(mlp)}\right) \\ \mathbf{s}_p^{(vncls)} &= \mathbf{W}^{(vncls)}\mathbf{x}_p^{(mlp)} + \mathbf{b}^{(vncls)} \end{aligned} \tag{7}$$

where $\mathbf{W}^{(vncls)} \in \mathbb{R}^{D_{mlp} \times V}$ projects over all V VN classes for all predicates. The probability for predicting a VN class $y_p^{(vncls)}$ for a given predicate and sentence is given by:

$$P(y_p^{(vncls)}|w_{1...T}, w_p) = \text{softmax}(\mathbf{s}_p^{(vncls)}) \tag{8}$$

This *single classifier* formulation is possible for lexicons like VN and FrameNet in which predicates share senses from a global sense inventory. While individual predicates have a specific set of valid senses, their senses are shared from the global lexicon. Kawahara and Palmer (2014) demonstrate

[2]In preliminary experiments, we found that directly modifying *Sent. A* drastically reduces the performance of the model and slows convergence.

that a single classifier approach to VN classification achieves competitive performance when using shared semantic features. Intuitively, by training the classifier across multiple verbs, the model parameters specific to each sense receive more updates, with infrequent verb-class pairs also benefiting from the examples of other verbs within the same class. At inference time, we constrain sense predictions to predicate-sense combinations observed in the training data, selecting the highest-scoring valid sense given the predicate. We evaluate models using both predicted and gold (ground truth) classes for w_s as PREDICTED CLASS and GOLD CLASS respectively.

VerbNet Classes without Disambiguation Like SRL, VerbNet classification accuracy declines in the long tail of low frequency senses and predicates. For this reason, incorrect sense predictions may negate the benefits of VN class features on precisely the instances for which they might be expected to be beneficial: OOV or rare predicates.

To avoid this problem while still retaining the benefits of parameter sharing for low frequency predicates with higher-frequency predicates belonging to the same VN class, we propose including the set of all possible classes for a given predicate as *Sent. B* features. To incorporate multiple senses, we simply concatenate them sequentially to *Sent. B*:

$$\mathbf{LM}(w_{1...T}, w_p w_{s_{1...k}}) \tag{9}$$

This allows the BERT encoder to attend over all possible VerbNet classes for a given predicate and sentence, without making a discrete decision about which class is correct. The extent and way in which the model incorporates the *Sent. B* tokens associated with the available classes is learned during training. The inputs to this model, later referred to as ALL CLASSES are identical to PREDICTED CLASS and GOLD CLASS models for monosemous predicates.

5 Joint VerbNet Classification and SRL

Features that are useful for SRL may also be useful in predicting the sense of a predicate. For example, surface-level syntactic awareness that the argument of a predicate is a clause instead of a noun phrase may change the expected sense of a verb (bring-11.3 vs. characterize-29.2):

Bob *took* Mary to the doctor.
John *took* Mary to be a doctor.

The semantic classes of arguments are also often important in determining the sense of a given predicate (dub-29.3.2 vs. get-13.5.1):

John *called* Mary a name.
John *called* Mary a car.

This dependency between SRL and predicate sense disambiguation together with the prevalence of shared features between the two tasks makes them a good candidate for multi-task learning (Caruana, 1998).

Multi-task Model Much of recent work in multi-task learning for SRL has focused on syntactic tasks such as syntactic parsing as auxiliary objectives (Strubell et al., 2018; Swayamdipta et al., 2018; Xia et al., 2019; Zhou et al., 2020). We first investigate an MTL approach that predicts semantic role labels and predicate senses independently given a shared BERT encoder. We extend our baseline SRL model, adding an additional *head* that is trained to predict the target predicate's sense, as described in Equation 8. The negative log likelihood of a single training instance with predicate p and token sequence $\mathbf{x} = w_{1...T}$ with T tokens is then given by:

$$\begin{aligned} -\sum_{t=1}^{T} \Big[\log P(y_{pt}^{(srl)} | \mathbf{x}, p) \Big] \\ + \lambda_{vncls} \log P(y_p^{(vncls)} | \mathbf{x}, p) \end{aligned} \tag{10}$$

with λ_{vncls} weighting the contribution of VerbNet class prediction to the overall objective. For brevity, we henceforth refer to this model as SRL + VSD.

We also investigate conditioning role labeling directly on predicted predicate senses. We implement this by concatenating a weighted label embedding of the target predicate's predicted class to each of the SRL head's input vectors, $\mathbf{x}_{pt}^{(srl)}$. To compute the weighted label embedding of a given VN class $y_p^{(vncls)}$ we follow Hashimoto et al. (2017):

$$\mathbf{y}_p^{(vncls)} = \sum_{k=1}^{K} P(y_p^{(vncls)} = k | \mathbf{x}, p) \mathbf{W}_{(k)}^{(vncls)} \tag{11}$$

with $\mathbf{W}^{(vncls)} \in \mathbb{R}^{K \times D_{vncls}}$ and $\mathbf{y}_p^{(srl)} \in \mathbb{R}^{D_{vncls}}$. The input to the SRL head is then given by:

$$\begin{aligned} \mathbf{x}_{pt}^{(srl)} = \big[\mathbf{LM}(w_{1...T}, w_p)_{(t)}; \\ \mathbf{W}_{(t=p)}^{(mark)}; \mathbf{y}_p^{(vncls)} \big] \end{aligned} \tag{12}$$

VerbNet class embeddings are initialized using the average of word embeddings corresponding to members of each class. During training, we use embeddings of predicted labels to avoid a discrepancy between the inputs to the SRL head between training and inference, when the gold labels are no longer available. In preliminary experiments, we used gold labels, similar to *teacher forcing* as described in Williams and Zipser (1989), but found that performance degraded when applied to predicted labels. We refer to the model described in this section as SRL | VSD.

6 Experiments

All models are implemented using Tensorflow 1.13 (Abadi et al., 2016) and are trained on a single NVIDIA GTX 1080 Ti GPU. We use the 110M parameter cased BERT-Base model available in Tensorflow Hub[3], with $D_{\text{LM}} = 768$. To align with Shi and Lin (2019), D_{mark} is set to 10, and LSTM and MLP hidden state sizes are set to 768 and 300 respectively. Dropout rates of 0.1 are applied to BERT outputs as well as after ReLu transforms in MLPs. Recurrent dropout (Gal and Ghahramani, 2016) with a rate of 0.1 is applied in LSTMs on hidden states and outputs. To initialize VerbNet class embeddings, we use 100-dimensional GloVe embeddings (Pennington et al., 2014) averaged over member verbs ($D_{vncls} = 100$). λ_{vncls} is set to 0.5 after a preliminary search over $\{0.1, 0.5, 1.0\}$.

We follow the fine-tuning methodology described in Devlin et al. (2019), using Adam (Kingma and Ba, 2014) with a batch size of 16. The learning rate is warmed up linearly from 0 to 5e-5 for 10% of training, then decayed linearly to 0 for the rest of training. Models are trained for up to 8 epochs. The best-performing checkpoint on the development set, evaluated at every half epoch, is selected for evaluation.

Unless otherwise mentioned, we train and evaluate all models with at least 7 independent random initializations, and present mean scores in our comparisons. To establish statistical significance, we apply a test for *Almost Stochastic Dominance* (Dror et al., 2019) between test score distributions, using $\alpha = 0.05$. Numbers in bold indicate highest average performance within a given evaluative setting, with a single star indicating statistical significance of almost stochastic dominance over our baseline models for each experiment, and two stars indicating stochastic dominance ($\epsilon = 0$). For example, a value in a table of $\mathbf{88.2}^{\star\star}{}_{\pm 0.2}$ indicates that a model has a mean test score (e.g. F_1 or accuracy) of 88.2, with a standard deviation of 0.2, and is stochastically dominant over the baseline.

[3] https://tfhub.dev/google/bert_cased_L-12_H-768_A-12/1

System	CoNLL-2005 WSJ	CoNLL-2005 Brown	CoNLL-2012 Test
Peters et al. (2018)	-	-	84.6
He et al. (2018)	87.4	80.4	85.5
Ouchi et al. (2018)	87.6	78.7	86.2
Li et al. (2019)	87.7	80.5	86.0
Shi and Lin (2019)	88.1	80.9	86.2
Our Baseline	$87.5_{\pm 0.2}$	$81.2_{\pm 0.4}$	$86.2_{\pm 0.1}$

Table 2: Comparison of baseline SRL system on CoNLL-2005 and CoNLL-2012 against models applying pre-trained encoders of comparable size (F_1).

Datasets We use English PropBank datasets from CoNLL-2005 (Carreras and Màrquez, 2005) and the CoNLL-2012 split (Pradhan et al., 2013) for OntoNotes (Hovy et al., 2006) in order to situate our baseline mode among recent work in PB SRL. We compare against models of similar size (120M parameters) with pre-identified predicates.

The SemLink corpus (Palmer, 2009) is currently the only dataset that contains explicit VerbNet thematic role annotations with VN sense annotations. SemLink contains mappings between VN, PB and FrameNet, with annotations performed over a subset of the CoNLL-2005 PB WSJ annotations and Brown corpus out-of-domain test set (Carreras and Màrquez, 2005). Using SemLink thus allows us to evaluate performance for both PB and VN roles on the same source text. Following Zapirain et al. (2008), we restrict evaluation to propositions with PB core arguments fully mapped to VN thematic roles. This accounts for 56% of the original corpus. We include PB modifier roles in addition to VN thematic roles.

Baseline Comparisons Our baseline SRL model achieves comparable performance to Shi and Lin (2019) on both CoNLL-2012 and CoNLL-2005 and thus has performance similar to state-of-the-art models of the same size.

To compare our VerbNet classification models against prior work, we train and evaluate a publicly available state-of-the-art VN classification system directly on the SemLink corpus. We use Clear-

WSD[4], which is a sense disambiguation library tailored for verb sense disambiguation based on linear models over features constructed from an ensemble of word representations applied over syntactic relations (Palmer et al., 2017).

VerbNet Models The results of our experiments are shown in Table 3. First, we find that incorporating gold VerbNet classes (GOLD CLASS) significantly improves VerbNet SRL, providing a 15% relative error reduction on out-of-domain data (80.1 to 83.0), and 6% reduction on in-domain data (87.4 to 88.2). In PB SRL, gold classes are also beneficial, but to a lesser degree. ALL CLASSES and PREDICTED CLASS models improve both in-domain and out-of-domain VN SRL.

Predicting both VN classes and semantic roles from a single encoder reduces the total computational resources required to make predictions from separate models, providing a practical benefit. Additionally, we are interested in determining whether our multi-task models lead to improvements in generalization. Our multi-task model SRL + VSD, which does not condition thematic role prediction on predicted senses, does not have a significant effect on VN SRL performance. However, we do find that conditioning SRL on VN class predictions in a multi-task model (SRL | VSD) leads to a significant improvement in performance on the out-of-domain Brown test set for VN SRL. No significant change is observed on the in-domain WSJ test set, or when the model is applied to PB SRL.

We also evaluate the impact of multi-task learning on predicate disambiguation (VN classification). First, we find that even our baseline model is competitive with the highly-specialized approach for verb sense disambiguation provided in ClearWSD (Table 4). Comparing our joint VN SRL models with a single task baseline for VN classification, we observe a significant improvement on WSJ test data when incorporating multi-task supervision from SRL. This approach is related to earlier use of SRL features for verb sense disambiguation reported in Dang and Palmer (2005), and the positive result is consistent with their findings.

7 Analysis

Monosemous vs. Polysemous Predicates To understand the impact of VerbNet class features, we break down our evaluation by polysemous and monosemous verbs in Table 5. First, we observe that incorporating VN classes improves F_1 scores for monosemous verbs in both models. This is expected, as monosemous verbs are typically lower frequency, with low-frequency and OOV verbs benefiting the most from parameter sharing with other verbs belonging to the same VN classes. We also observe a significant improvement on polysemous verbs in the WSJ (in-domain) test set when including VN features. However, polysemous verbs in the Brown (out-of-domain) test set only benefit from using explicitly predicted classes, but not when using all valid classes for each predicate.

Why does ALL CLASSES improve performance on out-of-domain data for monosemous verbs, but not polysemous verbs? Intuitively, the per-verb distributions of VerbNet classes may change considerably between two domains. Using a correctly-predicted class may help mitigate errors on verbs for which one class was dominant during training, but a different class or set of classes are observed during testing in the new domain. This benefit would not be observed with ALL CLASSES as for a given verb, the same classes used as model inputs during training would be used as inputs on out-of-domain data. However, VN classes receive fewer updates during training when using only predicted classes. Thus, verbs appearing in classes that never or rarely appeared during training will not benefit from PREDICTED CLASS features. ALL CLASSES may mitigate this issue, since even if a specific class does not appear in the training data, it still can receive updates from examples of polysemous member verbs that belong to other classes (and improved performance over PREDICTED CLASS on monosemous verbs on the out-of-domain Brown test set supports this). As future work, a promising direction may therefore be to combine PREDICTED CLASS and ALL CLASSES features.

Out-of-Vocabulary Predicates How well do models incorporating VerbNet features generalize on out-of-vocabulary and rare predicates? We split an evaluation on the WSJ development set into 5 bins by training set predicate frequency (shown in Figure 1). Comparing development F_1 scores for ALL CLASSES and PREDICTED CLASS models against our baseline model, we note that VN classes improve SRL performance most for predicates appearing 0-50 times in the training data, which account for 24.4% of instances in the development set.

[4] https://github.com/clearwsd/clearwsd

System	PropBank WSJ	PropBank Brown	VerbNet WSJ	VerbNet Brown
Zapirain et al. (2008)	$78.9_{\pm 0.9}$	–	$77.0_{\pm 0.9}$	$62.9_{\pm 1.0}$
Baseline	$88.5_{\pm 0.1}$	$82.4_{\pm 0.5}$	$87.4_{\pm 0.2}$	$80.1_{\pm 0.4}$
SRL + VSD	$88.2_{\pm 0.2}$	$82.8^{\star}_{\pm 0.6}$	$87.3_{\pm 0.1}$	$80.0_{\pm 0.7}$
SRL | VSD	$88.3_{\pm 0.2}$	$82.2_{\pm 0.4}$	$87.4_{\pm 0.2}$	$80.6^{\star\star}_{\pm 0.4}$
PREDICTED CLASS	$88.3_{\pm 0.1}$	$81.2_{\pm 0.6}$	$87.6^{\star\star}_{\pm 0.1}$	$80.9^{\star\star}_{\pm 0.6}$
ALL CLASSES	$\mathbf{88.6}^{\star}_{\pm 0.3}$	$82.3_{\pm 0.5}$	$\mathbf{87.6}^{\star\star}_{\pm 0.2}$	$\mathbf{81.1}^{\star\star}_{\pm 0.6}$
GOLD CLASS	$\mathbf{88.7}^{\star\star}_{\pm 0.0}$	$\mathbf{82.8}^{\star}_{\pm 0.2}$	$\mathbf{88.2}^{\star\star}_{\pm 0.2}$	$\mathbf{83.0}^{\star\star}_{\pm 0.9}$

Table 3: F_1 scores of models incorporating different predicate representations and sense distinctions on VerbNet and PropBank SRL on SemLink. SRL + VSD and SRL | VSD are multitask models for SRL and VerbNet classification, with the latter using predicted classes as features for SRL. ALL CLASSES, PREDICTED CLASS, and GOLD CLASS are SRL models using VerbNet class features (the list of all VerbNet classes the predicate belongs to, predicted VerbNet classes, and gold VerbNet classes respectively).

System	WSJ	Brown
ClearWSD	$97.0_{\pm 0}$	$89.3_{\pm 0}$
Baseline	$97.3_{\pm 0.1}$	$90.7_{\pm 0}$
SRL + VSD	$\mathbf{97.7}^{\star\star}_{\pm 0.1}$	$91.3_{\pm 0.4}$
SRL | VSD	$97.6^{\star\star}_{\pm 0.1}$	$91.3_{\pm 0}$

Table 4: VerbNet classification (sense disambiguation) accuracy on SemLink.

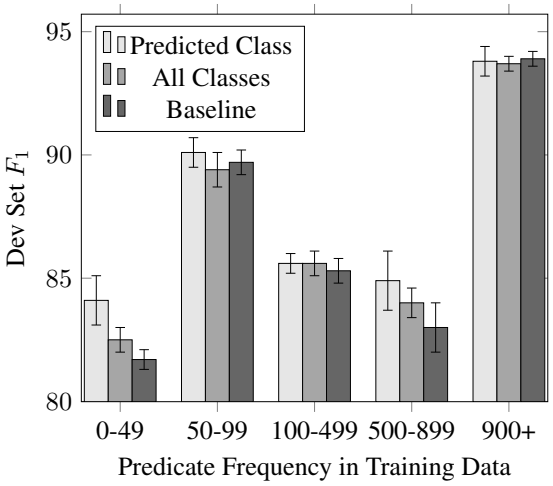


Figure 1: Evaluation by training set predicate frequency on the SemLink development data comparing the impact of VerbNet features.

Focusing on low-frequency predicates, we further divide our evaluation of predicates occurring fewer than 50 times in the training data into 6 bins, one of which is reserved for OOV predicates (Figure 2). From this analysis, we find that VN classes are most impactful on predicates appearing fewer than 10 times in the training data, with a large improvement over the baseline on OOV predicates when applying predicted classes.

8 Conclusions and Future Work

We investigate VerbNet classes as an effective level of abstraction for predicates when performing semantic role labeling. We find that incorporating features based on gold VerbNet classes improves both VerbNet and PropBank SRL, but when predicted classes are used, this effect is only observed for VerbNet. An improvement is also observed without explicit prediction of classes by including a list of all VerbNet classes the target predicate belongs to as features. Breaking down our evaluation into polysemous and monosemous predicates, we find that predicted classes help more on out-of-domain polysemous predicates, while using all valid VerbNet classes helps more on out-of-domain low-frequency predicates. In multi-task learning experiments motivated by the interdependence of VN classification and SRL, we find that joint training improves both tasks when conditioning role labeling on predicted predicates, facilitating VN semantic parsing. In future work, we will investigate alternative approaches incorporating the structure of VerbNet into the parsing of VerbNet semantic representations. Finally, we hope to expand our evaluations to larger, more diverse datasets to further investigate domain transfer.

System	Polysemous WSJ	Brown	Monosemous WSJ	Brown
Baseline	(+0.0) $88.2_{\pm 0.3}$	(+0.0) $81.8_{\pm 0.8}$	(+0.0) $85.9_{\pm 0.3}$	(+0.0) $77.7_{\pm 1.3}$
ALL CLASSES	(+0.4) $\mathbf{88.6}^{\star\star}{}_{\pm 0.2}$	(−0.2) $81.6_{\pm 0.8}$	(+0.2) $86.1^{\star}{}_{\pm 0.4}$	(+2.6) $\mathbf{80.3}^{\star\star}{}_{\pm 0.8}$
PREDICTED CLASS	(+0.3) $88.5^{\star\star}{}_{\pm 0.2}$	(+0.5) $\mathbf{82.3}^{\star}{}_{\pm 0.8}$	(+0.2) $\mathbf{86.1}^{\star\star}{}_{\pm 0.2}$	(+0.9) $78.6^{\star}{}_{\pm 1.3}$

Table 5: Evaluation of contribution of VerbNet features on polysemous vs. monosemous predicates for VerbNet SRL averaged over all models. Average change over the baseline performance is given in parentheses.

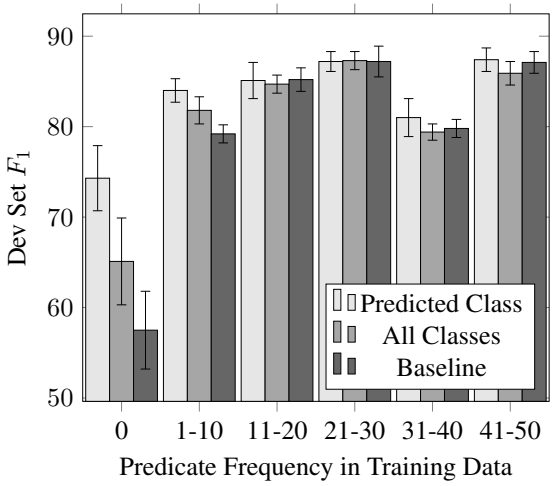


Figure 2: Evaluation by training set predicate frequency similar to Figure 1, but focused on low-frequency predicates. Most improvements are for predicates appearing fewer than 10 times in the training data.

Acknowledgments

We gratefully acknowledge the support of C3 (Cognitively Coherent Human-Computer Communication, subcontracts from UIUC and SIFT), DARPA AIDA Award FA8750-18-2-0016 (RAMFIS), and DTRA HDTRA1-16-1-0002/Project 1553695 (eTASC - Empirical Evidence for a Theoretical Approach to Semantic Components). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any government agency. This work was partially supported by research credits from Google Cloud. Finally, we thank the anonymous IWCS reviewers for their insightful comments and suggestions.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI*, pages 265–283.

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.

Marzieh Bazrafshan and Daniel Gildea. 2013. Semantic roles for string to tree machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 419–423, Sofia, Bulgaria. Association for Computational Linguistics.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar. Association for Computational Linguistics.

Claire Bonial, Olga Babko-Malaya, Jinho D Choi, Jena Hwang, and Martha Palmer. 2010. PropBank Annotation Guidelines. Technical report, Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder.

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. VerbNet representations: Subevent semantics for transfer verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163, Florence, Italy. Association for Computational Linguistics.

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)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Rich Caruana. 1998. Multitask learning. In *Learning to Learn*, pages 95–133. Springer.

Yun-nung Chen, William Yang Wang, and Alexander I Rudnicky. 2013. Unsupervised Induction and Filling of Semantic Slots for Spoken Dialogue Systems Using Frame-Semantic Parsing. *ASRU*, pages 120–125.

Peter Clark, Bhavana Dalvi, and Niket Tandon. 2018. What happened? leveraging verbnet to predict the effects of actions in procedural text. *arXiv:1804.05435*.

Daniel Dahlmeier, Hwee Tou Ng, and Tanja Schultz. 2009. Joint learning of preposition senses and semantic roles of prepositional phrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 450–458, Singapore. Association for Computational Linguistics.

Hoa Trang Dang and Martha Palmer. 2005. The role of semantic roles in disambiguating verb senses. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 42–49, Ann Arbor, Michigan. Association for Computational Linguistics.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, pages 4171–4186.

Kaustubh Dhole and Christopher D. Manning. 2020. Syn-QG: Syntactic and shallow semantic rules for question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 752–765, Online. Association for Computational Linguistics.

David Dowty. 1991. Thematic Proto-Roles and Argument Selection. *Language*, 67(3):547–619.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970, Lisbon, Portugal. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

James Gung. 2013. Using Relations for Identification and Normalization of Disorders: Team CLEAR in the ShARe/CLEF 2013 eHealth Evaluation Lab. In *CLEF (Working Notes)*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933, Copenhagen, Denmark. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Daisuke Kawahara and Martha Palmer. 2014. Single classifier approach for verb sense disambiguation based on generalized features. In *Proceedings of the Ninth International Conference on Language*

Resources and Evaluation (LREC'14), pages 4210–4213, Reykjavik, Iceland. European Language Resources Association (ELRA).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*.

Ilia Kuznetsov and Iryna Gurevych. 2020. A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182, Online. Association for Computational Linguistics.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dependency or Span, End-to-End Uniform Semantic Role Labeling. *AAAI*, 33:6730–6737.

Ding Liu and Daniel Gildea. 2010. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 716–724, Beijing, China. Coling 2010 Organizing Committee.

Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2017. GUIR at SemEval-2017 task 12: A framework for cross-domain clinical temporal information extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1024–1029, Vancouver, Canada. Association for Computational Linguistics.

Paola Merlo and Lonneke Van Der Plas. 2009. Abstraction and generalisation in semantic role labels: PropBank, VerbNet or both? In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 288–296, Suntec, Singapore. Association for Computational Linguistics.

Paloma Moreda and Manuel Palomar. 2006. The Role of Verb Sense Disambiguation in Semantic Role Labeling. *Lecture Notes in Computer Science Advances in Natural Language Processing*, pages 684–695.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Martha Palmer. 2009. SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon Conference*, pages 9–15. 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.

Martha Palmer, James Gung, Claire Bonial, Jinho Choi, Orin Hargraves, Derek Palmer, and Kevin Stowe. 2017. The Pitfalls of Shortcuts: Tales from the Word Sense Tagging Trenches. *Springer series Text, Speech and Language Technology*, Essays in Lexical Semantics and Computational Lexicography - In Honor of Adam Kilgarriff.

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.

Matthew 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.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. *Dissertation Abstracts International, B: Sciences and Engineering*, 66(6).

Peng Shi and Jimmy Lin. 2019. Simple BERT Models for Relation Extraction and Semantic Role Labeling. *arXiv:1904.05255*.

Kevin Stowe, Sarah Moeller, Laura Michaelis, and Martha Palmer. 2019. Linguistic analysis improves neural metaphor detection. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 362–371, Hong Kong, China. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3772–3782, Brussels, Belgium. Association for Computational Linguistics.

Gokhan Tur and Dilek Hakkani-Tür. 2005. Semi-Supervised Learning for Spoken Language Understanding Using Semantic Role Labeling. *Language*, pages 232–237.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. 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 700–706, Beijing, China. Association for Computational Linguistics.

Ronald J. Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*.

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:1609.08144*.

Qingrong Xia, Zhenghua Li, and Min Zhang. 2019. A syntax-aware multi-task learning framework for Chinese 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 5382–5392, Hong Kong, China. Association for Computational Linguistics.

Szu-ting Yi. 2007. *Robust Semantic Role Labeling Using Parsing Variations and Semantic Classes*. Ph.D. thesis, University of Pennsylvania.

Szu-ting Yi, Edward Loper, and Martha Palmer. 2007. Can semantic roles generalize across genres? In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 548–555, Rochester, New York. Association for Computational Linguistics.

Beñat Zapirain, Eneko Agirre, and Lluís Màrquez. 2008. Robustness and generalization of role sets: PropBank vs. VerbNet. In *Proceedings of ACL-08: HLT*, pages 550–558, Columbus, Ohio. Association for Computational Linguistics.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137, Beijing, China. Association for Computational Linguistics.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2020. LIMIT-BERT : Linguistics informed multi-task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online. Association for Computational Linguistics.

Critical Thinking for Language Models

Gregor Betz
KIT
Karlsruhe, Germany
gregor.betz@kit.edu

Christian Voigt
KIT
Karlsruhe, Germany
christian.voigt@kit.edu

Kyle Richardson
Allen Institute for AI
Seattle, WA, USA
kyler@allenai.org

Abstract

This paper takes a first step towards a critical thinking curriculum for neural auto-regressive language models. We introduce a synthetic corpus of deductively valid arguments, and generate artificial argumentative texts to train CRiPT: a critical thinking intermediarily pre-trained transformer based on GPT-2. Significant transfer learning effects can be observed: Trained on three simple core schemes, CRiPT accurately completes conclusions of different, and more complex types of arguments, too. CRiPT generalizes the core argument schemes in a correct way. Moreover, we obtain consistent and promising results for NLU benchmarks. In particular, CRiPT's zero-shot accuracy on the GLUE diagnostics exceeds GPT-2's performance by 15 percentage points. The findings suggest that intermediary pre-training on texts that exemplify basic reasoning abilities (such as typically covered in critical thinking textbooks) might help language models to acquire a broad range of reasoning skills. The synthetic argumentative texts presented in this paper are a promising starting point for building such a "critical thinking curriculum for language models."

1 Introduction

Pre-trained autoregressive language models (LM) such as GPT-2 and GPT-3 achieve, remarkably, competitive results in a variety of language modeling benchmarks without task-specific fine-tuning (Radford et al., 2019; Brown et al., 2020). Yet, it is also widely acknowledged that these models struggle with reasoning tasks, such as natural language inference (NLI) or textual entailment (Askell, 2020). Actually, that doesn't come as a surprise, given the tendency of humans to commit errors in reasoning (Kahneman, 2011; Sunstein and Hastie, 2015), their limited critical thinking skills (Paglieri, 2017), and the resulting omnipresence of fallacies and biases in texts and the frequently low argumentative quality of online debates (Hansson, 2004; Guiaşu and Tindale, 2018; Cheng et al., 2017): Neural language models are known to pick up and reproduce *normative* biases (e.g., regarding gender or race) present in the dataset they are trained on (Gilburt and Claydon, 2019; Blodgett et al., 2020; Nadeem et al., 2020), as well as other *annotation artifacts* (Gururangan et al., 2018); no wonder this happens with *argumentative* biases and reasoning flaws, too (Kassner and Schütze, 2020; Talmor et al., 2020). This diagnosis suggests that there is an obvious remedy for LMs' poor reasoning capability: make sure that the training corpus contains a sufficient amount of exemplary episodes of sound reasoning.

In this paper, we take a first step towards the creation of a "critical thinking curriculum" for neural language models. Critical thinking can be loosely defined as "reasonable reflective thinking that is focused on deciding what to believe or do." (Norris and Ennis, 1989) Generally speaking, our study exploits an analogy between teaching critical thinking to students and training language models so as to improve their reasoning skill. More specifically, we build on three key assumptions that are typically made in critical thinking courses and textbooks: First, there exist fundamental reasoning skills that are required for, or highly conducive to, a large variety of more specific and advanced critical thinking skills (e.g., Fisher, 2001, p. 7). Second, drawing deductive inferences is one such basic ability (e.g., Fisher, 2001, pp. 7–8). Third, reasoning skills are not (just) acquired by learning a theory of correct reasoning, but by studying lots of examples and doing "lots of good-quality exercises" (Lau and Chan, 2020), typically moving from simple to more difficult problems (e.g., Bowell and Kemp, 2014).

These insights from teaching critical thinking translate, with respect to our study, as follows (see Fig. 1). First of all, we design and build 'lots of good-quality exercises': a synthetic corpus of de-

Proceedings of the 14th International Conference on Computational Semantics, pages 63–75
June 17–18, 2021.

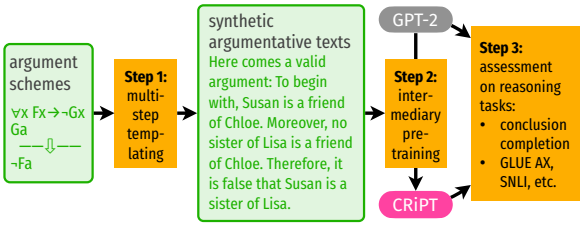


Figure 1: Training and testing of CRiPT language models (critical thinking intermediarily pre-trained transformer) with synthetic argumentative texts.

ductively valid arguments which instantiate a variety of (syllogistic) argument schemes, and which are rendered as text paragraphs (Section 3). Next, we use our synthetic argument text corpus to train and to evaluate GPT-2 (Section 4). The training, which maximizes a causal language modeling objective, can be conceived of as a generic, intermediary pre-training in the spirit of STILTS (Phang et al., 2018) and yields models we term CRiPT (critical thinking intermediarily pre-trained transformer).

Evaluating CRiPT's ability to correctly complete conclusions of arguments, we observe strong transfer learning effects/generalization (Section 5): Just training CRiPT on a few central core schemes (generalized modus ponens, contraposition and chain rule) allows it to accurately complete conclusions of different types of arguments, too (e.g., complex argumentative forms that involve dilemma and de Morgan). The language models appear to connect and generalize the core argument schemes in a correct way. In addition, CRiPT is equally able to apply learned argument patterns beyond the training corpus' domain.

Moreover, we test CRiPT on different reasoning benchmarks. Because we are particularly interested in transfer learning effects, we do so in a zero-shot set-up (i.e., evaluating our argumentation models on entirely unrelated NLU tasks, which follows recent work by Mitra et al. (2019); Shwartz et al. (2020); Ma et al. (2020)). We obtain consistent and promising results for the GLUE diagnostics (Wang et al., 2018) and SNLI (Bowman et al., 2015) benchmarks (Section 5), finding that training on core schemes clearly improves the NLU skills of pre-trained models.

All these transfer learning effects observed strengthen the analogy between teaching critical thinking and training language models: A variety of reasoning skills are improved by generic, intermediary pre-training on high-quality texts that exemplify a basic reasoning skill, namely simple deductive argumentation. Obviously, drawing correct inferences is just one of the elementary skills typically covered in critical thinking courses (Fisher, 2001). Critical thinking involves more than deduction. And it would hence, by analogy, be unreasonable to expect that intermediary pre-training on the synthetic argument corpus suffices to turn language models into accomplished reasoners. However, we have shown that argumentative texts (with valid syllogistic arguments) are certainly a good starting point when building a more comprehensive dataset for initial or intermediary pre-training that might help language models to acquire a broad range of reasoning skills. Or, to put it differently, the synthetic argumentative texts might belong to the core of a "critical thinking curriculum for language models." In the final section, we advance some ideas for complementing the artificial argument corpus so as to further improve the performance of LMs with regard to different reasoning benchmarks.

2 Related Work

To our knowledge, this paper is, together with Gontier et al. (2020), among the first to show that autoregressive language models like GPT-2 can learn to reason by training on a *text corpus* of correct natural language arguments. By contrast, previous work in this field, described below, has typically modeled natural language reasoning problems as classification tasks and trained neural systems to accomplish them. For example, Schick and Schütze (2021); Schick and Schütze (2020) find that a *masked* language model with classification head achieves remarkable NLU performance by pre-structuring the training data. This paper explores the opposite route: We start with highly structured (synthetic) data, render it as unstructured, plain text and train a *uni-directional* language model on the synthetic text corpus.

Over and above the methodological novelty of our approach, we discuss, in the following, related reasoning benchmarks and explain what sets our synthetic argument corpus apart from this work.

Rule reasoning in natural language Various datasets have been developed for (deductive) rule reasoning in natural language. One-step rule application (cf. Weston et al., 2016; Richardson et al., 2020; Tafjord et al., 2019; Lin et al., 2019) closely resembles the conclusion completion task for *gen-*

eralized modus ponens and *generalized modus tollens* schemes described below. However, we go beyond previous work in investigating the ability of LMs to infer conclusions that have a more complex logico-semantic structure (e.g., existential or universal statements). RuleTaker, arguably the most general system for rule reasoning in natural language so far, is a transformer model for multi-hop inference (Clark et al., 2020). PRover (Saha et al., 2020) extends RuleTaker by a component for proof generation and is able to construct valid proofs and outperforms RuleTaker in terms answer accuracy in a zero-shot setting.

Benchmarks for enthymematic reasoning An 'enthymeme' is an argument whose premises are not explicitly stated, e.g.: "Jerry is a mouse. Therefore, Jerry is afraid of cats." The following studies involve such reasoning with implicit assumptions, whereas our synthetic argument corpus doesn't: all premises are transparent and explicitly given. COMET generates and extends commonsense knowledge graphs (Bosselut et al., 2019). Trained on seed data, the model is able to meaningfully relate subject phrases to object phrases (by doing the type of completion tasks we introduce in Section 4). The Argument Reasoning Comprehension (ARC) dataset (Habernal et al., 2018) comprises simple informal arguments. The task consists in identifying which of two alternative statements is the missing premise in the argument (see also Niven and Kao, 2019). CLUTRR is a task generator for relational reasoning on kinship graphs (Sinha et al., 2019). CLUTTR takes a set of (conceptual) rules about family relations as given and constructs set-theoretic possible worlds (represented as graphs) which instantiate these rules. The task consists in inferring the target fact from the base facts alone – the conceptual rules remain implicit. Gontier et al. (2020) show that Transformers do not only learn to draw the correct conclusion (given a CLUTTR task), but also seems to acquire the ability to generate valid proof chains. Finally, training on synthetic knowledge-graph data *from scratch*, Kassner et al. (2020) find that BERT (Devlin et al., 2019) is able to correctly infer novel facts implicit in the training data.

Critical thinking tasks LogiQA (Liu et al., 2020) is a collection of publicly available critical thinking questions, used by the National Civil Servants Examination of China to assess candidates' critical thinking and problem solving skills. Its scope is much broader than our highly specific and carefully designed argument corpus.

3 An Artificial Argument Corpus

This section describes the construction of a synthetic corpus of natural language arguments used for training and evaluating CRiPT.[1]

The corpus is built around eight simple, deductively valid syllogistic argument schemes (top row in Fig. 2). These eight *base schemes* have been chosen because of their logical simplicity as well as their relevance in critical thinking and argument analysis (Feldman, 2014; Bowell and Kemp, 2014; Brun and Betz, 2016). Each of these eight base schemes is manually varied in specific ways to create further deductively correct variants, which are verified for correctness using an off-the-shelf theorem prover.

Negation variants of base schemes are created by substituting a sub-formula with its negation (e.g., $Fx \rightsquigarrow \neg F_1x$) and/or by applying *duplex negatio affirmat*. *Complex predicates* variants build on base schemes or their respective negation variants and are obtained by substituting atomic predicates with compound disjunctive or conjunctive ones (e.g., $Fx \rightsquigarrow F_1x \vee F_2x$). *De Morgan* variants of base schemes are finally derived by applying de Morgan's law to the respective variants created before (a de Morgan variant of modus ponens is, for instance: $\forall x : \neg(Fx \vee Gx) \rightarrow Hx; \neg Fa; \neg Ga \Rightarrow Ha$).

With 2-3 different versions for each of these variations of a base scheme (parameter *n* in Fig. 2), we obtain, in total, 71 distinct handcrafted argument schemes. In view of their simplicity and prominence in natural language argumentation, three of the eight *base schemes* are marked as *core schemes*: generalized modus ponens, generalized contraposition, hypothetical syllogism 1.

Natural language instances of the argument schemes can be created by means of a first-order-logic domain (with names and predicates) and natural language templates for the formal schemes. In order to obtain a large variety of realistic natural language arguments, we have devised (i) a

[1]The corpus as well as the source code used to generate it are available at https://github.com/debatelab/aacorpus. Selected example texts which illustrate, in particular, the multiple domains covered by the corpus are presented in Appendix A.

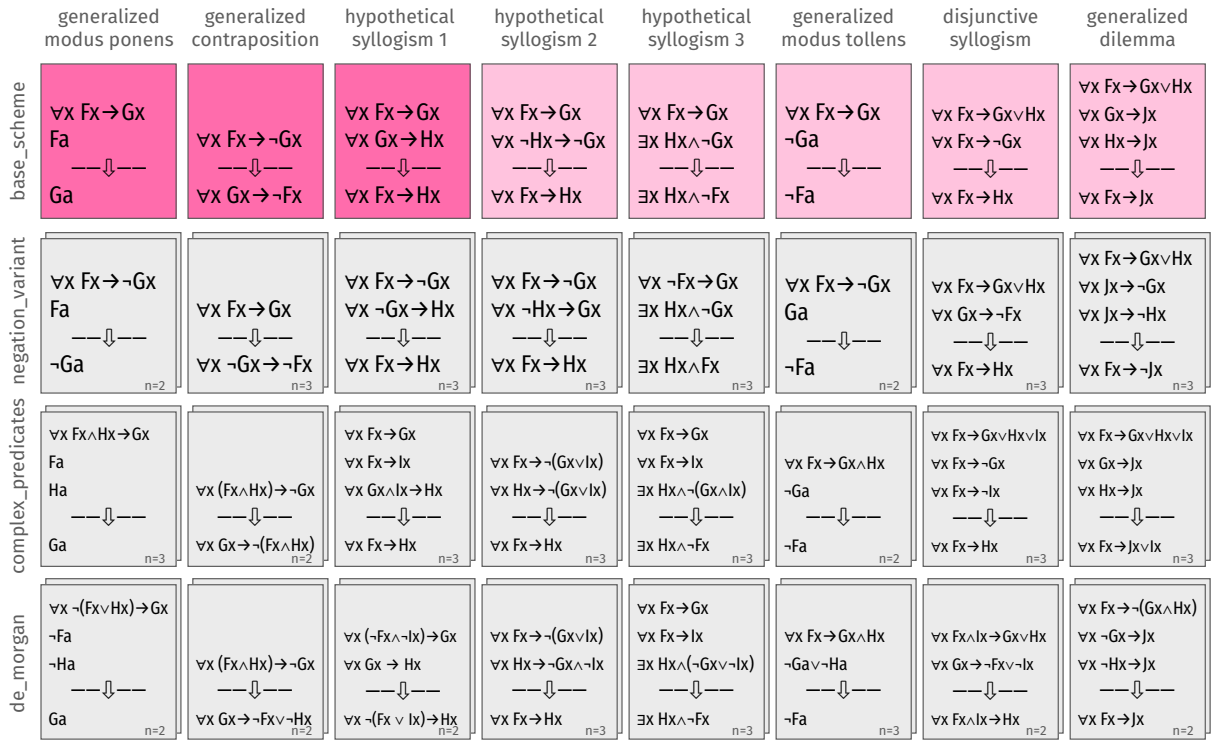


Figure 2: Syllogistic argument schemes used to create an artificial argument corpus with eight base schemes (upper row), three of which are core schemes (left). Parameter n indicates the number of different schemes belonging to one and the same base scheme group (column) and variant (row).

multi-stage templating process with (ii) alternative templates at each stage and (iii) multiple domains.

This process can be split into five consecutive steps.

In *step 1*, the argument scheme, which serves as formal template for the natural language argument, is chosen at random.

In *step 2*, each sentence in the formal scheme (premises and conclusion) is individually replaced by a natural language pattern in accordance with a randomly chosen template. For example, the formula "$\forall x Fx \rightarrow Gx$" might be replaced by any of the following natural language sentence schemes: "Every F is a G", "Whoever is a F is also a G", "Being a G is necessary for being a F", "If someone is a F, then they are a G". Some of these patterns (e.g., the fourth one in the above list) are reserved for generating an out-of-domain test dataset, and are not used for training.

In *step 3*, the entity- and property-placeholders in the resulting argument scheme are replaced argument-wise with names and predicates from a domain. We hence obtain an instance of the formal argument scheme as premise-conclusion list. Each domain provides hundreds of entity-names, which can be paired with different binary predicates to create thousands of different unary predicates. For example, the text in Fig. 1 is obtained by substituting predicates from the domain *female relatives*, which includes predicates like being a "sister of Anna", "granddaughter of Elsa", "cousin of Sarah", ... Once more, some domains are used for testing only, and not for training (see below and Section 4.2).

In *step 4*, the premises of the natural language argument are randomly re-ordered.

In *step 5*, the premise-conclusion list is packed into a text paragraph by adding an argument intro, framing the premises, and adding an inference indicator. Again, multiple templates are available for doing so, which yields a large variety of textual renderings of an argument.

Following this pipeline, we generate natural language instances of each formal argument scheme, thus creating:

1. a training set of argumentative texts, based on the default domains and templates (TRAIN);
2. an evaluation set of argumentative texts, based on the default domains and templates, which are used for development (DEV);
3. a test set of argumentative texts, based on the default domains and templates and used for

final tests (TEST_OUT-OF-SAMPLE);

4. a test set of argumentative texts, based on the domains and templates reserved for testing (TEST_OUT-OF-DOMAIN).

This represents the artificial argument text corpus we use to train and evaluate CRiPT.

4 Experiments with CRiPT

Our basis for training and evaluating CRiPT are three compact versions of GPT-2 with 117M, 345M and 762M parameters, as implemented by Wolf et al. (2019). We note that all of these models fall short of the full-scale model with 1542M parameters.[2]

4.1 Training

From the training items in the Artificial Argument Corpus (TRAIN) we sample three types of differently-sized training sets TRAIN01 $\subset$ TRAIN02 $\subset$ TRAIN03 as follows (see also the color pattern in Fig. 2):

- TRAIN01: all training items which are instances of a *core scheme*, i.e. generalized modus ponens, generalized contraposition, hypothetical syllogism 1 (N=4.5K, 9K, 18K, 36K)
- TRAIN02: all training items which are instances of a *base scheme* (N=4.5K, 9K, 18K, 36K)
- TRAIN03: all training items in the corpus (N=4.5K, 9K, 18K, 36K)

In an attempt to avoid over-fitting, we blend the training arguments with snippets from Reuters news stories (Lewis et al., 2004) and the standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2018), trying a mixing ratio of 1:1 and thus doubling training size to N=9K, 18K, 36K, 72K.[3] Training the BASE model (pre-trained GPT-2) on TRAIN01–TRAIN03 yields three corresponding CRiPT models (see Appendix B). For purpose of comparison, we have similarly trained three randomly initialized Transformer models (structurally identical with GPT-2) – none of these random models gains any performance through training on our critical thinking corpus.

[2]The fine-tuned models are released through https://huggingface.co/debatelab.

[3]We find that fine-tuning on the accordingly enhanced argument corpus still increases the model's perplexity on the Wiki103 dataset by a factor of 1.5 (see Appendix D), which suggests to mix a higher proportion of common texts into the training data in future work.

4.2 Testing

Conclusion Completion on Artificial Argument Corpus To test whether language models can reason correctly, we assess their ability to accurately complete conclusions of arguments in the artificial argument corpus. Here, we make use of the fact that, by construction, the conclusion of every argument in the corpus ends with a predicate (a property-term such as "sister of Chloe" or "supporter of Tottenham Hotspurs"), which is potentially preceded by a negator. First of all, as shown in Table 1, we test whether the model is able to correctly fill in the final predicate (task *split*). The second, more difficult task consists in completing the final predicate plus, if present, the preceding negator (task *extended*). With a third, adverserial task we check how frequently the model wrongly adjoins the complement of the correct completion of the *extended* task (task *inverted*).

Task	Conclusion with cloze-style prompt	Completion
split	Every *F* is a | *G*	*G*
	Some *F* is not a | *G*	*G*
	a is a *F* or not a | *G*	*G*
extended	Every *F* is | a *G*	a *G*
	Some *F* is | not a *G*	not a *G*
	a is a *F* or | not a *G*	not a *G*
inverted	Every *F* is | a *G*	not a *G*
	Some *F* is | not a *G*	~~not~~ a *G*
	a is a *F* or | not a *G*	~~not~~ a *G*

Table 1: Three conclusion completion tasks

Clearly, the higher the accuracy in the *split* and *extended* tasks, and the lower the accuracy in the *inverted* task, the stronger the model's reasoning performance.

Based on the artificial argument corpus (see Section 3), we generate and distinguish three different test datasets, each of which comprises the three tasks described above, as follows:

- *out of sample (oos)*: contains items from TEST_OUT-OF-SAMPLE, which share domain and natural language templates with the training data;
- *paraphrased (para)*: a sample of 100 items, randomly drawn from TEST_OUT-OF-SAMPLE, which have been manually reformulated so as to alter the premises' grammatical structure

imposed by the natural language templates;

- *out of domain (ood)*: contains items from TEST_OUT-OF-DOMAIN, which belong to different domains and instantiate grammatical patterns other than the training data.

Technically, conclusion completions, in all tasks and tests, are generated by the language model with nucleus sampling and top-p $= 0.9$ (Holtzman et al., 2019).

Classification for NLU Benchmarks To investigate transfer learning effects, we evaluate the trained models on standard NLU benchmarks, such as GLUE AX and SNLI. These benchmark tasks are classification problems. In the following, we describe how we use the generative language models to perform such classification.

Using simple templates, we translate each benchmark entry into alternative prompts (e.g., context and question) and/or alternative completions (e.g., answers). Consider for example a GLUE-style problem given by two sentences "The girl is eating a pizza." and "The girl is eating food" and the question whether one entails, contradicts, or is independent of the other. We can construct three prompts, corresponding to the three possible answers (entail / contradict / independent):

> *Prompt1*: The girl is eating a pizza. Therefore,
> *Prompt2*: The girl is eating a pizza. This rules out that
> *Prompt3*: The girl is eating a pizza. This neither entails nor rules out that
> *Completion*: the girl is eating food.

In this case, the correct match is obviously *Prompt1–Completion*. The ability of a language model to discern that "The girl is eating pizza" entails (and does not contradict) "The girl is eating food" will be reflected in a comparatively low conditional perplexity of *Completion* given *Prompt1* and a correspondingly high conditional perplexity of *Completion* given *Prompt2* or *Prompt3*.

Generally put, we classify a given input X by constructing N alternative prompts $p_1, \ldots p_N$ and a completion c, such that each pair (p_i, c) corresponds to a class $i \in \{1 \ldots N\}$ of the classification problem. The conditional perplexity of the completion c given prompt p_i according to the language model serves as prediction score for our classifier (as for instance in Shwartz et al., 2020).

5 Results

Conclusion Completion on Artificial Argument Corpus Does CRiPT correctly complete conclusions of natural language arguments? Fig. 3 displays the evaluation results in an aggregated way. Each subplot visualizes the accuracy of the models in the three completion tasks for a different test dataset (see Section 4.2).

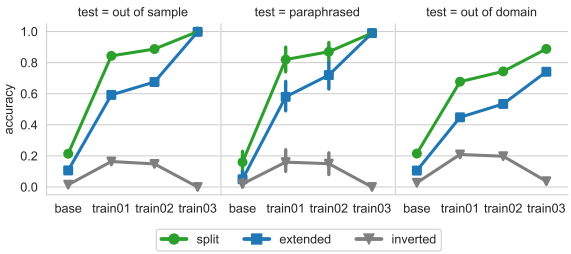


Figure 3: Accuracy of CRiPT in three conclusion completion tasks and on different test datasets (out of sample, paraphrased, out of domain).

We may observe, first of all, that pre-training on the argument corpus effectively improves conclusion-completion-skill. In all three test datasets, the accuracy in the *split* and *extended* tasks increases as models are trained on more and more argument schemes, far exceeding the base model's performance. Once CRiPT has seen all schemes (TRAIN03), accuracy levels reach 100% for in-domain and 70%-90% for out-of-domain tests. However, the TRAIN01 and TRAIN02 models do also generate more incorrect completions than the BASE model (*inverted* task). But the frequency of such incorrect completions increases much less than the frequency of correct ones (the gap between blue and gray curve widens), and it actually falls back to almost zero with the TRAIN03 model. Out-of-domain performance of CRiPT (right-hand plot) is qualitatively similar and only slightly less strong than in-domain performance (left-hand and middle plot). CRiPT models trained on a given domain are able to effectively exercise the acquired skill in other domains, and have hence gained topic-neutral, universal reasoning ability.

The strong performance of TRAIN01 models (Fig. 3) indicates that training on a few argument schemes positively affects performance on other schemes, too. To further investigate transfer learning, Table 2 contrasts (a) CRiPT's accuracy on schemes it has not been trained on – averaged over TRAIN01 and TRAIN02 models – with (b) its accuracy on schemes present in the respective train-

Task	BASE	(A) UNSEEN SCH.			(B) SEEN SCH.		
		oos	*para*	*ood*	*oos*	*para*	*ood*
split	21.4	85.4	82.0	69.4	99.9	99.2	89.0
ext.	10.7	60.3	59.3	45.8	99.9	99.2	76.2
inv.	1.5	16.9	18.0	22.1	0.0	0.0	3.2

Table 2: Accuracy of CRiPT models in three conclusion completion tasks and on different test datasets (out of sample: oos, paraphrased: para, out of domain: ood). Columns report, separately, the performance (A) on schemes the model has not been trained on (TR01–02), and (B) on schemes that are covered by the model's training data (TR01–03). For comparison, column BASE reports the performance of pre-trained GPT-2, averaged over all schemes.

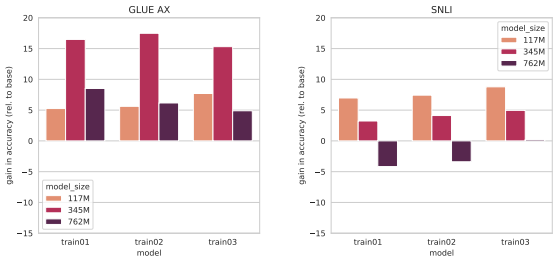


Figure 4: Gains in accuracy due to fine-tuning on the AAC (accuracy TRAIN model – accuracy BASE model) for differently sized models and different NLP benchmark tasks: the GLUE diagnostics data, and the SNLI dataset.

ing corpus – averaged over TRAIN01, TRAIN02, and TRAIN03 models. The upshot is that CRiPT performs much more strongly than the base model not only on argument schemes it has been trained on, but also on those schemes not seen yet. We take this to be a promising result as it strengthens the analogy between teaching critical thinking and training language models: intermediary pre-training on high-quality texts that exemplify a specific, basic reasoning skill – namely, simple deductive argumentation – improves other, more complex reasoning skills.

Moreover, a closer look at the scheme-specific performance suggests important variations in CRiPT's ability to generalize, for it seems to struggle with unseen schemes which involve negations (e.g., CRiPT-TRAIN02 generates more incorrect than correct completions of the *negation_variants* of generalized modus ponens, see Appendix C). This is consistent with the finding that some NLMs seemingly fail to understand simple negation (Kassner and Schütze, 2020; Talmor et al., 2020).

To further understand transfer learning effects, we next examine CRiPT's zero-shot performance in other NLP reasoning tasks (i.e., without task-specific fine-tuning).

GLUE AX The GLUE datasets (Wang et al., 2018) represent standard benchmarks for natural language understanding (NLU). We evaluate our models' NLU skill in terms of accuracy on the curated GLUE diagnostics dataset (Fig. 4).

Training on the artificial argument corpus substantially boosts accuracy on the GLUE diagnostics. Accuracy increases by at least 5 and up to 17 percentage points, depending on model size. Remarkably, training on the core scheme alone suffices to bring about these improvements.

This is a major finding and our clearest evidence so far that critical thinking pre-training involves substantial transfer learning effects.

SNLI Our assessment of CRiPT with respect to SNLI data (Bowman et al., 2015) proceeds in close analogy to the GLUE benchmark. The results (Fig. 4) are consistent with, albeit less definite than our previous findings for the GLUE benchmark: First and foremost, training on all schemes (TRAIN03) improves the performance by up to 8 percentage points. Training on fewer schemes is slightly less effective. However, only small and medium sized CRiPT profit from pre-training on the AAC; while the performance of the 762M model drops. This might be due to a coincidentally strong performance of the corresponding BASE model (see Appendix D), or suggest that large GPT-2 has already learned during general pre-training whatever is of relevance for SNLI in argumentative texts. (Further experiments, preferably involving more model versions, are required to clarify this.)

Besides GLUE AX and SNLI, we have assessed CRiPT on the semantically more demanding Argument Reasoning Comprehension task (Habernal et al., 2018) or the critical thinking assessment compiled in LogiQA (Liu et al., 2020), but found no performance increase compared to the base model.

6 Conclusion

This paper has taken a first step towards the creation of a critical thinking curriculum for neural language models. It presents a corpus of deductively valid, artificial arguments, and uses this artificial argument corpus to train and evaluate CRiPT – a Transformer language model based on GPT-2.

As our main finding, we observe strong transfer learning effects/generalization: Training CRiPT on a few central core schemes allows it to accurately complete conclusions of different types of arguments, too. The language models seem to connect and to generalize the core argument schemes in a correct way. Moreover, CRiPT is equally able to apply learned argument patterns beyond the domain it has been trained on, and there is evidence that generic language modeling skill facilitates the successful generalization of learned argument patterns as randomly initialized models fail to acquire any inference skill by critical thinking pre-training. (Accordingly, we expect our approach to scale to even larger versions of GPT-2.) These findings are consistent with previous work on rule reasoning (Clark et al., 2020). Moreover, CRiPT has been tested on different reasoning benchmarks. We obtain clear and promising results for the GLUE AX and SNLI benchmarks. All this suggests that there exist (learning-wise) fundamental reasoning skills in the sense that generic intermediary pre-training on texts which exemplify these skills leads to spillover effects and can improve performance on a broad variety of reasoning tasks. The synthetic argumentative texts might be a good starting point for building such a "critical thinking curriculum for language models."

There are different directions for advancing the approach adopted in this paper and further improving the general reasoning skill of neural language models:

- The syllogistic argument text corpus might be complemented with corpora of arguments that instantiate *different kinds of correct schemes*, e.g., propositional inference schemes, modal schemes, argument schemes for practical reasoning, complex argument schemes with intermediary conclusions or assumptions for the sake of the argument, etc. (Technically, we provide the infrastructure for doing so, as all this might be achieved through adjusting the argument corpus configuration file.)
- To succeed in NLI tasks, it doesn't suffice to understand 'what follows.' In addition, a system needs to be able to explicitly discern contradictions and *non sequiturs* (relations of logical independence). This suggests that the artificial argument corpus might be fruitfully supplemented with corpora of correctly identified aporetic clusters (Rescher, 1987) as well as corpora containing correctly diagnosed fallacies.
- In addition, the idea of curriculum learning for ML (Bengio et al., 2009) might be given a try. Accordingly, a critical thinking curriculum with basic exemplars of good reasoning would not only be used to fine-tune a pre-trained model, but would be employed as starting point for training a language model from scratch.

In conclusion, designing a critical thinking curriculum for pre-training neural language models seems to be a promising and worthwhile research program to pursue.

Acknowledgments

An earlier version of this work has been presented at Allen AIs Aristo Group, we profited from critical and constructive feedback.

References

Amanda Askell. 2020. Gpt-3: Towards renaissance models. In *Daily Nous Blog: Philosophers On GPT-3*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA. ACM.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tracey Bowell and Gary Kemp. 2014. *Critical Thinking: A Concise Guide*, 4th edition edition. Routledge, London.

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 (EMNLP)*. Association for Computational Linguistics.

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, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey 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.

Georg Brun and Gregor Betz. 2016. Analysing practical argumentation. In Sven Ove Hansson and Gertrude Hirsch-Hadorn, editors, *The Argumentative Turn in Policy Analysis. Reasoning about Uncertainty*, pages 39–77. Springer, Cham.

J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. *CSCW: Proceedings of the Conference on Computer-Supported Cooperative Work. Conference on Computer-Supported Cooperative Work, 2017*, page 1217–1230.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, pages 3882–3890.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Richard Feldman. 2014. *Reason and Argument*. Pearson, Harlow.

Alec Fisher. 2001. *Critical Thinking: An Introduction*. Cambridge University Press, Cambridge.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *CoRR*, abs/1812.08092.

Ben Gilburt and Mark Claydon. 2019. Examining gender bias in OpenAI's GPT-2 language model. *towardsdatascience.com*.

Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Christopher Pal. 2020. Measuring systematic generalization in neural proof generation with transformers.

Radu Cornel Guiaşu and Christopher W Tindale. 2018. Logical fallacies and invasion biology. *Biology & philosophy*, 33(5-6):34.

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.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1930–1940. Association for Computational Linguistics.

Sven Ove Hansson. 2004. Fallacies of risk. *Journal of Risk Research*, 7(3):353–360.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Daniel Kahneman. 2011. *Thinking, fast and slow*, 1st edition. Farrar, Straus and Giroux, New York.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Joe Lau and Jonathan Chan. 2020. Critical thinking web. *https://philosophy.hku.hk/think*.

D. D. Lewis, Y. Yang, T. Rose, and F. Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. *Proc. MRQA Workshop (EMNLP'19)*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2020. Knowledge-driven self-supervision for zero-shot commonsense question answering. *CoRR*, abs/2011.03863.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *CoRR*, abs/1909.08855.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *CoRR*, abs/2004.09456.

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.

SP Norris and RH Ennis. 1989. What is critical thinking. *The practitioner's guide to teaching thinking series: Evaluating critical thinking*, pages 1–26.

Fabio Paglieri. 2017. A plea for ecological argument technologies. *Philosophy & Technology*, 30(2):209–238.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *CoRR*, abs/1811.01088.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Preprint*.

Nicholas Rescher. 1987. Aporetic method in philosophy. *The Review of metaphysics*, 41(2):283–297.

Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8713–8721. AAAI Press.

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. Prover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 122–136. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4615–4629. Association for Computational Linguistics.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4505–4514. Association for Computational Linguistics.

Cass R Sunstein and Reid Hastie. 2015. *Wiser: getting beyond groupthink to make groups smarter*. Harvard Business Review Press, Boston.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. Quartz: An open-domain dataset of qualitative relationship questions. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5940–5945. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics - on what language model pre-training captures. *Trans. Assoc. Comput. Linguistics*, 8:743–758.

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.

J. Weston, A. Bordes, S. Chopra, and T. Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. *ICLR*.

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*, pages arXiv–1910.

A Appendix: Illustrative Examples of Synthetic Argumentative Texts

The following items are drawn from the artificial argument corpus and illustrate the synthetic

texts used to train and test CRiPT – specifically the various *domains* covered in the corpus. Links to the entire dataset and source code for generating synthetic arguments are released at https://github.com/debatelab/aacorpus.

Domain: **female_relatives**. *Base scheme group:* Generalized modus tollens. *Scheme variant:* base scheme. *Text:* It is not always easy to see who is related to whom – and in which ways. The following argument pertains to this question: To start with, Daisy is not a sister of Melissia. Now, being an ancestor of Kerstin is sufficient for being a sister of Melissia. Hence, it is false that Daisy is an ancestor of Kerstin.

Domain: **male_relatives**. *Base scheme group:* Hypothetical Syllogism 1. *Scheme variant:* negation_variant. *Text:* Is Fred a cousin of Robert? Is Joe related to Bob? In large families, it is sometimes difficult to keep track of all one's relatives. The following argument seeks to clarify some such relations: First of all, no schoolmate of Erik is a classmate of Andy. Next, whoever is not a classmate of Andy is a schoolmate of Marvin. We may conclude that every schoolmate of Erik is a schoolmate of Marvin.

Domain: **consumers_personalcare**. *Base scheme group:* Disjunctive Syllogism. *Scheme variant:* negation_variant. *Text:* Consumer research aims at understanding whether users of some products also tend to consume other ones, or not. The following argument seeks to clarify some such relations: Everyone who is an occasional purchaser of Bio Ionic shampoo is a rare consumer of The Body Shop soap, too. Every occasional purchaser of Bio Ionic shampoo is not a rare consumer of The Body Shop soap or a frequent consumer of Shiseido shampoo. It follows that everyone who is an occasional purchaser of Bio Ionic shampoo is a frequent consumer of Shiseido shampoo, too.

Domain: **chemical_ingredients**. *Base scheme group:* Generalized Contraposition. *Scheme variant:* complex_predicates. *Text:* Here comes a perfectly valid argument: No ingredient of Eyeshadow Quad is an ingredient of Midnight Black or an ingredient of Bubble Gum Laquer. We may conclude that no ingredient of Bubble Gum Laquer and no ingredient of Midnight Black is an ingredient of Eyeshadow Quad.

Domain: **football_fans**. *Base scheme group:* Generalized Dilemma. *Scheme variant:* base scheme. *Text:* Is Fred a fan of Liverpool? Are supporters of Real Madrid devotees of PSG? In European football, it is sometimes difficult to keep track of the mutual admiration and dislike. The following argument seeks to clarify some such relations: Every friend of FC Olexandriya is either a backer of The New Saints FC or an ex-fan of Olympique Lyonnais, or both. Everyone who is an ex-fan of Olympique Lyonnais is a devotee of RC Celta de Vigo, too. Everyone who is a backer of The New Saints FC is a devotee of RC Celta de Vigo, too. In consequence, being a devotee of RC Celta de Vigo is necessary for being a friend of FC Olexandriya.

Domain: **dinos**. *Base scheme group:* Modus barbara. *Scheme variant:* base scheme. *Text:* Consider the following argument: If someone is a predator of Iguanodon, then they are a prey of Stegosaurus. Parasaurolophus is a predator of Iguanodon. Thus, Parasaurolophus is a prey of Stegosaurus.

Domain: **philosophers**. *Base scheme group:* Hypothetical Syllogism 3 *Scheme variant:* negation_variant *Text:* Here comes a perfectly valid argument: If someone is not a teacher of Diodorus of Adramyttium, then they are a teacher of Dexippus. Moreover, someone is a student of Alexicrates and not a teacher of Dexippus. Thus, someone is a student of Alexicrates and a teacher of Diodorus of Adramyttium.

B Appendix: Training Parameters

We train differently sized versions of GPT-2 with causal language modeling objective (using default training scripts by Wolf et al. (2019)) on each of the 12 enhanced, differently sized training sets. This gives us 36 fine-tuned CRiPT models plus the three BASE models to evaluate. Unless explicitly stated otherwise, the main article reports results of the 762M parameter model trained on 72K items. We train the models on 8 GPUs for 2 epochs with batch size = 2, learning rate = 5×10^{-5}, gradient accumulation steps = 2, and default parameters of the HuggingFace implementation otherwise (Wolf et al., 2019).

C Appendix: Performance Metrics on Different Argument Schemes

Fig. 5 displays CRiPT's accuracy on conclusion completion tasks on specific argument schemes. Its subplots are arranged in a grid that mirrors the organisation of argument schemes as presented in the main article. Each subplot visualizes the abil-

ity of CRiPT to correctly complete arguments of the corresponding scheme (given the out-of-sample test dataset). Reported accuracy values that fall within gray background areas are attained by models which have seen the corresponding scheme during training. Vice versa, thick lines on white background visualize model performance on unknown schemes. Fig. 5 reveals, first of all, that even the BASE models (only pre-training, no fine-tuning) display a significant ability to correctly complete conclusions of some kinds of arguments. For example, GPT-2-762M achieves 50% accuracy (*split* task) in completing contrapositions, 30% accuracy in completing generalized modus ponens, and still 20% accuracy in completing disjunctive syllogism and dilemma arguments. These findings further corroborate the hypothesis that NLMs learn (basic) linguistic and reasoning skills "on the fly" by training on a large generic corpus (Radford et al., 2019).

D Appendix: Performance Metrics for Differently Sized Training Sets

Fig. 6 displays accuracy values on conclusion completion tasks for models trained on differently sized datasets.

Fig. 7 reports perplexity and NLU accuracy metrics for models trained on differently sized datasets.

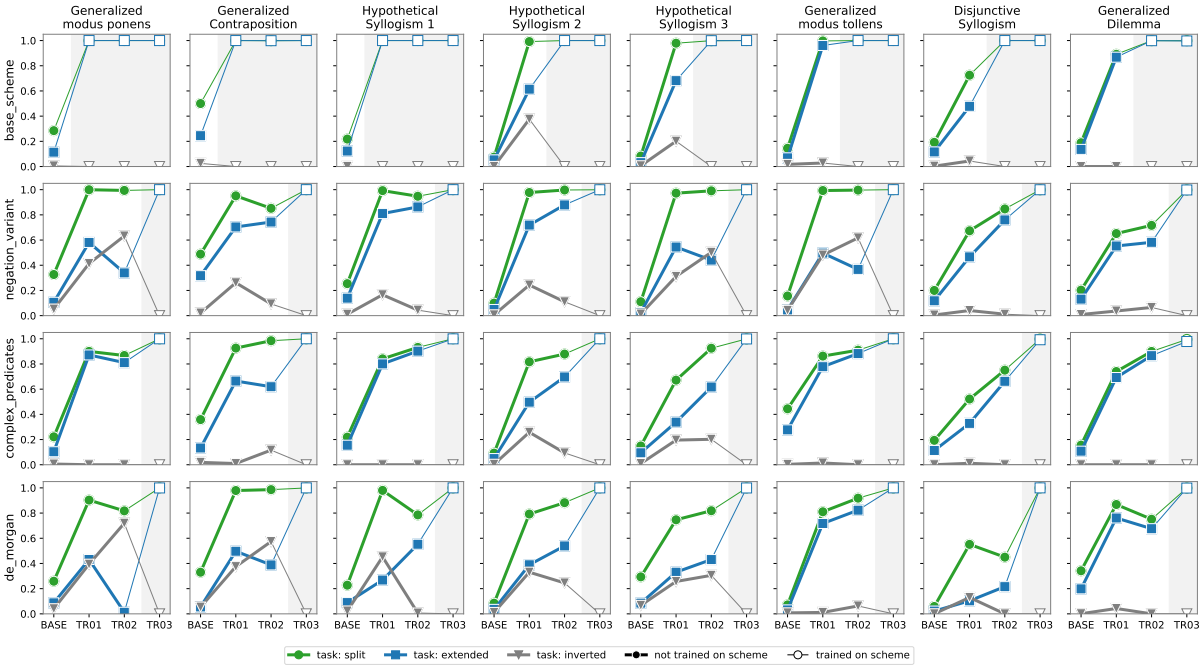


Figure 5: Accuracy of CRiPT in three conclusion completion tasks and on different test datasets (out of sample, paraphrased, out of domain) by argument scheme.

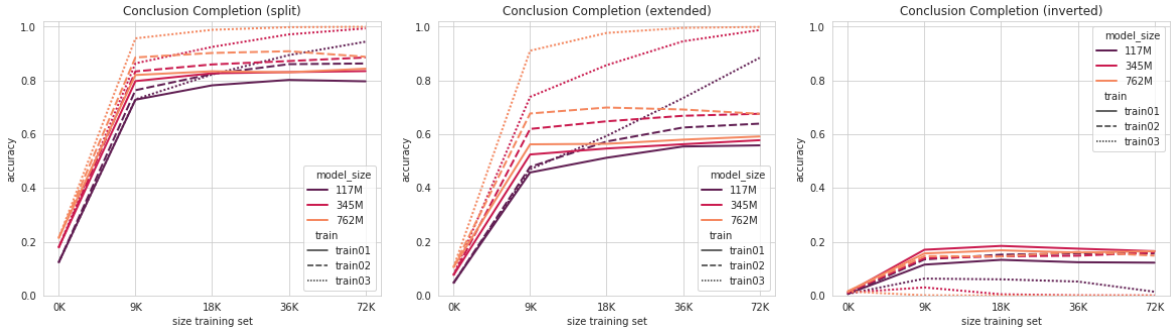


Figure 6: Accuracy on three conclusion completion tasks as a function of training corpus size.

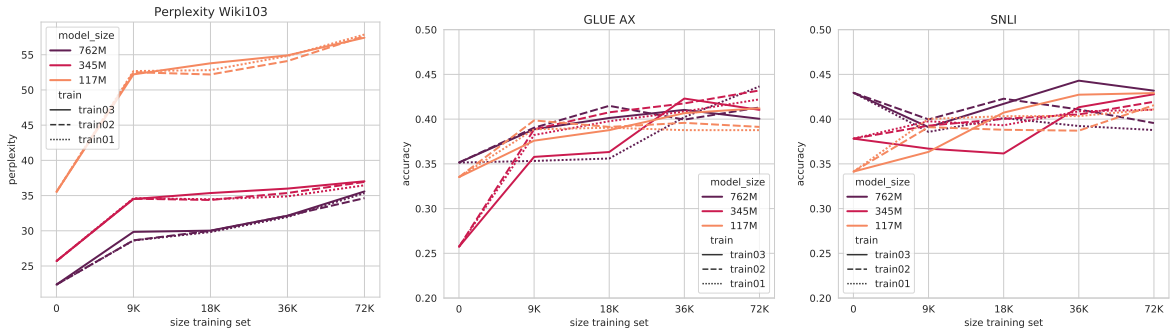


Figure 7: Perplexity and NLI metrics as a function of training corpus size.

Do Natural Language Explanations Represent Valid Logical Arguments? Verifying Entailment in Explainable NLI Gold Standards

Marco Valentino[†], **Ian Pratt-Hartman**[†], **André Freitas**[†‡]
Department of Computer Science, University of Manchester, United Kingdom[†]
Idiap Research Institute, Switzerland[‡]
{marco.valentino,ian.pratt,andre.freitas}
@manchester.ac.uk

Abstract

An emerging line of research in Explainable NLP is the creation of datasets enriched with human-annotated explanations and rationales, used to build and evaluate models with step-wise inference and explanation generation capabilities. While human-annotated explanations are used as ground-truth for the inference, there is a lack of systematic assessment of their consistency and rigour. In an attempt to provide a critical quality assessment of Explanation Gold Standards (XGSs) for NLI, we propose a systematic annotation methodology, named *Explanation Entailment Verification* (EEV), to quantify the logical validity of human-annotated explanations.

The application of EEV on three mainstream datasets reveals the surprising conclusion that a majority of the explanations, while appearing coherent on the surface, represent logically invalid arguments, ranging from being incomplete to containing clearly identifiable logical errors. This conclusion confirms that the inferential properties of explanations are still poorly formalised and understood, and that additional work on this line of research is necessary to improve the way Explanation Gold Standards are constructed.

Worldtree

Question: Which of the following characteristics would best help a tree survive the heat of a forest fire?
[A] large leaves [B] shallow roots
[*C] thick bark [D] thin trunks

Explanation:
Protecting something means preventing harm.
Fire causes harm to trees, forests, and other living things.
Thickness is a measure of how thick an object is.
A tree is a kind of living thing.

QASC

Question: Differential heating of air can be harnessed for what?
[*A] electricity production [B] erosion prevention
[C] transfer of electrons [D] reduce acidity of food

Explanation:
Differential heating of air produces wind.
Wind is used for producing electricity.

e-SNLI

Premise: A man in an orange vest leans over a pickup truck.
Hypothesis: A man is touching a truck.
Label: entailment

Explanation:
Man leans over a pickup truck implies that he is touching it.

Figure 1: Does the answer logically follow from the explanation? While step-wise explanations are used as ground-truth for the inference, there is a lack of assessment of their consistency and rigour. We propose EEV, a methodology to quantify the logical validity of human-annotated explanations.

1 Introduction

Explanation Gold Standards (XGSs) are emerging as a fundamental enabling tool for step-wise and explainable Natural Language Inference (NLI). Resources such as WorldTree (Xie et al., 2020; Jansen et al., 2018), QASC (Khot et al., 2020), among others (Wiegreffe and Marasović, 2021; Thayaparan et al., 2020b; Bhagavatula et al., 2020; Camburu et al., 2018) provide a corpus of linguistic evidence on how humans construct explanations that are perceived as plausible, coherent and complete.

Designed for tasks such as Textual Entailment (TE) and Question Answering (QA), these reference datasets are used to build and evaluate models with step-wise inference and explanation generation capabilities (Valentino et al., 2021; Cartuyvels et al., 2020; Kumar and Talukdar, 2020; Rajani et al., 2019). While these explanations are used as ground-truth for the inference, there is a lack of systematic assessment of their consistency and rigour, introducing inconsistency biases within the models.

This paper aims to provide a critical quality assessment of Eplanation Gold Standards for NLI in terms of their logical inference properties. By

Proceedings of the 14th International Conference on Computational Semantics, pages 76–86
June 17–18, 2021.

systematically translating natural language explanations into corresponding logical forms, we induce a set of recurring logical violations which can then be used as testing conditions for quantifying quality and logical consistency in the annotated explanations. More fundamentally, the paper reveals the surprising conclusion that a majority of the explanations present in explanation gold standards contain one or more major logical fallacies, while appearing to be coherent on the surface. This study reveals that the inferential properties of explanations are still poorly formalised and understood.

The main contributions of this paper can be summarised as:

1. Proposal of a systematic methodology, named *Explanation Entailment Verification* (EEV), for analysing the logical consistency of NLI explanation gold-standards.

2. Validation of the quality assessment methodology for three contemporary and mainstream reference XGSs.

3. The conclusion that most of the annotated human-explanations in the analysed samples represent logically invalid arguments, ranging from being incomplete to containing clearly identifiable logical errors.

2 Related Work

An emerging line of research in Explainable NLP is focused on the creation of datasets enriched with human-annotated explanations and rationales (Wiegreffe and Marasović, 2021). These resources are often adopted as Explanation Gold Standards (XGSs), providing additional supervision for training and evaluating explainable models capable of generating natural language explanations in support of their predictions (Valentino et al., 2021, 2020; Kumar and Talukdar, 2020; Cartuyvels et al., 2020; Thayaparan et al., 2020a; Rajani et al., 2019).

XGSs are designed to support Natural Language Inference, asking human-annotators to transcribe the reasoning required for deriving the correct prediction (Thayaparan et al., 2020b). Despite the popularity of these datasets, and their application for measuring explainability on tasks such as Textual Entailment (Camburu et al., 2018), Multiple-choice Question Answering (Xie et al., 2020; Jhamtani and Clark, 2020; Khot et al., 2020; Jansen et al., 2018), and other inference tasks (Wang et al., 2020; Ferreira and Freitas, 2020b,a; Bhagavatula et al., 2020), little has been done to provide a clear understanding on the nature and the quality of the reasoning encoded in the explanations.

Previous work on explainability evaluation has mainly focused on methods for assessing the quality and faithfulness of explanations generated by deep learning models (Camburu et al., 2020; Subramanian et al., 2020; Kumar and Talukdar, 2020; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Our work is related to this research, but focuses instead on the resources on which explainable models are trained. In that sense, this paper is more aligned to gold standard evaluation methods, which aim to design systematic approaches to qualify the content and the inference capabilities involved in mainstream NLP benchmarks (Lewis et al., 2021; Bowman and Dahl, 2021; Schlegel et al., 2020; Ribeiro et al., 2020; Pavlick and Kwiatkowski, 2019; Min et al., 2019). However, to the best of our knowledge, none of these methods have been adopted to provide a critical assessment of human-annotated explanations present in XGSs.

3 Explanation Gold Standards

Given a generic classification task T, an Explanation Gold Standard (XGS) is a collection of distinct instances of T, $XGS(T) = \{I_1, I_2, \ldots, I_n\}$, where each element of the set, $I_i = \{X_i, s_i, E_i\}$, includes a problem formulation X_i, the expected solution s_i for X_i, and a human-annotated explanation E_i.

In general, the nature of the elements in a XGS can vary greatly according to the task T under consideration. In this work, we restrict our investigation to Natural Language Inference (NLI) tasks, such as Textual Entailment and Question Answering, where problem formulation, expected solution, and explanations are entirely expressed in natural language.

For this class of problems, the explanation is typically a composition of sentences, whose role is to describe the reasoning required to arrive at the final solution. As shown in the examples depicted in Figure 1, the explanations are constructed by human annotators transcribing the commonsense and world knowledge necessary for the correct answer to hold. Given the nature of XGSs for NLI, we hypothesise that a human-annotated explanation represents a valid set of premises from which the expected solution logically follows.

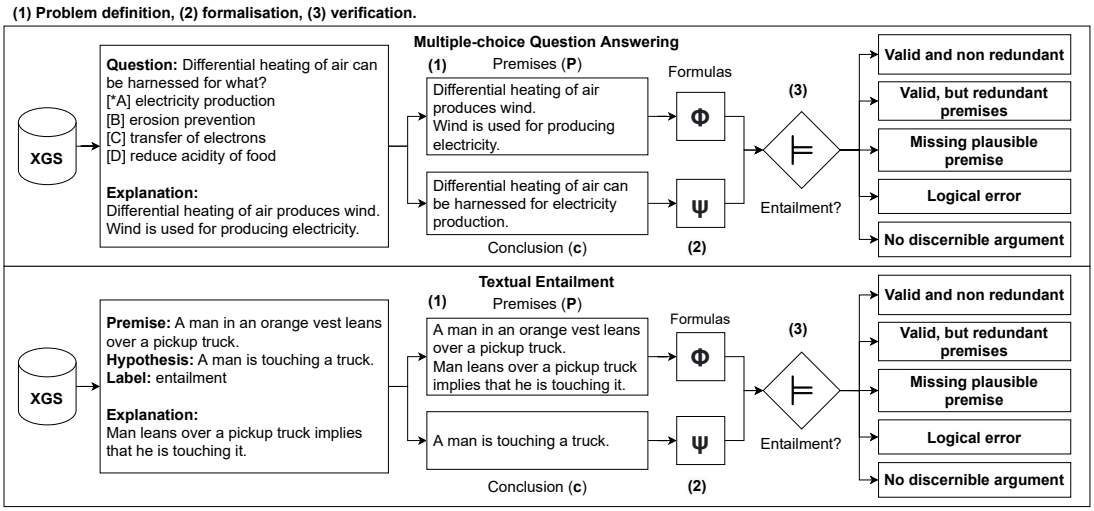


Figure 2: Overview of the Explanation Entailment Verification (EEV) applied to different NLI problems. EEV takes the form of a multi-label classification problem where, for a given NLI problem, a human annotator has to qualify the validity of the inference process described in the explanation through a pre-defined set of classes.

In order to validate or reject this hypothesis, we design a methodology aimed at evaluating XGSs in terms of logical entailment, quantifying the extent to which human-annotated explanations actually entail the final answer.

4 Explanation Entailment Verification

We present a methodology, named Explanation Entailment Verification (EEV), aimed at quantifying and assessing the quality of human-annotated explanations in XGS for NLI tasks, in terms of their logical inference properties.

To this end, we design an annotation framework that takes the form of a multi-label classification problem defined on a XGS. Specifically, the goal of EEV is to label each element in a XGS, $I_i = \{X_i, s_i, E_i\}$, using one of a predefined set of classes qualifying the validity of the inference process described in the explanation E_i.

Figure 2 shows a schematic representation of the annotation pipeline. One of the challenges involved in the design of a standardised methodology for EEV is the formalisation of an annotation task that is applicable to NLI problems with different shapes, such as Textual Entailment (TE) and Multiple-choice Question Answering (MCQA). To minimise the ambiguity in the annotation and make it independent of the specific NLI task, we define a methodology composed of three major steps: (1) *problem definition*; (2) *formalisation*; and (3) *verification*.

In the problem definition step, each example I_i in the XGS is translated into an entailment form ($P \models c$), identifying a set of sentences P representing the premises for the entailment, and a single sentence c representing its conclusion. As illustrated in Figure 2, this step defines an entailment problem with a single surface form that allows abstracting from the NLI task under investigation.

In the formalisation step, the sentences in P and c are translated into a logical form ($\Phi \models \psi$). Specifically, the formalisation is performed using event-based semantics, in which verbs correspond to event-types, and their objects to semantic roles (additional details on the formalism are provided in section 4.3). This step aims to minimise the ambiguity in the interpretation of the meaning of the sentences, supporting the annotators in the identification of logical errors and gaps in the explanations, and maximise the inter-annotator agreement in the downstream verification task.

The final step corresponds to the actual multi-label classification problem. Specifically, the annotators are asked to verify whether the formalised set of premises Φ entails the conclusion ψ ($\Phi \models \psi$) and to classify the explanation in the corresponding example $I_i = \{X_i, s_i, E_i\}$ selecting one of the following classes: (1) *Valid and non redundant*; (2) *Valid, but redundant premises*; (3) *Missing plausible premise*; (4) *Logical error*; (5) *No discernible argument*. The classes are mutually exclusive: each example can be assigned to one and only one label.

After EEV is performed for each instance in the dataset, the frequencies of the classification labels can be adopted to estimate and evaluate the

overall entailment properties of the explanations in the XGS under consideration.

4.1 Problem definition

The problem definition step consists in the identification of the sentences in $I_i = \{X_i, s_i, E_i\}$ that will compose the set of premises P and the conclusion c for the entailment problem $P \models c$.

Here, we describe the procedure adopted for translating a specific NLI task into the entailment problem of interest given its original surface form. In particular, we employ two different translation procedures for Textual Entailment (TE) and Multiple-choice Question Answering (MCQA) problems.

Textual Entailment (TE). For a TE task, the problem formulation X_i is generally composed of two sentences, p and h, representing a premise and a hypothesis (see e-SNLI in figure 1). Each example in a TE task can be classified using one of the following labels: *entailment*, *neutral*, and *contradiction* (Bowman et al., 2015). In this work, we focus on examples where the expected solution s_i is *entailment*, implying that the hypothesis h is a consequence of the premise p. Therefore, to define the entailment verification problem, we simply include the premise p in P and consider the hypothesis h as a the conclusion c. For this class of problems, the explanation E_i describes additional factual knowledge necessary for the entailment $p \models h$ to hold (Camburu et al., 2018). Specifically, the sentences in E_i can be interpreted as a further set of premises for the entailment verification problem and are included in P.

Multiple-choice Question Answering (MCQA). In the case of MCQA, X_i is typically composed of a question $Q_i = \{c_1, \ldots, c_n, q\}$, and a set of mutually exclusive candidate answers $A_i = \{a_1, \ldots, a_m\}$ (see QASC and Worldtree in figure 1). In this case, the expected label s_i corresponds to one of the candidate answers in A_i (Jansen et al., 2018; Khot et al., 2020). Q_i can include a set of introductory sentences $c_1, \ldots, c_n$ acting as a context for the question q. We consider each sentence c_i in the context as a premise for q and include it in P. Similarly to TE, we interpret the explanation E_i for a MCQA example as a set of premises that entails the correct answer s_i. Therefore, the sentences in E_i are included in P. The question q takes the form of an elliptical assertion, and the candidate answers are possible substitutions for the ellipsis. Therefore, to derive the conclusion c, we adopt the correct answer s_i as a substitution for the ellipsis in q. Details on the formalisation adopted for MCQA problems are described in section 4.3.

4.2 Verification

In the verification step, the annotators adopt the formalised set of premises Φ and conclusion ψ to classify the entailment problem in one of the following categories:

1. **Valid and non-redundant:** The argument is formally valid, and all premises are required for the derivation.

2. **Valid, but redundant premises:** The argument is formally valid, but some premises are not required for the derivation. This includes the cases where more than one premise is present, and the conclusion simply repeats one of the premises.

3. **Missing plausible premise:** The argument is formally invalid, but would become valid on addition of a reasonable premise, such as, for example, *"If x affects y, then a change to x affects y"*, or *"If x is the same height as y and y is not as tall as z then x is not as tall as z"*.

4. **Logical error:** The argument is formally invalid, apparently as a result of confusing *"and"* and *"or"* or *"some"* and *"all"*, or of illicitly changing the direction of an implication.

5. **No discernible argument:** The argument is invalid, no obvious rescue exists in the form of a missing premise, and no simple logical error can be identified.

4.3 Formalisation

In this section, we describe an example of formalisation for a MCQA problem. A typical multiple-choice problem is a triple consisting of a *question* Q together with a set of *candidate answers* $A_1, \ldots, A_m$. It is understood that Q takes the form of a elliptical assertion, and the candidate answers are possible substitutions for the ellipsis. The task is to determine which of the candidate answers would result in an assertion entailed by some putative knowledge-base. The corpora investigated feature a list of multiple-choice textual entailment problems together, in each case, with a specification of a correct answer and an *explanation* in the form of a set of assertions Φ from the knowledge

base providing a justification for the answer. For example, the following problem together with its resolution is taken from the Worldtree corpus (Jansen et al., 2018).

Question: A group of students are studying bean plants. All of the following traits are affected by changes in the environment except ...

Candidate answers: [A] leaf color. [B] seed type. [C] bean production. [D] plant height.

Correct answer: B

Explanation: (i) The type of seed of a plant is an inherited characteristic; (ii) Inherited characteristics are the opposite of learned characteristics; acquired characteristics; (iii) An organism's environment affects that organism's acquired characteristics; (iv) A plant is a kind of organism; (v) A bean plant is a kind of plant; (vi) Trait is synonymous with characteristic.

In formalising such problems, we represent the question as a sentence of first-order logic featuring a schematic formula variable P (corresponding to the ellipsis), and the candidate answers as first-order formulas. In the above example, we assume that the essential force of the question to find a characteristic of plants *not* affected by those plants' environments. That is, we are asked for a P making the schematic formula

$$\begin{aligned}\forall xyzwe(&\text{bnPlnt}(x) \wedge \text{env}(y,x) \wedge \\ &\text{changeIn}(z,y) \wedge \text{trait}(w,x) \wedge \text{affct}(e) \wedge \\ &\text{agnt}(e,z) \wedge P \rightarrow \neg\text{ptnt}(e,w)). \quad (1)\end{aligned}$$

into a true statement. We formalise the correct answer (B) by the atomic formula $\text{sdTp}(w,x)$ "w is the seed type of x", with the other candidate answers formalised similarly. In choosing predicates for formalisation, we typically render common noun-phrases using predicates, taking these to be relational if the context demands (e.g. "environment/seed type *of* a plant x"). In addition, we typically render verbs as predicates whose arguments range over eventualities (events, processes, etc.), related to their participants via a standard list of binary "semantic role" predicates (agent, patient, theme) etc. Thus, to say that "x affects y" is to report the existence of an eventuality e of type "affecting", such that x is the agent of e and y its patient. This approach, although somewhat strained in many general contexts, aids standardization and, more importantly, also makes it easier to deal with adverbial phrases. Of course, many choices in formalisation strategy inevitably remain.

The knowledge-base excerpt Φ is formalised straightforwardly as a finite set of first-order formulas, following the same general rendering policies. In the case of the above example, sentences (i), (ii) and (iv)–(vi) in Φ might be formalised as:

$$\begin{aligned}&\forall xy(\text{plnt}(x) \wedge \text{sdTp}(y,x) \rightarrow \text{char}(y,x) \wedge \text{inhtd}(y)) \\ &\forall xy(\text{char}(x,y) \wedge \text{inhtd}(x) \rightarrow \neg\text{acqrd}(x)) \\ &\forall x(\text{plnt}(x) \rightarrow \text{orgnsm}(x)) \\ &\forall x(\text{bnPlnt}(x) \rightarrow \text{plnt}(x)) \\ &\forall xy(\text{trait}(x,y) \leftrightarrow \text{char}(x,y)),\end{aligned}$$

with the more complicated sentence (iii) formalised as

$$\begin{aligned}\forall xyw(&\text{orgnsm}(x) \wedge \text{env}(y,x) \wedge \\ &\text{char}(w,x) \wedge \text{acqrd}(w) \rightarrow \\ &\exists e(\text{affct}(e) \wedge \text{agnt}(e,y) \wedge \text{ptnt}(e,w)))\end{aligned} \quad (2)$$

Denoting by ψ the result of substituting $\text{sdTp}(w,x)$ for P in (1), we ask ourselves: Does Φ entail ψ? A moment's thought shows that it does not. At the very least, statement (iii) in the explanation, whose *prima facie* formalisation is (2), must instead be read as asserting that an organism's environment affects *only* that organism's acquired characteristics, that is to say:

$$\begin{aligned}\forall xyw(&\text{orgnsm}(x) \wedge \text{env}(y,x) \wedge \text{char}(w,x) \wedge \\ &\exists e(\text{affct}(e) \wedge \text{agnt}(e,y) \wedge \text{ptnt}(e,w)) \rightarrow \\ &\text{acqrd}(w)).\end{aligned} \quad (3)$$

This is not unreasonable, of course. Generalizations in natural language are notoriously vague as to the direction of implication; let Φ' be the result of substituting (3) for (2) in Φ. Does Φ' entail ψ? Again, no. The problem this time is that, model-theoretically speaking, just because something is affected by a *change in* its environment, that does not mean to say it is affected by its environment. An assertion to the effect that it is would have to be postulated:

$$\begin{aligned}\forall xyzw(&\text{env}(y,x) \wedge \text{changeIn}(z,y) \wedge \\ &\exists e(\text{affct}(e) \wedge \text{agnt}(e,z) \wedge \text{ptnt}(e,w)) \rightarrow \\ &\exists e(\text{affct}(e) \wedge \text{agnt}(e,y) \wedge \text{ptnt}(e,w))).\end{aligned}$$

Let Φ'' be the result of augmenting Φ' in this way. Then Φ'' does indeed entail ψ.

Feature	Worldtree	QASC	e-SNLI
Task	MCQA	MCQA	TE
Multi-hop	yes	yes	no
Crowd-sourced	no	yes	yes
Explanation type	generated + composed	composed	generated
Avg. number of sentences	6	2	1

Table 1: Features of the datasets selected for the Explanation Entailment Verification (EEV).

Applying a general principle of charity, it is reasonable to take the interpretation of the explanation to be given by Φ'. However, the additional premise required to obtain Φ'' seems to have been forgotten. Although not a logical truth, it has the status of a plausible general principle of the kind that is frequently explicitly articulated in the Worldtree database. Therefore, we classify this example as a *missing plausible premise*.

5 Corpus Analysis

We employ EEV to analyse a set of contemporary XGSs designed for Textual Entailment and Multiple-choice Question Answering.

In the following sections, we describe the methodology adopted for extracting a representative sample from the selected XGSs, and for implementing the annotation pipeline efficiently. Finally, we present the results of the annotation, reporting the frequency of each entailment verification class and presenting a list of qualitative examples to provide additional insights on the logical properties of the analysed explanations.

5.1 Selected Datasets

We select three contemporary XGSs with different and complementary characteristics. In particular, we apply our methodology to two MCQA datasets (Worldtree (Jansen et al., 2018), QASC(Khot et al., 2020)) and one TE benchmark (e-SNLI (Camburu et al., 2018)).

The main features of the selected XGSs are reported in Table 1. *Multi-hop* indicates whether the problem requires step-wise reasoning, combining more than one sentence to compose the final explanation. *Crowd-sourced* indicates whether the resource is curated using standard crowd-sourcing platforms. *Explanation type* represents the methodology adopted to construct the explanations. *Generated* means that the sentences in the explanations are entirely created by human annotators. On the other hand, *composed* means that the sentences are retrieved from an external knowledge resource. Finally, the last row reports the *average number of sentences* composing the explanations.

5.2 Annotation Task

The bottleneck of the annotation framework lies in the formalisation phase, which is generally time consuming and requires trained experts in the field. In order to make the application of EEV efficient in practice, we extract a sub-set of $n = 100$ examples from each XGS (Worldtree, QASC, and e-SNLI). To maximise the representativeness of the explanations in the subset, given a fixed size n, we combine a set of sampling methodologies with effect size analysis. The details of the sampling methodology are described in section 5.3 while the results are presented in section 5.4. Code and data adopted for the experiments are available online [1].

The extracted examples are randomly assigned to 2 annotators with an overlap of 20 instances to compute the inter-annotator agreement. All the annotators are active researchers in the field of Natural Language Processing and Computational Semantics. Table 2 reports the inter-annotator agreement achieved on each dataset separately. Overall, we observe an average of 72% accuracy in the multi-label classification task, computed considering the percentage of overlaps between the final entailment verification classes chosen by the annotators.

5.3 Sampling Methodology

To maximise the representativeness of the explanations for the subsequent annotation task, while analysing a fixed number n of examples for each dataset, we combine a set of sampling methodologies with effect size analysis. In this section, we describe the sampling techniques adopted for each dataset.

A stratified sampling methodology has been adopted for the Worldtree corpus (Xie et al., 2020; Jansen et al., 2018). The stratified sampling con-

[1] https://github.com/ai-systems/explanation-entailment-verification/

sists in partitioning the dataset using a set of classes and performing random sampling from each class independently. This strategy guarantees that the same amount of examples is extracted from each class. The stratified technique requires the classes to be collectively exhaustive and mutually exclusive – i.e, each example has to belong to one and only one class. To apply stratified sampling on Worldtree, we consider the high-level topics introduced in (Xu et al., 2020), which are used to classify each question in the dataset according to one of the following categories: Life, Earth, Forces, Materials, Energy, Scientific Inference, Celestial Objects, Safety, Other. The same technique cannot be applied to e-SNLI (Camburu et al., 2018) and QASC (Khot et al., 2020) since the examples in these datasets are not partitioned using any abstract set of classes. In this case, therefore, we use random sampling on the whole dataset to extract a fixed number n of examples.

Once a fixed number of examples n is extracted from each dataset, we consider the annotated explanation sentences of each example to verify whether the extracted set of explanations is representative of the whole dataset. To perform this analysis, we assume the predicates in the explanation sentences to be the expression of the type of knowledge of the whole explanation. Therefore, we consider the extracted sample of explanations representative if the distribution of predicates in the sample is correlated with the same distribution in the whole dataset. To this end, we compute the frequencies of the verbs appearing in the explanation sentences from the extracted sub-set and original dataset separately. Subsequently, we compare the frequencies in the sub-sample with the frequencies in the whole dataset computing a Pearson correlation coefficient. In this case, a coefficient greater than .7 indicates a strong correlation between the types of explanations in the sample and the types of explanations in the original dataset. After running the sampling for t times independently, we select the subset of explanations for each dataset with the highest Pearson correlation coefficient. Table 3 reports the Pearson correlation for the subsets adopted in our analysis with fixed sample size $n = 100$.

5.4 Results

The quantitative analysis presented in this section aims to empirically assess the hypothesis that human-annotated explanations in XGSs constitute

Dataset	Agreement Accuracy
Worldtree	.70
QASC	.70
e-SNLI	.75

Table 2: Inter-annotator agreement computed in terms of accuracy in the multi-label classification task considering the first annotator as a gold standard.

Dataset	Correlation Coefficient
Worldtree	.964
QASC	.958
e-SNLI	.987

Table 3: Effect size analysis of the samples extracted from each XGS for the downstream EEV annotation.

valid and non-redundant logical arguments for the expected answers. We report the quantitative results of the explanation entailment verification in Table 4. Specifically, the table reports the percentage of the frequency of each verification class in the analysed samples. The column *AVG* reports the average for each class.

Overall, we observe that the results of the annotation task tend to reject our research hypothesis, with an average of only 20.42% of analysed explanations being classified as *valid and non redundant* arguments. When considering also *valid, but redundant* explanations (21.91%), the average percentage of valid arguments reaches a total of 42.33%. Therefore, we can conclude that the majority of the explanations represent invalid arguments (57.66%).

We observed that the majority of invalid arguments are classified as *missing plausible premise*. This finding implies that a significant percentage of annotated explanations are incomplete arguments (26.00%), that can be made valid on addition of a reasonable premise. We attribute this result to the tendency of human explainers to take for granted part of the world knowledge required in the explanation (Walton, 2004).

A lower but significant percentage of explanations contain identifiable logical errors (11.19%), which result from confusing the set of quantifiers and logical operators, or from illicitly changing the direction of an implication. Similarly, 20.47% of the explanations where labeled as *no discernible arguments*, where no obvious premise can be added to make the argument valid and no simple logical error can be detected. This result can be attributed partly to natural errors occurring in a gold standard

Entailment Verification Class	Worldtree	QASC	e-SNLI	AVG
Valid and non-redundant	12.24	17.65	**31.37**	20.42
Valid, but redundant premises	26.53	7.84	**31.37**	21.91
Missing plausible premise	**38.78**	21.57	17.65	**26.00**
Logical error	6.12	17.65	9.80	11.19
No discernible argument	16.33	**35.29**	9.80	20.47
Valid argument	38.77	25.49	**62.74**	42.33
Invalid argument	61.23	**74.51**	37.25	**57.66**

Table 4: Results of the application of EEV for each entailment verification category.

creation process, partly to the effort required for human-annotators to identify logical fallacies in their explanations. In the remaining of this section, we analyse the results obtained on each XGS.

Worldtree. The analysed sample contains the highest percentage of incomplete arguments, with a total of 38.78% explanations classified as *missing plausible premise*. This result can be explained by the fact that the questions in Worldtree require complex forms of reasoning, facilitating the construction of arguments containing implicit world knowledge and missing premises. At the same time, the dataset contains the smallest percentage of logical errors (6.12%). We attribute this outcome to the fact that Worldtree is not crowd-sourced, implying that the quality of the annotated explanations is more easily controllable using internal verification methods.

QASC. This XGS contains the highest rate of invalid arguments (62.74%), with 35.29% of the explanations classified as *no discernible argument*. One of the factors contributing to these results might be related to the length of the constructed explanations, which is limited to 2 facts extracted from a predefined corpus of sentences. The high rate of no discernible arguments and missing premises (35.29% and 21.57% respectively) suggests that the majority of the questions require additional world knowledge and more detailed explanations. This conclusion is also supported by the percentage of *valid, but redundant* arguments, which is the lowest among the analysed samples (7.84%). Finally, the highest rate of logical errors (17.65%) might be due to a combination of factors, including the complexity of the question answering task and the adopted crowd-sourcing mechanism, which prevent a thorough quality assessment.

e-SNLI. The sample includes the highest percentage of valid arguments with a total of 31.37%. However, we noticed that the complexity of the reasoning involved in e-SNLI is generally lower than Worldtree and QASC, with most of the textual entailment problems being an example of *monotonicity reasoning*. This observation is supported by the highest percentage of *valid, but redundant* cases (31.37%), where the explanation simply repeats the content of the conclusion. This occurrs quite often for examples of lexical entailment, where the words in the conclusion are a subset of the words in the premise. The lexical entailment instances, in fact, do not require any additional world knowledge, making any attempt of constructing an explanation redundant. Despite these characteristics, our evaluation suggests that a significant percentage of arguments are invalid (37.25%). Again, this percentage might be the results of different factors, including the errors produced by the crowd-sourcing process.

Table 5 reports a set of representative cases extracted from the evaluated samples. For each entailment verification class, we report an example extracted from the XGS with the highest percentage of instances in that class.

5.5 Contrastive Explanations

Previous studies highlight the fact that explanations are *contrastive* in nature, that is, they describe why an event *P* happened instead of some counterfactual event *Q* (Miller, 2019; Lipton, 1990). Following this definition, we perform an additional analysis to verify whether the explanations contained in MCQA datasets are *contrastive* with respect to the wrong candidate answers – i.e., the explanation supports the validity of the correct answer while excluding the set of alternative choices. In order to quantify this aspect, we asked the annotators to label the questions with more than one plausible answer, whose explanations do not mention any discriminative commonsense or world knowledge that explains why the gold answer is correct instead of the alternative choices.

Problem Formulation	Explanation	XGS
Valid and non-redundant (20.42%)		
Premise: A smiling woman is playing the violin in front of a turquoise background. **Hypothesis:** A woman is playing an instrument.	A violin is an instrument.	e-SNLI
Valid, but redundant premises (21.91%)		
Premise: Four people are bandaging a head wound. **Hypothesis:** People are bandaging an injured head.	People are bandaging an injured head wound.	e-SNLI
Missing plausible premise (26.00%)		
Question: A group of students are studying bean plants. All of the following traits are affected by changes in the environment except [A] Leaf color [*B] Seed type [C] Bean production [D] Plant height	The type of seed of a plant is an inherited characteristic. Inherited characteristics are the opposite of learned characteristics; acquired characteristics. An organism's environment affects that organism's acquired characteristics. A plant is a kind of organism. Trait is synonymous with characteristic.	Worldtree
Logical error (11.19%)		
Question: What group of animals do chordates belong to? [A] graptolites [B] more abundant [C] warm-blooded [D] four limbs [E] epidermal [*F] Vertebrates [G] animals [H] insects	Chordates have a complete digestive system and a closed circulatory system. Vertebrates have a closed circulatory system.	QASC
No discernible argument (20.47%)		
Question: What do plants require for reproduction? [A] energy [B] nutrients [C] bloom time [*D] animals [E] sunlight [F] Energy. [G] food [H] hormones	Plants require seed dispersal for reproduction. Seeds are probably dispersed by animals.	QASC

Table 5: Examples of explanations classified with different entailment verification categories.

Dataset	Non contrastive explanations
Worldtree	26.53
QASC	**49.02**

Table 6: Percentage of explanations in the MCQA sample labeled as non contrastive.

The results of this experiment are reported in Table 6. Overall, we found that a significant percentage of explanations are labeled as non contrastive. This outcome is particularly evident for QASC. We attribute these results to the presence of multi-adversary answer choices in QASC, which are generated automatically to make the dataset more challenging for language models. However, we found that this mechanism can produce questions with more than one plausible correct answer, which can cause the explanation to loose its contrastive function (see QASC examples in Table 5).

6 Conclusion and Future Work

This paper proposed a systematic annotation methodology to quantify the logical validity of human-annotated explanations in Explanation Gold Standards (XGSs). The application of the framework on three mainstream datasets led us to the conclusion that a majority of the explanations represent logically invalid arguments, ranging from being incomplete to containing clearly identifiable logical errors.

The main limitation of the framework lies in the scalability of its current implementation, which is generally time consuming and requires trained semanticists. One way to improve its efficiency is to explore the adoption of supporting tools for the formalisation, such as semantic parsers and/or automatic theorem provers.

Despite the current limitations, this study offers some important pointers for future work. On the one hand, the results suggest that logical errors can be reduced by a careful design of the gold standard, such as authoring explanations with internal verification strategies or reducing the complexity of the reasoning task. On the other hand, the finding that a large percentage of curated explanations still represent incomplete arguments has a deeper implication on the nature of explanations and on what annotators perceive as a valid and complete logical argument. Therefore, we argue that future progress on the design of XGSs will depend, among other things, on a better formalisation and understanding of the inferential properties of explanations.

References

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

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.

Samuel R. Bowman and George E. Dahl. 2021. What will it take to fix benchmarking in natural language understanding?

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.

Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. 2020. Make up your mind! adversarial generation of inconsistent natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4157–4165, Online. Association for Computational Linguistics.

Ruben Cartuyvels, Graham Spinks, and Marie-Francine Moens. 2020. Autoregressive reasoning over chains of facts with transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6916–6930, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Deborah Ferreira and André Freitas. 2020a. Natural language premise selection: Finding supporting statements for mathematical text. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2175–2182, Marseille, France. European Language Resources Association.

Deborah Ferreira and André Freitas. 2020b. Premise selection in natural language mathematical texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7365–7374, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 137–150, Online. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Peter Lipton. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247–266.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. 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.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself!

leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Viktor Schlegel, Marco Valentino, Andre Freitas, Goran Nenadic, and Riza Batista-Navarro. 2020. A framework for evaluation of machine reading comprehension gold standards. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5359–5369, Marseille, France. European Language Resources Association.

Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5594–5608, Online. Association for Computational Linguistics.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020a. Explanationlp: Abductive reasoning for explainable science question answering.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020b. A survey on explainability in machine reading comprehension.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021. Unification-based reconstruction of multi-hop explanations for science questions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 200–211, Online. Association for Computational Linguistics.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2020. Explainable natural language reasoning via conceptual unification.

Douglas Walton. 2004. A new dialectical theory of explanation. *Philosophical Explorations*, 7(1):71–89.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 307–321, Barcelona (online). International Committee for Computational Linguistics.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable nlp.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5456–5473, Marseille, France. European Language Resources Association.

Dongfang Xu, Peter Jansen, Jaycie Martin, Zhengnan Xie, Vikas Yadav, Harish Tayyar Madabushi, Oyvind Tafjord, and Peter Clark. 2020. Multi-class hierarchical question classification for multiple choice science exams. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5370–5382, Marseille, France. European Language Resources Association.

Looking for a Role for Word Embeddings in Eye-Tracking Features Prediction: Does Semantic Similarity Help?

Lavinia Salicchi
The Hong Kong Polytechnic University
lavinia.salicchi@connect.polyu.hk

Alessandro Lenci
University of Pisa
alessandro.lenci@unipi.it

Emmanuele Chersoni
The Hong Kong Polytechnic University
emmanuelechersoni@gmail.com

Abstract

Eye-tracking psycholinguistic studies have suggested that context-word semantic coherence and predictability influence language processing during the reading activity.

In this study, we investigated the correlation between the cosine similarities computed with word embedding models (both static and contextualized) and eye-tracking data from two naturalistic reading corpora. We also studied the correlations of surprisal scores computed with three state-of-the-art language models.

Our results show strong correlation for the scores computed with BERT and GloVe, suggesting that similarity can play an important role in modeling reading times.

1 Introduction

Eye-tracking data recorded during reading provide invaluable evidence about the factors influencing language comprehension. Research in computational modeling has particularly focused on two factors: i.) the semantic coherence of a word with the rest of the sentence (Ehrlich and Rayner, 1981; Pynte et al., 2008; Mitchell et al., 2010), measured via *semantic similarity* metrics and ii.) its predictability from previous context, as measured by *surprisal* (Hale, 2001; Levy, 2008). Intuitively, words that have low semantic coherence and low in-context predictability (i.e., high surprisal) induce longer reading times.

In distributional semantics (Lenci, 2018), words and their sentence contexts are represented with dense vectors called *embeddings* and produced by Distributional Semantic Models (DSM). In this paper, we modeled semantic coherence with the cosine similarity between the embeddings of words and their sentence contexts, and then we tested the correlation of the metric with the eye-tracking measures annotated on the GECO and Provo corpora. We analyzed the correlations for the similarity computed with 10 different embedding models (both static and contextualized), as well as for surprisal scores computed with several state-of-the-art neural language models. Among all the features under investigation, the similarity scores obtained with BERT and GloVe obtained the best correlations across features in both the benchmark corpora.

2 Related Work

Hollenstein et al. (2019) proposed a framework to evaluate six state-of-the-art word embedding models (GloVe, Word2Vec, WordNet2Vec, FastText, ELMo, BERT). The evaluation was based on the model capability to reflect semantic representations in the human mind, using cognitive data in different datasets for eye-tracking, EEG, and fMRI. Word embedding models were used to train neural networks on a regression task. While we aim at creating a computational model of the relationship between context processing and the integration of a new word during naturalistic reading, Hollenstein et al. (2019) evaluated embedding models on the prediction of out-of-context word features. The results of their analyses showed that BERT, ELMo, and FastText have the best prediction performances. On the other hand, approaches based on powerful Transformers language models were outperformed by a classifier using linguistic and psychometric features (Bestgen, 2021) in the recent CMCL 2021 Shared Task on Eye-Tracking Data Prediction (Hollenstein et al., 2021).

A series of contributions explored the role of surprisal in modeling reading times in naturalistic settings, coming to the general conclusion that the predictive power is strongly related to the language model quality, i.e. models with better perplexity perform better (Smith and Levy, 2013; Goodkind and Bicknell, 2018). Later work explored the most recent neural models, including LSTM (van Schijndel and Linzen, 2018), GRU (Aurnhammer and Frank, 2019), Transformers (Merkx and Frank, 2020) and GPT-2 (Wilcox et al., 2020), basically

Proceedings of the 14th International Conference on Computational Semantics, pages 87–92
June 17–18, 2021.

confirming this relationship.[1]

Early studies had also found correlations between semantic distance, computed by word embeddings, and eye-tracking features in reading processes (Pynte et al., 2008; Mitchell et al., 2010). However, the more recent work by Frank (2017) pointed out that, since word embeddings are based on co-occurrences, semantic distance may actually represent word predictability, rather than semantic relatedness, and that those early findings were actually due to a confound between these two concepts. To test this hypothesis, the author used linear regression models with and without surprisal, testing 5 surprisal measures. The results show that the effects of similarity on reading times disappear when surprisal is factored out, thereby proving the existence of a complex interplay between the two factors. Frank's experiments were carried out in a naturalistic reading setting and, to our knowledge, there have been no eye-tracking studies with controlled stimuli investigating a possible separate effect of the two components (for example, by comparing the fixation patterns of words that have low predictability, but different degrees of coherence with the sentence or with the discourse context).

3 Experimental Setting

3.1 Datasets

Traditional corpora annotated with eye-tracking data consist of short isolated sentences (or even single words) with particular structures or lexemes, in order to investigate specific syntactic and semantic phenomena. In the present work, we used GECO (Cop et al., 2017) and Provo (Luke and Christianson, 2018), two eye-tracking corpora containing long, complete, and coherent texts. **GECO** is a monolingual and bilingual (English and Dutch) corpus composed of the entire Agatha Christie's novel *The Mysterious Affair at Styles*. The corpus is freely downloadable with a related dataset containing eye-tracking data of 33 subjects (19 of them bilingual, 14 English monolingual) reading the full novel text, presented paragraph-by-paragraph on a screen. GECO is composed of $54,364$ tokens. **Provo** contains 55 short English texts about various topics, with 2.5 sentences and 50 words on average, for a total of $2,689$ tokens, and a vocabulary of $1,197$ words. These texts were read by 85 subjects and their eye-tracking measures were collected in an available on-line dataset. GECO and Provo data are particularly interesting because they are recorded during naturalistic reading, instead of short selected stimuli.

For every word in the corpora, we extracted its mean *total reading time*, mean *first fixation duration*, and mean *number of fixations*, by averaging over the subjects. The choice of modeling mean eye-tracking measures is justified by the high inter-subject consistency of the recorded data. For instance, Cop et al. (2017) report an overall inter-subject correlation of 0.9 for the total reading times in GECO.

3.2 Word Embeddings

Table 1 shows the embeddings types used in our experiments, consisting of 6 non-contextualized, static DSMs and 4 contextualized DSMs. The former include predict models (**SGNS** and **FastText**) (Mikolov et al., 2013; Levy and Goldberg, 2014; Bojanowski et al., 2017) and count models (**SVD** and **GloVe**) (Bullinaria and Levy, 2012; Pennington et al., 2014). [2] Four DSMs are window-based and two are syntax-based (**synt**). Embeddings have 300 dimensions and were trained on a corpus of 3.9 billion tokens ca. (a concatenation of ukWaC and a 2018 dump of Wikipedia). Pre-trained contextualized embeddings include the 512-dimensional vectors produced by the three layers of the **ELMo** bidirectional LSTM architecture (Peters et al., 2018), the $1,024$-dimensional vectors produced by the 24-layers **BERT-Large** Transformer architecture (BERT-Large, Cased) (Devlin et al., 2019), the $1,600$-dimensional vectors by **GPT2-xl** (Radford et al.), and finally, the 200-dimensional vectors produced by the **Neural Complexity** model by van Schijndel and Linzen (2018).

3.3 Method

Our main goals were to investigate the potential contribution of cosine similarity in predicting eye-tracking features, to compare different word embedding models, and then to evaluate whether the information represented by cosine similarity is similar to the one represented by surprisal.

For each target word w in GECO and Provo, we measured the **cosine similarity** between the embedding of w and the embedding of the context

[1] Notice however that doubts have been raised on the reliability of perplexity as a metric for comparing large pretrained models, since it does not allow to compare models with different vocabularies (Hao et al., 2020).

[2] For the distinction between count and predict DSM, we refer to Baroni et al. (2014).

Model	Hyperparameters
Non-contextualized DSMs	
SVD.w2	count DSM with 345K window-selected context words, window of width 2, reduced with SVD
SVD.synt	count DSM with 345K syntactically typed context words reduced with SVD
GloVe	count DSM with context window of width 2, reduced with log-bilinear regression
SGNS.w2	Skip-gram with negative sampling, context window of width 2, 15 negative examples
SGNS.synt	Skip-gram with negative sampling, syntactically-typed context words, 15 negative examples
FastText	Skip-gram with subword information, context window of width 2, 15 negative examples
Contextualized DSMs	
ELMo	Pretrained ELMo embeddings on the 1 Billion Word Benchmark
BERT	Pretrained BERT-Large embeddings on the concatenation of the Books corpus and Wikipedia
GPT2-xl	Pretrained GPT2-xl embeddings on WebText
Neural Complexity	Pretrained Neural Complexity embeddings on Wikipedia

Table 1: List of the embedding models used for the study, together with their hyperparameter settings.

c formed by the previous words in the same sentence. We then computed the Spearman correlation between the cosine and the eye-tracking data for w (total reading time, first fixation duration, and number of fixations). To create context embedding, we used an **additive model**: the context vector is the sum of all its word embeddings.

Given the bidirectional nature of BERT, the input to this model needed a special pre-processing: To prevent that the vectors representing words within the context were computed using the target word itself, we passed to BERT a list of sub-sentences, each of which were composed of context words only. So given the sentence *The dog chases the cat*:
S[0] = ["The"]
S[1] = ["The dog"]
S[2] = ["The dog chases"]
S[3] = ["The dog chases the"]
S[4] = ["The dog chases the cat"]
Starting from the second sub-sentence, the cosine similarity was computed between the last word vector and the sum of words vectors belonging to the previous sub-sentence (list element). So, to compute the cosine similarity between *cat* and the previous context, we selected *cat* from S[4] and $The + dog + chases + the$ from S[3].

For BERT we used as context also the embedding produced by the model for the special token **CLS**, which is created using a weighted additive model. As for the $simple$ additive model, BERT was fed with sub-sentences, and for each target word the CLS-context-vector was the one computed at the previous list element. In the previous example, given *cat* as target word, we used the CLS vector representing all the S[3] elements.

Given the positive effect of semantic coherence on language processing, we expected that the eye-tracking data for w had a *negative correlation* with its cosine similarity with c: **The higher the cosine, the lower the reading time of w measured by eye-tracking**.

We used BERT, GPT2-xl and Neural Complexity to compute word-by-word surprisal. Like with cosine similarity, the input sentences for BERT were organized in sub-sentences, and the last token (i.e., the target word), was replaced with the special tag [MASK]. Finally, we computed the Spearman correlation between the **surprisal** of w, and the eye-tracking data for the target word. Differently from the cosine, we expected the surprisal to be *positively correlated* with the word reading time: **The less predictable a word is, the slower its processing will be**.

The analyses have been performed with the following models: 6 values of cosine similarity between non-contextualized vectors, 51 values of cosine similarity between contextualized vectors (48 from 24 layers of BERT in two different ways to compute the context vector, and 3 from ELMo, GPT2-xl and Neural Complexity), 3 values of surprisal from BERT, GPT2-xl, Neural Complexity.

4 Results and Discussion

Looking at the correlations results, it is clear that every model performed better on Provo. One possible explanation for this difference is that GECO eye-tracking data are recorded on participants reading a literary text, while Provo materials are online news articles, science magazines and only partially short text from works of fiction. The consequence is a difference in the syntactic complexity of sentence structure and in the frequency of words. This gap implies that the modeling of GECO contexts is less directly reducible to an additive fashion of processing, and, most importantly, is more likely to find *Out Of Vocabulary* words in GECO, rather than in Provo.

Corpus	Model	total reading time	1st fix. duration	number fixations
	BERT Additive (22)	-0.54	-0.53	-0.55
	BERT CLS (22)	**-0.57**	**-0.56**	**-0.58**
	ELMo (1)	-0.35	-0.34	-0.36
	FastText	-0.39	-0.38	-0.40
	GloVe	-0.45	-0.44	-0.46
GECO	SGNS.w2	-0.40	-0.39	-0.40
	SGNS.synt	-0.30	-0.29	-0.30
	SVD.w2	*-0.07*	*-0.06*	*-0.07*
	SVD.synt	-0.24	-0.23	-0.24
	GPT2-xl	-0.05	-0.05	-0.05
	NC	-0.12	-0.11	-0.12
	BERT Additive (22)	-0.65	-0.66	-0.66
	BERT CLS (22)	**-0.71**	**-0.72**	**-0.71**
	ELMo (1)	-0.36	-0.36	-0.37
	FastText	-0.57	-0.56	-0.57
	GloVe	-0.65	-0.65	-0.66
Provo	SGNS.w2	-0.60	-0.60	-0.60
	SGNS.synt	-0.42	-0.42	-0.43
	SVD.w2	*-0.03*	*-0.02*	*-0.03*
	SVD.synt	-0.32	-0.32	-0.32
	GPT2-xl	-0.37	-0.38	-0.38
	NC	-0.16	-0.17	-0.17

Table 2: Spearman correlations between the target-context cosine and the eye-tracking measures. Numbers in parenthesis indicate models' layers.

Corpus	Model	total reading time	1st fixation duration	number fixations
	BERT	0.28	0.26	0.28
GECO	GPT2-xl	0.41	0.39	0.41
	NC	0.31	0.30	0.32
	BERT	0.25	0.24	0.24
Provo	GPT2-xl	0.44	0.43	0.44
	NC	0.46	0.48	0.46

Table 3: Spearman correlations between surprisal and eye-tracking measures.

Another aspect that is quite evident are the similar correlation values among different eye-tracking features. This aspect is not surprising: in the original datasets of GECO and Provo, it can be noticed that many words show the same value for the total reading time and the first fixation duration. This happens when i) the word is not read (0 ms for both the features); ii) the word is read only once (total reading time and first fixation duration overlap). Also regarding the similar values of the correlations between similarity and number of fixations and between similarity and total reading times, taking into account the original data gives us an explanation of the results: since the total reading time is computed summing the duration of all the multiple fixations, the higher the number of fixation, the higher the total reading time, leading to a similar tendency in the values of the two features. For these reasons, the total reading time may be considered as a "bridge" field, that holds close relations with both first fixation duration and number of fixations, justifying the similar correlation values in our results.

Comparing word embedding models, we may notice that correlations can reach very high values, up to -0.71 for the total reading time (by BERT CLS layer 22), suggesting that semantic coherence -modeled as cosine similarity between context and target- can be a strong predictor of eye-tracking measures of reading process. GloVe (mean correlation over eye-tracking features on GECO: -0.45, on Provo: -0.65) and BERT (mean correlation over eye-tracking features on GECO: -0.57, on Provo: -0.71) score the best results on both corpora, and in the latter case the [CLS] context model brings some advantage over the simple additive one. The lower BERT layers show a steadily decreasing performance (see Figure 1). This was expected because, as it was pointed out in the layers analysis by Tenney et al. (2019), the BERT architecture reproduces the classical NLP pipeline: the lower layers process mainly the syntactic information, while the highest ones give a more precise representation of semantic relations. We also notice a strong variability among the embedding models, which is orthogonal to the contextualized vs. non-contextualized dichotomy. The ELMo contex-

tualized vectors perform much worse than BERT ones, probably because they have a lower degree of contextualization, and syntax-based count models are not significantly worse than predict DSMs.

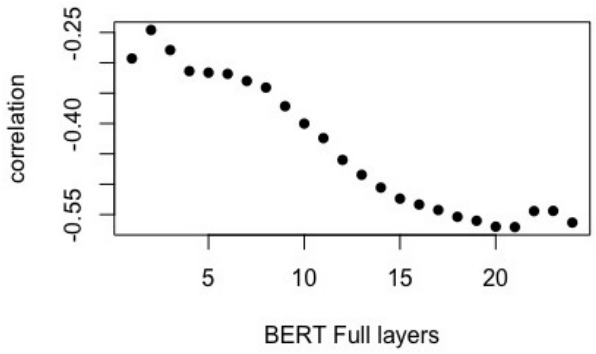


Figure 1: Spearman's correlation of different layers of BERT on GECO.

Regarding the correlations between the target word surprisal computed with BERT, GPT2-xl and Neural Complexity (NC) and the eye-tracking measures (see Table 3), the first striking fact is that the absolute values are generally lower than the scores obtained with the cosine (higher correlations are reached by GPT2-xl con GECO, mean correlation = 0.40, and by NC on Provo, mean correlation = 0.47). This might prompt us to conclude that surprisal is a much weaker predictor than semantic coherence. However, a significant negative correlation between cosine similarity and surprisal (e.g. with BERT it is -0.40 on GECO and -0.32 on Provo) supports the hypothesis by Frank (2017) that there is a strong overlap between semantic coherence and surprisal. Factoring out the contribution of these two factors on eye-tracking features will be the next step of our research work.

5 Conclusions and ongoing work

In this paper, we have used contextualized and non-contextualized DSMs to compute the cosine between a target word and the previous sentence context. Our results show that cosine similarity is able to achieve very high correlations with the eye-tracking metrics of GECO and Provo, especially with the BERT and GloVe models, providing further evidence that semantic coherence is potentially very useful in modeling reading times. Furthermore, we computed word-by-word surprisal using BERT, GPT2-xl, and Neural Complexity.

Among the language models, the best results have been achieved by GPT2-xl, confirming the previous findings that Transformers are very good at modeling sentence processing metrics (Wilcox et al., 2020; Hao et al., 2020; Merkx and Frank, 2021). However, the absolute value of correlation is lower than the one obtained with cosine similarity scores: for example, the mean correlation achieved on Provo with the cosine similarity between vectors produced by BERT is -0.71, while the correlation between eye tracking features and the surprisal computed by the same model is 0.24. The comparison between correlations reached by cosine similarity and surprisal may lead us to the conclusion that semantic coherence is a stronger predictor of eye-tracking features than word predictability. However, given the significant degree of correlation between cosine similarity and surprisal, further investigations are needed to disentangle the two factors.

Our next step will be to include Transformers-based surprisal and vector-based cosine similarity in a large-scale regression study to predict eye tracking features, in order to ensure a close comparison with the experimental setting of Frank (2017), and to investigate if semantic similarity models can actually play a distinct role from surprisal in the prediction of reading times. Differently from Frank (2017), we plan to test with several regression models, from a simple linear regression to more advanced regression models (e.g. Gradient Boosting, Multilayer Perceptron etc.), and with different word embedding models, in order to account for the different types of semantic similarity computed by static and contextualized embeddings.

References

Christoph Aurnhammer and Stefan L Frank. 2019. Evaluating Information-theoretic Measures of Word Prediction in Naturalistic Sentence Reading. *Neuropsychologia*, 134.

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 ACL*.

Yves Bestgen. 2021. LAST at CMCL 2021 Shared Task: Predicting Gaze Data During Reading with a Gradient Boosting Decision Tree Approach. In *Proceedings of the NAACL Workshop on Cognitive Modeling and Computational Linguistics*.

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.

John A Bullinaria and Joseph P Levy. 2012. Extracting Semantic Representations from Word Co-Occurrence Statistics: Stop-Lists, Stemming, and SVD. *Behavior Research Methods*, 44(3):890–907.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting GECO: An Eye-Tracking Corpus of Monolingual and Bilingual Sentence Reading. *Behavior Reseach Methods*, 49(2):602–615.

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*.

Susan E. Ehrlich and Keith Rayner. 1981. Contextual Effects on Word Perception and Eye Movements During Reading. *Journal of Verbal Learning and Verbal Behavior*, 20:641–65.

Stefan L Frank. 2017. Word Embedding Distance Does not Predict Word Reading Time. In *Proceedings of CogSci*, pages 385–390.

Adam Goodkind and Klinton Bicknell. 2018. Predictive Power of Word Surprisal for Reading Times is a Linear Function of Language Model Quality. In *Proceedings of the LSA Workshop on Cognitive Modeling and Computational Linguistics*.

John Hale. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. In *Proceedings of NAACL*.

Yiding Hao, Simon Mendelsohn, Rachel Sterneck, Randi Martinez, and Robert Frank. 2020. Probabilistic Predictions of People Perusing: Evaluating Metrics of Language Model Performance for Psycholinguistic Modeling. In *Proceedings of the EMNLP Workshop on Cognitive Modeling and Computational Linguistics*.

Nora Hollenstein, Emmanuele Chersoni, Cassandra L Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2021. CMCL 2021 Shared Task on Eye-Tracking Prediction. In *Proceedings of the NAACL Workshop on Cognitive Modeling and Computational Linguistics*.

Nora Hollenstein, Antonio de la Torre, Nicolas Langer, and Ce Zhang. 2019. CogniVal: A Framework for Cognitive Word Embedding Evaluation. In *Proceedings of CONLL*.

Alessandro Lenci. 2018. Distributional Models of Word Meaning. *Annual Review of Linguistics*, 4:151–171.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of ACL*.

Roger Levy. 2008. Expectation-based Syntactic Comprehension. *Cognition*, 106(3):1126–1177.

Steven G Luke and Kiel Christianson. 2018. The Provo Corpus: A Large Eye-tracking Corpus with Predictability Norms. *Behavior Research Methods*, 50(2):826–833.

Danny Merkx and Stefan L Frank. 2020. Comparing Transformers and RNNs on Predicting Human Sentence Processing Data. *arXiv preprint arXiv:2005.09471*.

Danny Merkx and Stefan L Frank. 2021. Human Sentence Processing: Recurrence or Attention? In *Proceedings of the NAACL Workshop on Cognitive Modeling and Computational Linguistics*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Jeff Mitchell, Mirella Lapata, Vera Demberg, and Frank Keller. 2010. Syntactic and Semantic Factors in Processing Difficulty: An Integrated Measure. In *Proceedings of ACL*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL*.

Joel Pynte, Boris New, and Alan Kennedy. 2008. Online Contextual Influences During Reading Normal Text: A Multiple-Regression Analysis. *Vision research*, 48(21):2172–2183.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. In *Open-AI Blog*.

Marten van Schijndel and Tal Linzen. 2018. A Neural Model of Adaptation in Reading. In *Proceedings of EMNLP*.

Nathaniel J Smith and Roger Levy. 2013. The Effect of Word Predictability on Reading Time Is Logarithmic. *Cognition*, 128(3):302–319.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. *arXiv preprint arXiv:1905.05950*.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger Levy. 2020. On the Predictive Power of Neural Language Models for Human Real-Time Comprehension Behavior.

Automatic Assignment of Semantic Frames in Disaster Response Team Communication Dialogues

Natalia Skachkova and Ivana Kruijff-Korbayová
DFKI, Saarland Informatics Campus, 66123 Saarbrücken Germany
natalia.skachkova@dfki.de;ivana.kruijff@dfki.de

Abstract

We investigate frame semantics as a meaning representation framework for team communication in a disaster response scenario. We focus on the automatic frame assignment and retrain PAFIBERT, which is one of the state-of-the-art frame classifiers, on English and German disaster response team communication data, obtaining accuracy around 90%. We examine the performance of both models and discuss their adjustments, such as sampling of additional training instances from an unrelated domain and adding extra lexical and discourse features to input token representations. We show that sampling has some positive effect on the German frame classifier, discuss an unexpected impact of extra features on the models' behaviour and perform a careful error analysis.

1 Introduction

In this paper we employ the theory of frame semantics as a meaning representation framework for dialogues from the domain of disaster response. Our work is part of a larger research project developing methods to capture and interpret verbal team communication in disaster response scenarios and use the extracted run-time mission knowledge for mission process assistance, as described in (Willms et al., 2019). Team communication interpretation encompasses several aspects, some of which have been addressed in earlier publications of our team: (Anikina and Kruijff-Korbayova, 2019) present dialogue act classification results; (Skachkova and Kruijff-Korbayova, 2020) provide an analysis of contextual reference phenomena. The present paper complements this by results on semantic frame assignment. To our knowledge, our work is the first to use semantic frames in the domain of disaster response, and one of the few attempts implementing a frame classifier for dialogue.

Frame semantics is a paradigm defining the meaning of words through the context they are used in (Fillmore, 1976). This assumes that, depending on context, a word (or an expression) is able to evoke in our minds a certain event or situation together with a set of slots called frame elements associated with it, even if some of these slots were not explicitly filled in the sentence.

Using frame semantics as a meaning representation requires frame semantic parsing, namely identifying frame-evoking elements (targets) and the corresponding frames, as well as recognizing certain spans as frame elements and classifying them. In this paper we address the task of automatic semantic frame assignment, given a target. The novelty is that we work on English and German dialogues in robot-assisted disaster response teams.

We use the TRADR corpus (Kruijff-Korbayová et al., 2015), which contains transcribed communication in teams of firefighters using robots for incident site reconnaissance during a series of exercises that simulated situations after a disaster, such as a fire, explosion, etc. Towards the aim of creating structured representations of the events and activities during a first response mission by means of semantic frames, we experiment with some existing models. We start with a simple sequence classification approach that assumes fine-tuning of a pretrained BERT model (Devlin et al., 2019) on the TRADR corpus. Next, we use one of the existing state-of-the-art frame classifiers called PAFIBERT (Tan and Na, 2019). We re-implement and train it on the English FrameNet (Baker et al., 1998) data, and evaluate the model on English TRADR dialogues. We also experiment with re-training PAFIBERT on the TRADR data, despite the small corpus size. In addition, we investigate a possibility of training a frame classifier on mixed data - FrameNet or SALSA (Burchardt et al., 2006) plus TRADR - and consider three sampling approaches. Finally, we examine whether enriching the input

Proceedings of the 14th International Conference on Computational Semantics, pages 93–109
June 17–18, 2021.

with lexical and discourse features has an effect on the classifier performance. In contrast to many papers that report standard accuracy or F-score to measure the performance of a frame classifier, we use the index of balanced accuracy metric (García et al., 2009) designed specifically for imbalanced data.

In Section 2 we give a brief overview of the theory of frame semantics. In Section 3 we introduce the most notable frameworks designed to perform automatic frame assignment or frame-semantic parsing. In Section 4 we examine the distribution of semantic frames and the role of ambiguous targets in the TRADR corpus, compare our data with FrameNet and SALSA, and explain how we prepared and split all the data into training, validation and test sets. Section 5 describes the experiments and their results. In Section 6 we make a conclusion and indicate possible further steps.

2 Frame Semantics

According to Petruck (2019), frame semantics is a research program in empirical semantics which emphasizes the continuities between language and experience, and provides a framework for presenting the results of that research.

The theory of frame semantics goes back to the 1970s. One of the pioneers in this area was Charles J. Fillmore. He suggested that a language description should include not only lexicon and grammar, but also a set of 'frames' that incorporate the semantics of the language elements (Fillmore, 1976). Fillmore (1982) uses the word 'frame' as a general cover term for such concepts as 'schema', 'script', 'scenario', or 'cognitive model'. He defines a frame as a system of concepts which are related to each other, and states that one cannot understand a concept without understanding the whole structure it is a part of. Frame semantics tries to describe and formalize such structures.

The FrameNet project (Baker et al., 1998) is considered one of the first practical realizations of the theory of frame semantics for English. One of its achievements was the creation of a lexical database that covers more than 13,000 word senses, is both human- and machine-readable and available online. Besides, more than 200,000 sentences were annotated with about 1,200 semantic frames, and are now known as the FrameNet corpus.

Examples 2.1 and 2.2 present a definition of the *Inspecting* frame and its frame elements (FEs), annotated with respect to the target *inspected*. Note that FEs can be 'core' (i.e. essential to the meaning of a frame) and 'non-core' (i.e. not uniquely characterizing). Usually, core FEs are part of the frame definition, like INSPECTOR and GROUND in Example 2.2.

Example 2.1 *'Inspecting' Frame Definition*
An INSPECTOR *directs his/her perceptual attention to a* GROUND *to ascertain whether the* GROUND *is intact or whether an* UNWANTED_ENTITY *is present. Alternatively, the desired outcome of the inspection may be presented as a* PURPOSE.

Example 2.2 *'Inspecting' Frame's FEs*
[$_{INSPECTOR}$ *He] moved toward the control panel and [*$_{TARGET}$ ***inspected****] [*$_{GROUND}$ *it] [*$_{LOCATION\_OF\_PROTAGONIST}$ *from a distance], [*$_{MEANS}$ *without touching it].*

Databases similar to FrameNet were also created for other languages. In Section 4 we compare the FrameNet corpus and its German counterpart SALSA with the TRADR data.

3 Related Work

Frame semantics is not one of the most common meaning representation frameworks. However, research in the area of frame-semantic parsing has increased since frame-semantic structure extraction was included as a task in SemEval'07 (Baker et al., 2007). Most of the existing works present models trained on text data. Some of the projects deal only with automatic frame assignment, others have a bigger goal, namely, recognizing targets, frames and frame elements. In what follows we will focus on automatic frame assignment.

Most of the early frameworks are based on the idea of learning the frame labels from frame-evoking targets represented as rather elaborated sets of features, which include the target's lemma, its part of speech, etc. Many features rely on dependency syntax. For non-ambiguous targets a frame can be retrieved using a simple mapping. If the target is ambiguous, the correct label is learned using a Naive Bayes classifier, e.g., as shown by Erk (2005), or an SVM classifier like in the framework called LTH (Johansson and Nugues, 2007), or a discriminative probabilistic (log-linear) model like in SEMAFOR by Das et al. (2010).

The success of neural networks for many NLP tasks resulted in a gradual switch from the feature-based approaches to embeddings and a broader

usage of neural networks for the task of automatic frame assignment. One of the first semantic parsers to use embeddings was developed by Hermann et al. (2014). They represent targets as vectors, certain parts of which are reserved for certain argument representations. All frame labels are also vectors, and the classifier learns to minimize the distance between the targets and the correct labels. Other frameworks based on embeddings and various types of neural networks include SimpleFrameId (Hartmann et al., 2017) - a two-layer network which also allows to perform frame filtering using mappings of certain lexical units to certain frames from the FrameNet database; a framework by Yang and Mitchell (2017) that performs frame identification using a simple multi-layer network; TSABCNN (Zhao et al., 2018), which uses *word2vec* embeddings and convolutional neural networks.

Recently, there appeared frameworks that rely on BERT embeddings and pretrained models. E.g., PAFIBERT (Tan and Na, 2019) fine-tunes the pretrained BERT model using an attention mechanism to give weights to words that make up the context of the target. An interesting alternative approach was presented by Kalyanpur et al. (2020). They interpret frame-semantic parsing as a sequence-to-sequence generation problem. Their approach is based on the encoder-decoder architecture, namely on the T5 model, which is available via the *HuggingFace* library (Wolf et al., 2020).

Ribeiro et al. (2020) treat automatic frame assignment as a clustering problem. They focus on verbal frame-evoking targets and represent them using contextualized ELMo embeddings. The targets are treated as nodes in a graph, and clustered using the *Chinese Whispers* algorithm (Biemann, 2006). A new instance is classified by determining the closest cluster.

All the above frameworks were trained on text data. We found only two frame-semantic parsers designed specifically for dialogue. One of them was created in the course of the LUNA project (Raymond et al., 2008) and focuses mostly on frame element classification (Coppola et al., 2008). The other was presented by Trione et al. (2015). Its main goal is actually to speed up the manual annotation process, not pure frame-semantic parsing. Frames are detected with the help of a hand-crafted set of lexical triggers, which includes 200 most frequent words from 7 domains.

A comparison of the frameworks mentioned above, as well as the results of their evaluation on the test data can be found in Appendix D. We do not place them here for space reasons.

For the experiments on the TRADR data presented in this paper we have chosen the PAFIBERT approach (Tan and Na, 2019). PAFIBERT is one of the state-of-the-art frame classifiers, it showed about 89% accuracy when evaluated on the FrameNet test set, and it is easy to re-implement.

4 Data for experiments

The TRADR corpus consists of 15 files with dialogues, six files contain dialogues in English, and nine - in German. Six German dialogues were translated into English in order to get more English training data. TRADR dialogues comprise the communication in first responder teams using robots for disaster site reconnaissance. Each team consists of several operators (OP) who control ground and airborne robots, a team leader (TL) and sometimes also a mission commander (MC).

Table 1 shows the distribution of dialogue turns, utterances and tokens between the mission participants in both English and German TRADR dialogues. Also, average numbers of utterances per turn and tokens per utterance are given. We see that both English and German parts of the data contain approximately the same number of dialogue turns, however the turns in the English dialogues are slightly longer, and as a result the English part of the corpus is 1.5 times larger. The utterances are usually rather short - 7-9 tokens on average, as the team participants try to be brief and precise.

	MC	TL	OP	Total
German data				
# Dialogue turns	60	984	1,020	2,064
# Utterances	61	997	1,027	2,085
# Tokens	526	6,165	7,875	14,566
Avg. # utt. per DT	1.02	1.01	1.01	1.01
Avg. # tokens per utt.	8.62	6.18	7.67	6.99
English data (including translations)				
# Dialogue turns	60	1,013	1,021	2,094
# Utterances	61	1,306	1,186	2,553
# Tokens	820	9,983	11,353	22,156
Avg. # utt. per DT	1.02	1.29	1.16	1.23
Avg. # tokens per utt.	13.44	7.64	9.57	8.68

Table 1: TRADR corpus overview

We annotated the utterances in the English TRADR dialogues with frame-evoking targets, corresponding lexical units (LUs), frames and parent frames. Frame elements were not annotated. The German TRADR data was annotated similarly, except that we replaced targets and LUs with 'tar-

get related elements', which represent the whole phrase that the target is a part of. We assumed that each utterance can potentially have several targets or groups of frame related elements. As a result, the number of frame instances in the TRADR corpus is larger than the number of utterances given in Table 1. While annotating our data with semantic frames we tried to follow the FrameNet annotation guidelines (Ruppenhofer et al., 2006). Due to the specifics of our domain, many FrameNet frame definitions had to be adapted. Also, ten new frames were introduced. The English and German parts of the corpus were annotated by two different annotators. To check the reliability of the annotation, one dialogue in German (534 frame instances) was also annotated by the person responsible for the annotation of the English dialogues. Inter-annotator agreement measured using Cohen's Kappa (Carletta, 1996) reached 0.73, which is considered reliable. A team communication example annotated with semantic frames, as well as the definitions of the new frames are available in Appendix C. We are making the annotated data available online.[1]

In total, the English and German parts of the TRADR corpus contain 4,191 and 3,519 frame instances, respectively. These instances are distributed between 190 (English) and 152 (German) different frame labels. The distribution of the frame labels is not uniform.Thus, in English TRADR almost 60% of all the instances belong to the top ten most frequent frames, and 137 out of 190 frames have only ten or less samples, which all together make up about 10% of the data. In German TRADR the instances of the top ten most frequent frames make up approximately 58%, and instances of 105 infrequent frames - almost 11% of the data. The fact that the TRADR data is highly imbalanced motivates the choice of performance metrics for the evaluation of the frame classifiers that will be discussed in the next section.

The English TRADR data counts 434 different LUs. Their distribution is also not uniform: the top ten most common LUs occur in about 40% of all the utterances and at the same time make only slightly more than 2% of the total of different LUs. All LUs are distributed between seven different POS tags. 75% of the utterances contain verbal targets. The second frequent POS tag is an interjection - almost 8% of all the targets.

[1]The TRADR data and the semantic frame annotations can be obtained at http://talkingrobots.dfki.de/.

Only about 15% of all LUs in English TRADR are ambiguous. However, they are realized in nearly 53% of utterances containing targets. Simple calculations show that on average a single LU evokes 1.24 frames. So, while the ambiguous LUs are not very frequent in comparison to non-ambiguous ones, the frames that they evoke are frequent, and this may become a problem for the frame classifier, as it is not always possible to perform frame disambiguation using the utterance context.

Besides TRADR we also use the FrameNet and SALSA datasets for our experiments, so it is necessary to compare them with our data. The differences between the corpora are summarized in Table 14, presented in Appendix. Note that for the experiments all duplicate sentences/utterances (i.e. equal strings with equal labels), as well as elliptical utterances and communication fragments (in TRADR) were removed. The numbers in Table 14 are based on the cleaned versions of the corpora. The only exception is the average utterance length in the TRADR corpus, that was calculated based on the original data in Table 1.

The FrameNet and SALSA data are very different from TRADR, cf. Table 14. First, they are much larger and come from other domains (note that the domains of FrameNet and SALSA are quite close to each other). Both FrameNet and SALSA include many more frames than TRADR, and despite the fact that many frames are common for all the corpora (e.g., about 93% of frame labels in English TRADR also occur in FrameNet), the frame distributions are very different. The fact that less than 65% of TRADR LUs are common with FrameNet LUs, which are much more numerous, supports this. Both FrameNet and SALSA are also imbalanced and FrameNet contains ambiguous targets.

Data	TRADR				Frame-Net	# cls
	Eng	# cls	Ger	# cls		
Training	1,955	81	1,902	72	143,509	931
Validation	489	81	476	72	35,877	931
Test	268	81	259	72	19,923	931
Test (subs.)	234	50	-	-	-	-

Table 2: Training, validation & test data sizes

All the datasets were shuffled and randomly split into training, validation and test data as shown in Table 2. Note that the number of classes (frame labels) is smaller than given in Table 14, as all the frames that have less than five instances were removed. This was necessary to perform 5-fold cross-

validation. Note that we have two English TRADR test sets. The second one is a subset of the first one, and contains the instances of 50 frames common to both FrameNet and TRADR. It is needed to test the PAFIBERT model trained on FrameNet.

5 Experiments and Discussion

In this section we will present semantic frame classifiers for both English and German TRADR dialogues. Our main focus is on the English data. We introduce several models, split into basic and adjusted, and discuss their performance.

As all our datasets have hundreds of classes and are highly imbalanced, many typical performance metrics, e.g., accuracy, precision, F-score, are not reliable (Tharwat, 2020). Instead, we use the index of balanced accuracy (IBA) metric as our main performance measure, calculated using the *Python imbalanced-learn* package (Lemaître et al., 2017). The package also outputs the scores of the common metrics, such as recall, precision and F-score, and we show them for the sake of comparison, as most papers on automatic frame assignment report either accuracy, or these metrics. All the metrics are calculated using macro-averaging.

5.1 Basic models

The first group includes four models. The first one is a naive baseline, represented by the *BertForSequenceClassification* model from the *Transformers* library (Wolf et al., 2020) fine-tuned on English TRADR. *BertForSequenceClassification* was chosen as the most straightforward way to perform sequence classification. It is a pretrained BERT model with an additional linear layer on top of the pooled output. The other three models reproduce the architecture of PAFIBERT. The implementation details can be found in the original paper by Tan and Na (2019). One of the models was trained on the FrameNet data, another - purely on English TRADR data, the last one - on German TRADR data.

All four models were trained with 5-fold cross-validation. As both English and German TRADR datasets are small, different splits into training and test parts may result in noticeable performance variance. We used cross-validation to get a more reliable estimation of the performance of the models, not for hyper-parameter search. All hyper-parameters were taken from the original paper. Following Tan and Na (2019), training was performed for 8 epochs per fold using an adaptive learning rate that starts with 3e-5 and an *AdamW* optimizer. In the course of cross-validation we always saved the model with the best IBA validation score. Next, the model was evaluated on the test data.

The performance of the basic models is summarized in Table 3. We see that *BertForSequenceClassification* demonstrates rather unsatisfactory performance - IBA only 32% - 37%. The reason for this is the fact that simple fine-tuning does not integrate information about the frame-evoking targets and their contexts, so that it is impossible for the model to guess what tokens in the sequence it has to focus on. It is obvious that in order to improve the performance, we need to tell the model which tokens in each utterance it should pay attention to, and PAFIBERT provides a convenient way to do so.

Classifier	Test set	PRE	REC	F1	IBA
BertForSequence-Classification (EN)	TR (EN)	0.33	0.39	0.35	0.37
	TR (subs.)	0.30	0.35	0.31	0.32
PAFIBERT trained on FrameNet (EN)	FN	0.92	0.92	0.92	0.91
	TR (subs.)	0.71	0.53	0.58	0.51
Basic model (EN)	TR (EN)	0.90	0.89	0.89	0.88
	TR (subs.)	0.91	0.88	0.88	0.86
Basic model (DE)	TR test (DE)	0.84	0.84	0.83	0.83

Table 3: Basic models: results; "TR" stands for TRADR test set, "FN" for FrameNet test.

As Table 3 shows, PAFIBERT trained on the FrameNet data has IBA of 91% when evaluated on the test set coming from the same distribution. This score is actually even slightly better than the standard accuracy of 89% reported by Tan and Na (2019). However, when tested on TRADR data, the model shows much worse results, namely, only 51% IBA, despite the fact that the majority of the 50 frames from the given test set have enough instances in the training set.

The main reasons why this classifier fails on the TRADR data are as follows. First of all, due to the fact that FrameNet is very fine-grained, many TRADR instances got classified as belonging to very specific frames which we did not use when annotating the TRADR data, like *'Interior_profile_relation'* and *'Non_gradable_proximity'* (we used their parent frame *'Locative_relation'* instead). Another reason is that TRADR instances of certain frames have targets that, due to domain differences, are not typical for these frames in FrameNet. For instance, all TRADR samples of *'Create_representation'* frame were misclassified, because the model expected *'draw'*, *'carve'* or *'sketch'* as targets, but got

'take/make a picture' and labeled the input utterances as *'Physical_artwork'* instead. Finally, there is also a problem of ambiguity. For example, the target *'change'* can evoke both *'Replacing'* and *'Cause_change'* frames, and the target *'lie'* - *'Posture'* and *'Being_located'*.

So, the error analysis shows that the PAFIBERT model trained on FrameNet is domain-specific, it does not generalize well, and we cannot simply reuse it for TRADR data without special modifications or further fine-tuning.

Now let us have a look at the performance of the PAFIBERT models trained on English and German TRADR. Despite the relatively small size of the training data, the models manage to achieve IBA scores of about 88% (English) and 83% (German). The English model also demonstrates quite good performance (86% IBA) on the subset of the main English TRADR test set, used to evaluate PAFIBERT trained on the FrameNet data. Notice that the IBA metric is fairer than standard accuracy: despite the fact that the subset of the TRADR test set does not contain the instances of the most frequent domain-specific *'Communication_by_protocol'* and *'Communication_response_message'* frames, which are easier to recognize due to their shortness and typical structure, the IBA score for this test set is only 2% lower.

Classifier model	Basic (EN)	Basic (DE)
# errors	30/268	42/259
Target ambiguity	22/30 (73%)	26/42 (62%)
Silly mistakes	5/30 (17%)	14/42 (33%)
Incorrect parsing	2/30 (7%)	2/42 (5%)
Incorrect translations	1/30 (3%)	-

Table 4: PAFIBERT trained on TRADR: error analysis

In order to understand why we have 5% difference in performance between the English and German frame classifiers trained on TRADR, we performed error analysis. The results are summarized in Table 4. We see that the majority of errors happens because of ambiguous targets, and the proportion of such errors is about 10% higher among the errors made by the English frame classifier. At the same time the German frame classifier makes much more the so-called silly mistakes, which encompass the cases when the assigned frame has nothing to do with the given target. We attribute the worse performance of the German classifier mostly to the fact that instead of targets we used 'frame related elements', which sometimes contain several tokens and can be confusing for the classifier. Differences between the languages (i.e. in morphology, syntax, semantics) may also be important. E.g., verbs with separable prefixes, like *'zurückkehren'* or *'vorbeikommen'*, as targets may lead to errors, as the prefixes often get disregarded. Finally, because of small test sizes, the role of chance (in)correct assignments may get exaggerated.

5.2 Adjustments of PAFIBERT

Aiming at performance improvement, we experimented with several adjustments of the PAFIBERT model trained on TRADR. Below we discuss the results and analyse the errors.

Sampling We performed a series of experiments with sampling additional training examples from the subsets of the FrameNet and SALSA corpora, which contain only instances of those frames that occur in TRADR. The FrameNet subset for sampling has 21,492 instances (about 12% of the whole FrameNet corpus), the SALSA subset - 2,486 (about 7% of the corpus). The experiments can be split into two groups. The first group includes training models with different portions of blindly sampled data. The second part involves experiments with informed sampling. Each model is trained on a mixture of TRADR and sampled data, and validated solely on TRADR data.

In the blind sampling scenario we train ten models gradually increasing the amount of additional training examples randomly chosen from the FrameNet or SALSA subsets.

# sampled inst.	PRE	REC	F1	$IBA_{0.1}$
2,149 inst. (10%)	0.92	0.90	0.90	**0.89**
4,298 inst. (20%)	0.91	0.87	0.88	0.86
6,447 inst. (30%)	0.91	0.89	0.89	0.88
8,596 inst. (40%)	0.92	0.90	0.90	**0.89**
10,746 inst. (50%)	0.92	0.89	0.89	0.88
12,895 inst. (60%)	0.91	0.89	0.89	0.88
15,044 inst. (70%)	0.92	0.89	0.89	0.88
17,193 inst. (80%)	0.91	0.88	0.88	0.87
19,342 inst. (90%)	0.92	0.88	0.89	0.87
21,492 inst. (100%)	0.91	0.90	0.90	**0.89**
Basic model (EN)	0.90	0.89	0.89	0.88

Table 5: Blind random sampling from FrameNet

The results for the English frame classifier are in Table 5. We see that there is no clear correlation between the sampled data size and performance. Three models demonstrate an improvement by 1% in comparison with the basic model, however, this difference is insignificant according to the McNe-

mar's test. A lack of positive influence of the blind sampling can be caused by the fact that the subset of the FrameNet data used for sampling only contains a small amount of really useful instances. If only a part of the subset is sampled, these instances have high chances to be left out due the randomization of the sampling procedure. In case the whole subset is sampled, the additional instances may dominate the original ones, as the FrameNet subset for sampling is much larger than TRADR.

In contrast to this, the effect of the blind random sampling on the German frame classifier is clearly positive. As Table 6 shows, having more training data leads to the IBA score increase by 4%.

# sampled inst.	PRE	REC	F1	$IBA_{0.1}$
248 inst. (10%)	0.86	0.85	0.85	0.84
497 inst. (20%)	0.88	0.88	0.87	**0.87**
745 inst. (30%)	0.87	0.87	0.86	0.86
994 inst. (40%)	0.87	0.85	0.85	0.84
1,243 inst. (50%)	0.89	0.88	0.88	**0.87**
1,491 inst. (60%)	0.89	0.88	0.87	**0.87**
1,740 inst. (70%)	0.89	0.88	0.87	**0.87**
1,988 inst. (80%)	0.89	0.88	0.88	**0.87**
2,237 inst. (90%)	0.89	0.88	0.87	**0.87**
2,486 inst. (100%)	0.90	0.88	0.88	**0.87**
Basic model (DE)	0.84	0.84	0.83	0.83

Table 6: Blind random sampling from SALSA

To get an explanation why blind sampling has a different impact on the two classifiers, we plot the learning curves that show how training and validation losses depend on the proportion of sampled data. As Figure 1 shows, adding training instances from FrameNet and SALSA does not lead to validation loss decrease and better generalization ability of the models. Notice that even without sampling the gap between the two curves in each plot is large, with training losses being close to zero, which is usually interpreted as overfitting. This finding lead us to check the learning curves of PAFIBERT trained on the much larger FrameNet data. The overfitting problem occurs in that case, too (see Appendix A). To tackle the overfitting issue, we tried out several experiments with increased dropout rate and fewer training epochs, but they only led to the IBA score decrease. We conclude that some fundamental changes in PAFIBERT's architecture would be needed to avoid overfitting.

In both plots in Figure 1 the validation loss grows together with the number of sampled examples. This means that even if the models continue making correct predictions, their confidence sinks. In case of the German frame classifier this growth is not so rapid, which can probably be explained by the fact that the SALSA subset for sampling is much smaller than the corresponding FrameNet subset. Knowing that IBA is actually improving, we hypothesize that sampled data from an unrelated domain can be helpful, but the right amount of these instances and their quality criteria are rather difficult to determine.

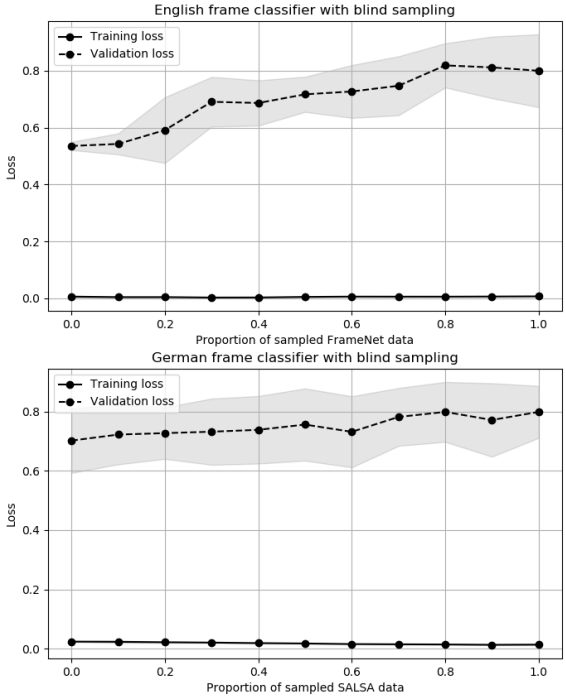


Figure 1: Learning curves of English and German frame classifies with blind sampling

The main disadvantage of blind sampling is that the instances are picked out regardless of their distribution in both original training data and data held out for sampling, which may aggravate the imbalance problem.

To overcome this, we tried two approaches using informed sampling. One is balancing sampling. It assumes sampling for each class no more than the maximum number of instances of the most common frame in the TRADR training data. The approach is supposed to deal with the class imbalance problem. However, this method also has a potential disadvantage. In case the number of original TRADR utterances is small, and the number of sampled instances is much larger, with their targets being different from those in the original utterances, the model will be biased towards the dominating training samples and thus prone to misclassification of the TRADR test examples. To avoid this we introduce equal sampling, which has an additional

constraint that the number of sampled examples cannot exceed the number of the original ones.

The scores in Table 7 show that the informed sampling does not produce the expected positive effect on the English frame classifier.

Sampling type	PRE	REC	F1	$IBA_{0.1}$
Balancing: 10,902 inst. ($\approx$ 51%)	0.91	0.88	0.88	0.87
Equal: 1,622 inst. ($\approx$ 7.5%)	0.92	0.89	0.89	0.88
Basic model (EN)	0.90	0.89	0.89	0.88

Table 7: Informed random sampling from FrameNet

However, as Table 8 demonstrates, in case of the German frame classifier the informed sampling clearly has a positive influence. Its significance was confirmed by the corresponding McNemar's tests. Balancing sampling helps to reduce the number of silly errors, as well as errors caused by ambiguous target expressions. The latter reduction is mostly due to a better recognition of ambiguous target expressions represented by single tokens, reflexive verbs and verbs with separable prefixes. The difference in performance between the balancing and equal sampling approaches was insignificant according to McNemar's test.

Sampling type	PRE	REC	F1	$IBA_{0.1}$
Balancing: 2,375 inst. ($\approx$ 96%)	0.89	0.89	0.88	0.88
Equal: 611 inst. ($\approx$ 25%)	0.91	0.91	0.90	**0.90**
Basic model (DE)	0.84	0.84	0.83	0.83

Table 8: Informed random sampling from SALSA

So, we see that sampling failed to produce any positive effect on the English frame classifier, but worked for the German one. We hypothesize that this happens to a large extent because sampling mostly helps to resolve simple mistakes, but is less effective in cases where disambiguation is necessary. More complex morphology of German may also be a reason why additional training examples proved to be more useful.

Extra features Our next goal is to check if extending BERT embeddings with extra features has any positive impact on the performance of PAFIBERT. We divide the features into two groups: lexical features that include POS tags and subword masks, and discourse features represented by speaker tags and dialogue acts. Our modifications of the original architecture by Tan and Na (2019) are given in Appendix, Figures 3 and 4.

The introduction of lexical features is motivated by the following reasons. First, we have cases, when the POS tag of a target may be important to differentiate one frame from another. E.g., in the utterance *"Can you position yourself onto the track?"* the target *'position'* is a verb and evokes the *'Placing'* frame, while in the utterance *"What's your current position?"* *'position'* is a noun that induces the frame *'Locale_by_collocation'*. Second, BERT tokenization splits the tokens that are not included in the tokenizer vocabulary, and sometimes it happens that some parts of a token lie outside of the target's context window.

POS tagging was done with a tagger from the *Python SpaCy* library (Honnibal and Montani, 2017). There are 19 coarse-grained tags that follow the Universal Dependencies scheme. We add two more tags to this set: SPECIAL to mark special tokens used by BERT and separate them from 'normal' ones, and PAD for padded tokens. If a token gets split by the tokenizer, each sub-token is assigned the POS tag of the original word. Our subword masks are bit vectors where all sub-tokens are marked with ones, and intact tokens - with zeros.

Embeddings for lexical features are trained together with the model. They are concatenated with the BERT model output, namely with (sub)token vectors, and used as input for the position-based attention layer of PAFIBERT. As (sub)token representations get longer, we have to increase the size of the first linear layer of PAFIBERT accordingly.

The second group of additional features includes discourse features, namely the speaker tag and dialogue act type, which also can be useful for frame disambiguation. E.g., given a short utterance *"Try it"* with the target *'try'*, the classifier may have difficulties labeling it, because to assign the correct frame it needs to know the perspective, i.e. the speaker. If the speaker is the team leader, then the correct frame is *'Attempt_suasion'*, if it is an operator, then it should be the *'Attempt'* frame. The information about the dialogue act type can be used to strengthen the impact of the speaker tag, because there exist a strong correlation between the speaker and the dialogue act in the tradr dialogues (Anikina and Kruijff-Korbayová, 2019).

Following Anikina and Kruijff-Korbayová (2019), we use three labels to encode the speakers:

MC for the mission commander, TL for the team leader and OPERATOR for the rest of the team.

As for dialogue acts, we use 12 labels based on the ISO-24617-2 guidelines Bunt (2019), with a few modifications. Eight tags correspond to those used in Anikina and Kruijff-Korbayová (2019): *'Affirmative'*, *'Confirm'*, *'Contact'*, *'Disconfirm'*, *'Inform'*, *'Negative'*, *'Question'* and *'Request'*. The other four labels are *'Communication Management'*, *'Time Management'*, *'Discourse Structuring'* and *'Social Obligations'*.

Embeddings for discourse features are trained jointly with the model. Since they characterize the whole utterance and not separate (sub)tokens, we concatenate them with the output of the PAFIBERT position-based attention layer. We increase the size of the first linear layer in the model accordingly.

The performance of the English frame classifier trained on the data enriched with lexical and dialogue features is given in Table 9. We test the features separately and in combinations. We see that taken separately, the features do not bring any improvement, and sometimes the scores are actually slightly worse than the score achieved by the basic classifier. The combination of POS tags and subword masks seems to increase the performance by 1%, but the difference is insignificant according to the McNemar's test.

Feature	PRE	REC	F1	$IBA_{0.1}$
POS tag	0.89	0.88	0.88	0.87
Subword mask	0.89	0.88	0.87	0.87
POS tag + Subw. mask	0.91	0.90	0.90	**0.89**
Speaker	0.89	0.88	0.88	0.88
Dialogue act	0.89	0.87	0.87	0.86
Speaker + Dialogue act	0.90	0.88	0.88	0.88
POS tag + Subw. mask + Sp.	0.88	0.88	0.87	0.87
Basic model (EN)	0.90	0.89	0.89	0.88

Table 9: Extra features: English frame classifier

As for the German frame classifier, we tested only the impact of extra lexical features. Dialogue features were not used, as the current data does not include speaker and dialogue act annotations. The results were similar to those demonstrated by the English frame classifier with extra lexical features. We do not include them here due to space constraints. They are available in Appendix B.

It is difficult to say why neither lexical nor discourse features lead to performance improvement. One of possible reason is that our learned feature embeddings are rather short (2-4 neurons) in comparison with input embeddings (768 neurons) or context-target embeddings (1536 neurons), so their impact on the whole (sub)token/utterance representations is actually negligible or even confusing. We think that in order to get a better estimation of the role of additional features, some further experiments with more data are necessary.

6 Conclusion and Future Work

We investigated the potential of frame semantics as a meaning representation framework for English and German dialogues in the domain of robot-assisted disaster response team communication. We found semantic frames convenient for capturing the meaning of an utterance depending on the target - the approach is span-based and does not require complex data annotation or pre-processing.

We reused the PAFIBERT model on the TRADR data and achieved an IBA score of 88%–90% on the test sets. Our results are comparable with those reported by Tan and Na (2019), who trained their models on the much larger FrameNet corpus. However, being a powerful model, PAFIBERT memorized the small TRADR training data, leading to overfitting and thus lack of generalization.

We also studied the impact of sampling additional training instances from an unrelated domain on the classifier's performance, and found that it was useful only for the German frame classifier. Error analysis indicates that sampling is beneficial for handling silly errors, but rather ineffective for cases that require disambiguation. We did not perform any experiments with over- and/or undersampling which imply sampling from the original dataset and are often used with imbalanced data. This can be a subject for further research. Especially interesting is an approach that assumes generating synthetic training instances, e.g., embeddings incorporating the targets with their contexts.

In contrast to our expectations, both lexical and discourse features failed to demonstrate a positive influence on the models' performance.

Error analysis showed that the largest group of errors is due to ambiguous targets, many of which evoke semantically close frames. The problem of disambiguation requires more research in order to improve the performance of the models.

Acknowledgments

This work is part of the project "A-DRZ: Setting up the German Rescue Robotics Center", funded by the German Ministry of Education and Research

(BMBF), grant No. I3N14856.[2] We would like to thank our A-DRZ colleagues for discussions, Daria Fedorova for data annotation and the IWCS 2021 reviewers for valuable comments which we hope helped us to improve the paper.

References

Tatiana Anikina and Ivana Kruijff-Korbayova. 2019. Dialogue act classification in team communication for robot assisted disaster response. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 399–410, Stockholm, Sweden. Association for Computational Linguistics.

Tatiana Anikina and Ivana Kruijff-Korbayová. 2019. Dialogue act classification in team communication for robot assisted disaster response. In *Proceedings of SIGDIAL 2019*.

Collin F. Baker, Michael Ellsworth, and Katrin Erk. 2007. SemEval-2007 Task 19: Frame semantic structure extraction. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 99–104.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING-ACL '98: Proceedings of the Conference*, pages 86–90, Montreal, Canada.

Claude Barras, Edouard Geoffrois, Zhibiao Wu, and Mark Liberman. 1998. Transcriber: a free tool for segmenting, labeling and transcribing speech. In *First International Conference on Language Resources and Evaluation (LREC)*, pages 1373–1376.

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.

Harry Bunt. 2019. *Guidelines for using ISO standard 24617-2*. [s.n.]. TiCC TR 2019–1.

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. 2006. The SALSA corpus: a German corpus resource for lexical semantics. In *LREC*, pages 969–974.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*, 22(2):249–254.

Bonaventura Coppola, Alessandro Moschitti, Sara Tonelli, and Giuseppe Riccardi. 2008. Automatic Framenet-based annotation of conversational speech. In *2008 IEEE Spoken Language Technology Workshop*, pages 73–76.

[2] https://www.rettungsrobotik.de

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956, Los Angeles, California. Association for Computational Linguistics.

Dipanjan Das and Noah A Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444.

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.

Katrin Erk. 2005. Frame assignment as word sense disambiguation. In *Proceedings of IWCS*, volume 6.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32.

Charles J. Fillmore. 1982. *Frame semantics*, pages 111–137. Hanshin Publishing Co., Seoul, South Korea.

FrameNet. 2021. The official website for the FrameNet project. https://framenet.icsi.berkeley.edu/fndrupal/. Accessed: 2021-02-03.

Vicente García, Ramón Alberto Mollineda, and José Salvador Sánchez. 2009. Index of balanced accuracy: A performance measure for skewed class distributions. In *Iberian conference on pattern recognition and image analysis*, pages 441–448. Springer.

Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. 2017. Out-of-domain FrameNet semantic role labeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 471–482, Valencia, Spain. Association for Computational Linguistics.

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.

Matthew Honnibal and Ines Montani. 2017. SpaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Richard Johansson and Pierre Nugues. 2007. LTH: Semantic structure extraction using nonprojective dependency trees. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 227–230, Prague, Czech Republic. Association for Computational Linguistics.

Aditya Kalyanpur, Or Biran, Tom Breloff, Jennifer Chu-Carroll, Ariel Diertani, Owen Rambow, and Mark Sammons. 2020. Open-domain frame semantic parsing using transformers.

Ivana Kruijff-Korbayová, Francis Colas, Mario Gianni, Fiora Pirri, Joachim de Greeff, Koen Hindriks, Mark Neerincx, Petter Ögren, Tomáš Svoboda, and Rainer Worst. 2015. TRADR project: Long-term human-robot teaming for robot assisted disaster response. *KI - Künstliche Intelligenz*, 29(2):193–201.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Miriam R L Petruck. 2019. Meaning representation of null instantiated semantic roles in FrameNet. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 121–127, Florence, Italy. Association for Computational Linguistics.

Christian Raymond, Kepa Joseba Rodriguez, and Giuseppe Riccardi. 2008. Active Annotation in the LUNA Italian Corpus of Spontaneous Dialogues. In *LREC*.

Eugénio Ribeiro, Andreia Sofia Teixeira, Ricardo Ribeiro, and David Martins de Matos. 2020. Semantic frame induction as a community detection problem.

Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*. International Computer Science Institute, Berkeley, California. Distributed with the FrameNet data.

Natalia Skachkova and Ivana Kruijff-Korbayova. 2020. Reference in team communication for robot-assisted disaster response: An initial analysis. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 122–132, Barcelona, Spain (online). Association for Computational Linguistics.

Sang-Sang Tan and Jin-Cheon Na. 2019. Positional attention-based frame identification with BERT: A deep learning approach to target disambiguation and semantic frame selection. *arXiv preprint arXiv:1910.14549*.

Alaa Tharwat. 2020. Classification assessment methods. *Applied Computing and Informatics*.

Jeremy Trione, Frederic Bechet, Benoit Favre, and Alexis Nasr. 2015. Rapid FrameNet annotation of spoken conversation transcripts. In *Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11)*, London, UK. Association for Computational Linguistics.

Christian Willms, Constantin Houy, Jana-Rebecca Rehse, Peter Fettke, and Ivana Kruijff-Korbayová. 2019. Team communication processing and process analytics for supporting robot-assisted emergency response. In *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 216–221. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. HuggingFace's Transformers: State-of-the-art natural language processing.

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.

Hongyan Zhao, Ru Li, Fei Duan, Zepeng Wu, and Shaoru Guo. 2018. TSABCNN: Two-stage attention-based convolutional neural network for frame identification. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 289–301, Cham. Springer International Publishing.

A PAFIBERT: training and validation losses

Figure 2 shows the changes of training and validation losses with each training epoch of our re-implementation of the original PAFIBERT according to Tan and Na (2019). One can see that the model is powerful enough to memorize the training data by the end of the training, but, judging by the gap between the two curves, it has difficulties in generalizing and making confident predictions. Starting from the second epoch, the validation loss almost does not change, and it is also larger than the validation loss of the frame classifiers trained on TRADR, which can probably be attributed to the fact that FrameNet has many more classes than TRADR.

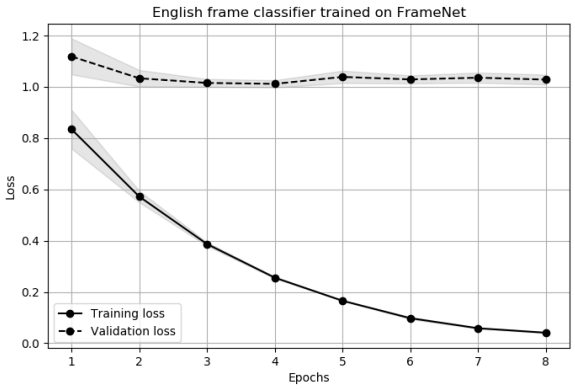


Figure 2: Traning and validation losses of the original PAFIBERT model

B German frame classifier with lexical features

Table 10 shows the performance of the German frame classifier with extra lexical features. Extending token embeddings with the corresponding POS tag embeddings seems to have a small positive effect on the IBA score, however, it is not significant according to the McNemar's test. Adding subword mask embeddings as well as using the combination of two extra features also does not seem to influence the performance of the classifier. Finally, we try extending token embeddings with POS tag embeddings together with the equal sampling from SALSA. However, the equal sampling, which earlier helped us achieve the IBA score of 90%, fails to provide the anticipated positive effect - the current score is only 84%, and the McNemar's test interprets the improvement as insignificant. We conclude that adding lexical features confuses the frame classifier, so that sampling looses its positive effect on the accuracy.

Model	PRE	REC	F1	$IBA_{0.1}$
POS tag	0.87	0.86	0.86	**0.85**
Subword mask	0.86	0.84	0.84	0.83
POS tag + subword mask	0.84	0.84	0.83	0.82
POS tag + equal sampling	0.87	0.85	0.85	0.84
Basic model (DE)	0.84	0.84	0.83	0.83

Table 10: Extra features: German frame classifier

C Team communication example

Table 11 shows one of the TRADR dialogues. The first column presents the speakers, the second - the utterances that sometimes also contain output from the *Transcriber* tool (Barras et al., 1998) (e.g., [ent=unk.skippable]), the third - the assigned frames depending on the targets (given in bold). According to our annotation approach, each utterance may contain several targets and thus evoke several frames. To make the dependencies between the targets and the corresponding frames clear, we annotated only one target-frame pair per row. This resulted in creating copies of the utterances containing several targets. They are given in *italics*. Most of the targets in the example dialogue are verbs which reflects our focus on various activities performed as part of the rescue mission.

The team communication example also illustrates two out of ten frames that we had to introduce during the annotation, as the FrameNet database (FrameNet, 2021) is not exhaustive, and it was not always possible to adapt the available frames to new phenomena. These two frames are *'Communication_by_protocol'* and *'Communication_response_message'*. They are domain-specific and are actually the most frequent in the whole TRADR corpus. Other eight frames that were introduced are rare. Table 12 contains the definitions and examples of all the new frames that we introduced. Frame elements are given in CAPITAL letters. We have not worked out their definitions yet. This is planned for future work.

The presented dialogue also has instances of the FrameNet frames that we adapted. Assigning the frame labels, sometimes it was impossible to follow the frame definitions given in the FrameNet database strictly. Considering that FrameNet is not exhaustive and that we were cautious to introduce too many new frames, we had to interpret

certain frame definitions in a more relaxed way. E.g., FrameNet defines the frame *'Existence'* as *"An Entity is declared to exist, generally irrespective of its position or even the possibility of its position being specified. ⟨...⟩ This frame is to be contrasted with Presence, which describes the existence of an Entity in a particular (and salient) spacio-temporal context, and which also entails the presence of an observer who can detect the existence of the Entity in that context."* We used *Existence* in a more straightforward way, namely with a reference to some news, findings, updates, etc. are present/available at a certain moment. Other adapted frames present in the dialogue are *Presence* and *Identity*. We do not present a full list of the adapted frames here, as there are quite many of them.

Notice that some utterances in the dialogue do not contain targets, as they are elliptical. In such cases we usually try to infer the missing elements, and assign the frame label that corresponds to the 'restored' utterance.

D Approaches to automatic frame assignment: a summary

Table 13 summarizes the characteristics of most of the frameworks mentioned in Section 3. The frameworks are given in chronological order, which helps illustrate the shift from the rule- and/or feature-based approaches to the embeddings-based ones, as well as the replacement of more 'traditional' classifiers with neural networks. The introduction of embeddings allowed to avoid manual feature engineering, and helped achieve better or comparable results with much less effort. However, the embeddings (even contextual ones, like ELMo or BERT) are still not able to deal with sense ambiguity effectively, which is one of the main problems in automatic frame assignment task.

The last row shows the performance of the frameworks. Those that have scores given were trained on the FrameNet corpus (versions may differ) and evaluated on one of the most commonly used Das test set (Das and Smith, 2011), which represents a part of FrameNet 1.5 data. Unfortunately, it is not always possible to compare the frameworks directly, as some researchers report F-score as a performance measure, others - accuracy. Five frameworks were evaluated on different test data, and we therefore omit their scores.

TL	Andreas, Andreas from Markus, **come in**.	Communication_by_protocol
OP	Yes, Andreas **come in**.	Communication_by_protocol
	⟨...⟩	
OP	Yes, for information, I am **ready** [EHM].	Activity_ready_state
	Shall I go ahead with my search command, or begin?	Desirable_event
	*Shall I **go ahead** with my search command, or begin?*	Activity_ongoing
	*Shall I go ahead with my search command, or **begin**?*	Activity_start
TL	Yes, **begin** immediately without possible – least possible time delay, to [EHM] have a higher chance for person rescue.	Activity_start
	*Yes, begin immediately without possible – least possible time delay, to [EHM] have a higher **chance** for person rescue.*	Likelihood
OP	Yes, **understood**, I begin with the search.	Communication_response_message
	*Yes, understood, I **begin** with the search.*	Activity_start
	⟨...⟩	
TL	Andreas from Markus, **come in**. [ent=unk.skippable]	Communication_by_protocol
OP	Yes, Andreas, **come in**.	Communication_by_protocol
TL	[ent=unk.skippable] **Are there** already any noteworthy findings? [ent=unk.skippable]	Existence
OP	**Negative**. No noteworthy findings. [ent=unk.skippable]	Communication_response_message
	Negative. No noteworthy findings. [ent=unk.skippable]	Existence
TL	Yes, **understood**.	Communication_response_message
	[ent=unk.skippable] Daniel, Daniel from Markus, **come in**.	Communication_by_protocol
	[ent=unk.skippable] Andreas from Markus, **come in**.	Communication_by_protocol
	⟨...⟩	
OP	Andreas, Markus from Andreas, **come in**.	Communication_by_protocol
TL	Andreas, **come in**.	Communication_by_protocol
OP	On first floor in the smoke **found** a barrel, green, labeled as environmentally hazardous material.	Locating
TL	Yeah, **can** you [unintelligible] whether anything is leaking?	Capability
	*Yeah, can you [unintelligible] whether anything is **leaking**?*	Fluidic_motion
OP	Yeah. It **is** a 200 liter barrel, whether anything is leaking I cannot currently tell.	Identity
	*Yeah. It is a 200 liter barrel, whether anything is **leaking** I cannot currently tell.*	Fluidic_motion
	*Yeah. It is a 200 liter barrel, whether anything is leaking I **can**not currently tell.*	Capability
	*Yeah. It is a 200 liter barrel, whether anything is leaking I cannot currently **tell**.*	Becoming_aware
TL	[EHM] Any thermal emission?	Presence
OP	No thermal emission.	Presence
TL	Okay. Priority on **continuing** person search.	Activity_ongoing
	Andreas from Markus, priority on **continuing** person search.	Activity_ongoing

Table 11: English TRADR dialogue annotated with semantic frames

Be_piece_of	
Inherits from:	Being_included
Definition:	A PART is considered to be a constituent of some entity described by the WHOLE. The relation is seen from the point of view of the PART.
Examples:	*I can also see [PART fragments] that **belong to** [WHOLE the building] [PART lying around here].*
Being_reasonable	
Inherits from:	Gradable_attributes
Definition:	Certain BEHAVIOR of PROTAGONIST is seen as practical and sensible.
Examples:	*As I can't see anything at the moment, it would definitely **make sense** if [PROTAGONIST you] [BEHAVIOR let the* UAV *guide you to some other points as soon as they've started again].*
Communication_by_protocol	
Inherits from:	Communication
Definition:	A COMMUNICATOR speaks to an ADDRESSEE using the phrases of special form (protocol) to establish/finish the conversation by radio.
Examples:	*[COMMUNICATOR Team leader] [ADDRESSEE for Tango].* *[COMMUNICATOR Team leader], **here** is [ADDRESSEE Tango].* *[COMMUNICATOR* UAV*] [ADDRESSEE to* UGV-1*] please **answer**.* *[COMMUNICATOR* UAV*] **speaking** [ADDRESSEE* IDI*].*
Communication_fragment	
Inherits from:	None
Definition:	An auxiliary frame which serves the purpose of marking conversational fillers and sequences with unclear meaning. The frame is characterized by conflation of target and FRAGMENT itself.
Examples:	*[FRAGMENT Also... I'm with... erm...]* *[FRAGMENT Eeh eeh my my my...]* *[*FRAGMENT *Whether a person or its... below at the bottom edge there's a...]*
Communication_response_message	
Inherits from:	Statement
Definition:	A COMMUNICATOR gives a short usually positive or negative reply to an ADDRESSEE's question or request. Sometimes a TOPIC is also mentioned.
Examples:	***Roger** [TOPIC that], [ADDRESSEE team leader].* ***Okay**.* ***Yes** [COMMUNICATOR by ground operator 1].*
Correction	
Inherits from:	Communication
Definition:	A COMMUNICATOR informs an ADDRESSEE that what the PATIENT has communicated is not right, true or suitable by providing the corrected version of the MESSAGE.
Examples:	*[COMMUNICATOR I] have to **correct** [PATIENT myself]: [MESSAGE* UGV-1*].*
Face_direction	
Inherits from:	State
Definition:	An ENTITY faces a particular DIRECTION.
Examples:	*For your information: [ENTITY it]'s **looking** [DIRECTION towards south].*
Lead	
Inherits from:	Cause_to_perceive
Definition:	An ENTITY leads in a particular DIRECTION or to some GOAL.
Examples:	*[ENTITY The stairwell] **leads** [DIRECTION upwards].* *There's smoke development at [ENTITY the first stairs] that **go** [DIRECTION upwards].*
Level_of_clarity	
Inherits from:	Gradable_attributes
Definition:	A DEGREE to which a REPRESENTATION is clear and detailed.
Examples:	*Yes, [REPRESENTATION the pictures] aren't [DEGREE very] **sharp**.*
Level_of_substance	
Inherits from:	Gradable_attributes
Definition:	A DEGREE of smoke in the air at some LOCATION.
Examples:	*It's actually [DEGREE quite] **smoky** [LOCATION* DNI*].*

Table 12: TRADR: new frames

Characteristic features	Framework											
	Erk (2005)	LTH (2007)	LUNA (2008)	SEMAFOR (2010)	Hermann et al. (2014)	SimpleFrameId (2017)	Open-Sesame (2017)	Yang & Mitchell (2017)	TSABCNN (2018)	PAFIBERT (2019)	Ribeiro et al. (2020)	Kalyanpur et al. (2020)
hand-crafted rules		✓	✓	✓	✓							
hand-crafted features	✓	✓	✓	✓	✓							
kernels			✓									
parsing		✓	✓	✓	✓	✓					✓	
embeddings					✓	✓	✓	✓	✓	✓	✓	✓
Naive Bayes classifier	✓											
SVM		✓	✓									
conditional log-linear model				✓	✓							
neural network						✓	✓	✓	✓	✓		✓
CRF							✓	✓				
clustering											✓	
graph structure											✓	
Frame assignment accuracy	n/a	n/a	n/a	82.97*	88.41	87.63	70.9*†	88.2	89.72	89.57	n/a	n/a

Table 13: Comparison of various frame-semantic parsing frameworks; scores marked with '*' stand for F-score (the authors do not report accuracy); 'n/a' means that the authors used a test set different from Das and Smith (2011); † stands for joint evaluation of frame assignment and argument identification

Corpus	English TRADR	German TRADR	FrameNet	SALSA
Domain	team communication in disaster response	team commumication in disaster response	mostly business, politics, economics related texts	newspaper texts
# inst.	2,930	2,813	199,508	35,236
# tokens	31,211	33,625	4,751,140	838,307
# classes	190 (177 occur in FrameNet)	152 (80 occur in SALSA)	1,014	880
Avg. sent. len.	8.68	6.99	22.92	21.78
# LUs	434 (280 occur in FrameNet)	-	8,333	-
% ambig. LUs wrt. # LUs	14.98	-	15.61	-
% ambig. LUs wrt. all inst.	52.90	-	34.99	-

Table 14: Corpora comparison

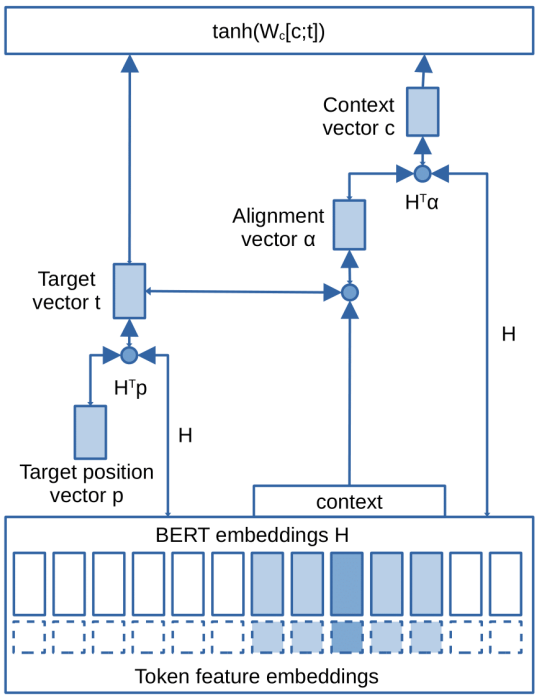


Figure 3: Adding lexical features (dashed borders) to PAFIBERT

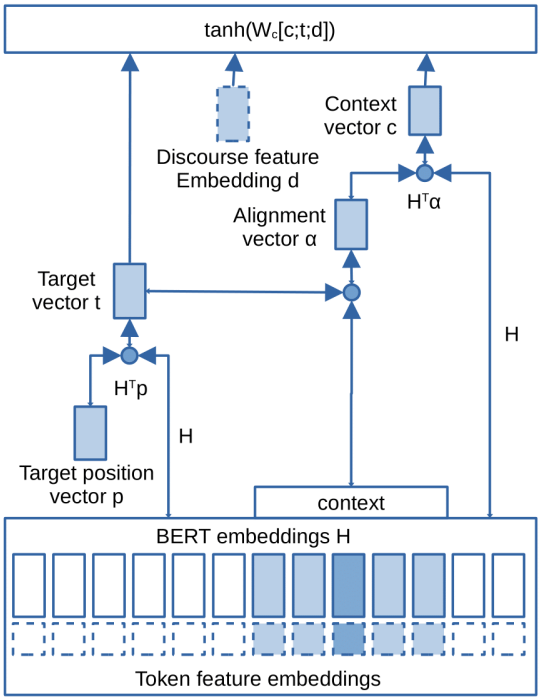


Figure 4: Adding lexical and discourse features (dashed borders) to PAFIBERT

Implicit representations of event properties within contextual language models: Searching for "causativity neurons"

Esther Seyffarth[†] **Younes Samih**[†] **Laura Kallmeyer**[†] **Hassan Sajjad**[‡]
[†]Heinrich Heine University Düsseldorf, Düsseldorf, Germany
[‡]Qatar Computing Research Institute, Hamad Bin Khalifa University
{seyffarth,samih,kallmeyer}@phil.hhu.de hsajjad@hbku.edu.qa

Abstract

This paper addresses the question to which extent neural contextual language models such as BERT implicitly represent complex semantic properties. More concretely, the paper shows that the neuron activations obtained from processing an English sentence provide discriminative features for predicting the (non-)causativity of the event denoted by the verb in a simple linear classifier. A layer-wise analysis reveals that the relevant properties are mostly learned in the higher layers. Moreover, further experiments show that appr. 10% of the neuron activations are enough to already predict causativity with a relatively high accuracy.[1]

1 Introduction and motivation

In natural language processing (NLP), machine learning models based on artificial neural networks have achieved impressive results in recent years, due to large amounts of available training data and powerful computing infrastructures. Contextual language models (LMs) such as ELMO (Peters et al., 2018), BERT (Devlin et al., 2019), and XLNet (Yang et al., 2019) have particularly contributed to this. However, it is oftentimes not clear which kinds of generalizations these models make, i.e., what exactly they learn. In this respect, neural networks suffer from a lack of transparency and interpretability. Recent research has started to investigate these questions. Since the successful use of neural word embeddings and LMs (e.g., Word2Vec, Mikolov et al. 2013; ELMO, Peters et al. 2018; BERT, Devlin et al. 2019) for a range of NLP/NLU tasks, it is clear that LMs capture meaning to a certain degree, in particular lexical meaning. Concerning syntactic information, work on different types of language models, in particular RNNs and transformer-based contextual language models, has shown that these models learn morphology (Liu et al., 2019a), syntactic structure and syntactic preferences to a certain degree (see Futrell and Levy, 2019; Lin et al., 2019; Hewitt and Manning, 2019; McCoy et al., 2020; Wilcox et al., 2019; Hu et al., 2020; Warstadt et al., 2020).

In this paper, we expand the question of what linguistic properties these models learn towards whether pretrained contextualized models capture more abstract semantic properties, in particular properties that contribute to the structure of the semantic representation underlying a given sentence. More concretely, we investigate whether an LM such as BERT represents whether a sentence denotes a causative event or not. If this was the case, we would expect a systematic difference between for instance BERT's neuron activations for (1-a) and for (1-b).

(1) a. Kim broke the window.
b. Kim ate an apple.

Note that the two sentences share almost no lexical elements, so the neuron activations are expected to be mostly different. Our research question is focused on whether there are systematic activation patterns that can be observed that are common to all instances of causative sentences, and others that are common to all instances of noncausative sentences, independent of sentence content.

One of the common approaches to probe neural network models is to use a probing classifier. Given a linguistic property of interest, the idea is to extract contextualized activations of units (words/phrases/sentences) relevant to the property. A classifier is then trained to learn the property by using the extracted activations as features. The performance of the classifier is taken to approximate

[1]Our datasets are available at https://github.com/eseyffarth/predicting-causativity-iwcs-2021

Proceedings of the 14th International Conference on Computational Semantics, pages 110–120
June 17–18, 2021.

the degree to which the language model learned the linguistic property. We also use probing classifiers and probe the model as a whole, its individual layers and its neurons with respect to causativity. We use the NeuroX toolkit (Dalvi et al., 2019b) to conduct the probing experiments.

We experiment using two 12-layer pretrained models, BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019), as well as a distilled version of BERT, DistilBERT (Sanh et al., 2019). Our findings and contributions are as follows: We create a novel dataset of sentences with verbs that are labeled for causativity/non-causativity. Using this dataset for probing, we show that this abstract semantic property is learned by the pretrained models. It is better represented in the higher layers of the model and, furthermore, there is a subset of appr. 10% of the neurons that encodes the property in question.

2 Related work

A number of interpretation studies have analyzed representations of pre-trained models and showed that they learn linguistic information such as part of speech tagging, semantic tagging and CCG tagging (Conneau et al., 2018; Liu et al., 2019a; Tenney et al., 2019a,b; Voita et al., 2019). A typical procedure to analyze representation is a post-hoc analysis using a probing classifier. It has been shown that word-level concepts are learned at lower layers while sentence-level concepts are learned at higher layers (Liu et al., 2019b). Dalvi et al. (2019a) extended the layer-level analysis towards individual neurons of the network. They proposed linguistic correlation analysis (`LCA`) to identify neurons with respect to a linguistic property. Durrani et al. (2020); Dalvi et al. (2020) later used `LCA` to analyze pre-trained models in the context of linguistic learning and redundancy in the network respectively.

In this work, we also aim to analyze pre-trained models at model-, layer- and neuron-level using post-hoc analysis methods. Different from others, we concentrate on an abstract, structure-building semantic property, namely causativity of events. Our focus is on *lexical causatives*, that is, verbs whose lexical meaning has a causative aspect (Dowty, 1979). In Dowty's aspect calculus, such verbs are analyzed as $[\phi$ CAUSE $\psi]$, where ϕ and ψ are sentences and causation is a "two-place sentential connective", notably even for sentences that only contain a single verb phrase. Thus, *John killed Bill* is decomposed as in (2) (Dowty, 1979, p. 91).

(2) [[*John does something*] CAUSE [BECOME¬[*Bill is alive*]]]

The "semantically bipartite" nature of causative verbs means that sentences with such verbs actually express not one event, but two subevents, one being the causing event and the other one being the caused event, or result, of the first. This event structure is a challenge to model with NLP systems when no superficial indicators for causativity are available. While there are verbs that are lexically causative (such as *refresh*) and verbs that are lexically noncausative (such as *prefer*), there are also verbs that vary in their causativity depending on the context in which they appear (such as *open*). Our goal is to determine to what extent the causativity or noncausativity of these types of verbs is implicitly learned by large language models.

3 Method

Over the last years, there has been an increasing interest in assessing linguistic properties encoded in neural representations. A common method to reveal these linguistic representations employs diagnostic classifiers or probes (Hupkes et al., 2018). A common diagnostic classifier is a linear classifier trained for the underlying linguistic task, using the activations generated from the trained neural network model as features. The performance of the classifier is used as a proxy to measure the amount of linguistic information present in the activations. We also use a linear classifier for probing.

Consider a pre-trained neural network model $\mathbf{M}$ with L layers: $\{l_1, l_2, \ldots, l_L\}$, where each layer l_i is of size H. Given a dataset $\mathbb{D} = \{s_1, s_2, ..., s_T\}$ consisting of T sentences, the contextualized embedding of sentence s_j at layer l_i is $\boldsymbol{z}_j^i = l_i(s_j)$. In pretrained models like BERT, a special token [CLS] is appended with every training instance during training. The token is later optimized for sentence embedding during transfer learning (Devlin et al., 2019). We consider the representations of [CLS] for sentence embedding in this study. The [CLS] representation extracted from various layers is used as input features to the probing classifier.

Model-level probing: To assess to what extent a linguistic property is learned in the model, we first take the sentence representations of all layers as features for linear classification, i.e., all $\boldsymbol{z}_j^i$ for

$1 \leq i \leq L$ and $1 \leq j \leq H$. The classifier is trained by minimizing the following loss function:

$$\mathcal{L}(\theta) = -\sum_{j} \log P_\theta(t_{s_j}|s_j) \quad (1)$$

where t_{s_j} is the predicted label for sentence s_j. In this work, binary labels are used to encode whether the property is present in a sentence or not.

Layer-level probing: Here, we question how much individual layers of a model represent our property of interest. We train a linear classifier on the activations of each individual layer. The performance of each layer serves as a proxy to how much information it encodes with respect to our property.

Neuron-level probing: While the layer-level probing tells about how much linguistic information is learned in a layer, it does not tell about the learning of individual neurons in the network. It is possible that while a particular layer performs best in the layer-level probing, the best neurons learning about the linguistic property are spread across many layers. In neuron-level probing, we aim to identify the most salient neurons across the network that learn the linguistic property at hand.

We follow the linguistic correlation analysis method (`LCA`) of Dalvi et al. (2019a) to conduct this analysis. Given representations of the model as in the model-level probing, `LCA` trains an ElasticNet (Zou and Hastie, 2005) classifier, and provides a salient list of neurons with respect to the linguistic property. ElasticNet provides a balance between selecting very focused localized features and distributed features (here: neurons). Equation (2) gives the loss function:

$$\begin{aligned}\mathcal{L}(\theta) \quad = \quad & -\textstyle\sum_j \log P_\theta(t_{s_j}|s_j) \\ & +\lambda_1\|\theta\|_1 + \lambda_2\|\theta\|_2^2\end{aligned} \quad (2)$$

where λ_1 and λ_2 are parameters, for which we use the suggested value of 0.00001 (Dalvi et al., 2019a).

4 Data

To prepare our datasets, we create different sets of verbs that are labeled for (non)causativity, and then use them as seeds to collect sentences from a corpus to be used as input to the classifier.

4.1 Verb set selection

Causative and noncausative verbs We collect a set of English verbs that are either always causative or never causative when appearing in basic transitive sentences (NP V NP). This property is derived from VerbNet 3.3 (Kipper et al., 2000) according to the event-semantic description of each basic transitive syntactic frame in each verb class. We only consider members of VerbNet classes where either all basic transitive frames or none of them are associated with causativity. Two trained linguists manually prune the lists of causative and noncausative verbs to remove ambiguous verbs and other edge cases. This results in a list of 2157 causative and 617 noncausative verbs.

Alternating verbs We also create a set of verbs whose causativity property depends on whether they appear in transitive or intransitive sentences. This is the case for verbs in VerbNet that are marked with the "Causative" property in basic transitive syntactic frames, and with the "Inchoative" property in basic intransitive frames. These verbs participate in the causative-inchoative alternation. They represent a special case for our experiments because the classifier needs to distinguish between causative and noncausative uses of identical verbs, whereas the sets of causative and noncausative verbs are completely distinct. In this setting, the classifier cannot rely purely on the verb lemma (because alternating verbs can appear in both classes), and it also cannot rely purely on the (in)transitivity of sentences (because verbs outside the alternation can be causative in intransitive sentences). Since this makes the task more difficult, we expect the classification accuracy to be lower in this setting than in settings with non-alternating verbs.

4.2 Sentence selection

We collect three datasets for our experiments.[2] All sentences are extracted from ENCOW (Schäfer and Bildhauer, 2012; Schäfer, 2015), an English web corpus (9.6 billion tokens) annotated with dependencies created with MaltParser. Each dataset contains 40,000 sentences in the `train` portion, 5,000 sentences in `dev` and 5,000 sentences in the `test` portion. Each portion contains an equal number of causative and noncausative instances. Each test set contains sentences that were not previously seen in the train set, but not all verbs in the test set are unseen.

[2] All datasets are available at https://github.com/eseyffarth/predicting-causativity-iwcs-2021

Transitive sentences, same sentence length The first dataset (D_{tr_5}) is based on the sets of causative and noncausative verbs and contains only transitive sentences of length 5 (including punctuation). This yields a dataset where all sentences have the same basic syntactic pattern. Examples are given in (3) (root verbs in bold).

(3) a. The answer **surprised** me . *(caus)*
b. It **contains** no surprises . *(noncaus)*

Transitive sentences, varying sentence length The second dataset (D_{tr}) is based on the same verb sets, but contains sentences of varying lengths between 5 and 20 tokens. Examples are given in (4).

(4) a. This **affects** the calculation . *(caus)*
b. I **envy** you in that respect ! *(noncaus)*

Intransitive and transitive sentences, varying length The third set (D_{all}) is based on the verb set that includes verbs in the causative-inchoative alternation. Sentences in D_{all} are either transitive or intransitive and have a length between 5 and 20 tokens. Again, each portion contains an equal number of causative and noncausative instances, consisting of verbs of all three types (alternating, always causative, always noncausative). Examples are given in (5); note that (5-e) and (5-f) share the same alternating root verb.

(5) a. I **bring** a book ! *(caus)*
b. Everything about them **intimidates** . *(caus)*
c. Each layer **had** its own opacity . *(noncaus)*
d. A total of 24 people **attended** . *(noncaus)*
e. He **opened** the pack . *(caus)*
f. The main console **opens** . *(noncaus)*

5 Evaluation

5.1 Experimental Settings

Pre-trained models We conduct experiments using three transformer-based pre-trained language models: BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and XLNet (Yang et al., 2019). The BERT model is an auto-encoder trained with two unsupervised objectives: masked word prediction and next sentence prediction. It is pre-trained on Wikipedia text and BooksCorpus (Zhu et al., 2015), and comes with hundreds of millions of parameters. It contains an encoder with 12 Transformer blocks, hidden size of 768, and 12 self-attention heads. DistilBERT is an approximate distilled version of BERT. It is comprised of 6 encoder layers while retaining 97% of BERT performance. We also employ XLNet-base in all our experiments. Although it is trained with the same parameter configurations as BERT-base, it uses improved training methodology based on a permutation auto-regressive objective function.

Data	BERT	DistilBERT	XLNet
D_{tr_5}	95.24	93.34	90.92
D_{tr}	89.48	87.28	88.84
D_{all}	85.28	83.96	86.00

Table 1: Model-level results (accuracy) using all neurons for classification

Since we are interested in analyzing sentence representations, we use the representation of the [CLS] token. However, the representation of [CLS] is not optimized for sentence embedding in the pre-trained models. In order to tune it for sentence representation, we fine-tune the pre-trained model on a sentence classification task, the Stanford sentiment treebank (Socher et al., 2013). We understand that by fine-tuning the pre-trained model, the representations of the network are tuned for the task. An alternate strategy is to use average activations of words in a sentence as sentence representation. We did not explore it in this paper.

Probing Classifier We train a linear classifier using a categorical cross-entropy loss, optimized using Adam. For neuron-level analysis, we used elastic-net regularization. We used the recommended values of elastic-net parameters, i.e., λ_1 and λ_2 each equal to 0.0001.

5.2 Results

Model-level Results Table 1 presents the results of using all neuron activations of the model as features for classification. The general high classification results show that the model has learned causitivity. However, as the dataset becomes hard in terms of varying sentence length and including more challenging instances with alternating verbs, the performance drops to as low as 83.96% for DistilBERT, which is still substantially better than random performance (50%).

Layer-level Results Here we want to see which layers of pretrained models learn causativity. We train our probing classifier on individual layers. Figure 1 summarizes the results. As a general trend, causitivity is best represented at the higher layers

		BERT	DistilBERT	XLNet
	Neu_a	9984	5372	9984
D_{tr_5}	Neu_t	1000/10%	540/10%	300/3%
	Acc_t	95.06	92.6	92.02
D_{tr}	Neu_t	1000/10%	540/10%	1000/10%
	Acc_t	88.70	86.06	89.24
D_{all}	Neu_t	1000/10%	540/10%	1000/10%
	Acc_t	86.48	82.66	86.8

Table 2: Selecting minimal number of neurons. Neu_a = Total number of neurons, Neu_t = Top selected neurons, Acc_t = Accuracy after retraining the classifier using only selected neurons.

	verb type	BERT	DistilBERT	XLNet
D_{tr_5}	caus	95.24	93.04	98.84
	noncaus	96.44	94.92	84.12
D_{tr}	caus	90.44	89.60	90.44
	noncaus	88.88	84.76	86.12
D_{all}	all alternating	81.52	75.43	83.05
	alt. caus	89.73	84.35	94.87
	alt. noncaus	52.59	43.97	41.38
	nonalt. caus	91.25	84.60	93.93
	nonalt. noncaus	85.36	86.03	79.28

Table 3: Accuracy per verb type and data set in all settings. D_{tr_5}, D_{tr} and D_{all} each contain an equal number of caus(ative) and noncaus(ative) instances.

of the models, which is in line with previous findings that sentence-level properties such as syntax are better learned at higher layers (Durrani et al., 2020). For all models, we see a slight drop in the performance for the last layer, which is due to the fact that the last layer is optimized for the objective function (Kovaleva et al., 2019). Compared to BERT and DistilBERT, the middle layer of XLNet consistently showed a small drop in the performance for all datasets. This trend is more prevalent in the neuron-level results. We discuss it later in this section.

Neuron-level Results We use LCA to determine a minimal set of neurons that still achieve a classification performance (Acc_t) within 2% of the performance using all the neurons of the network for classification. We additionally evaluate the effectiveness of the LCA method by comparing the classification performance using the top selected neurons with the randomly selected neurons. We found the salient neurons of LCA to perform substantially better than random neurons.

Table 2 presents the numbers of salient neurons selected for each model and for each dataset together with the resulting classification accuracy. Note that in the case of BERT and the dataset D_{all} and also for XLNet on all datasets, the accuracy increased due to the elimination of non-discriminative features.

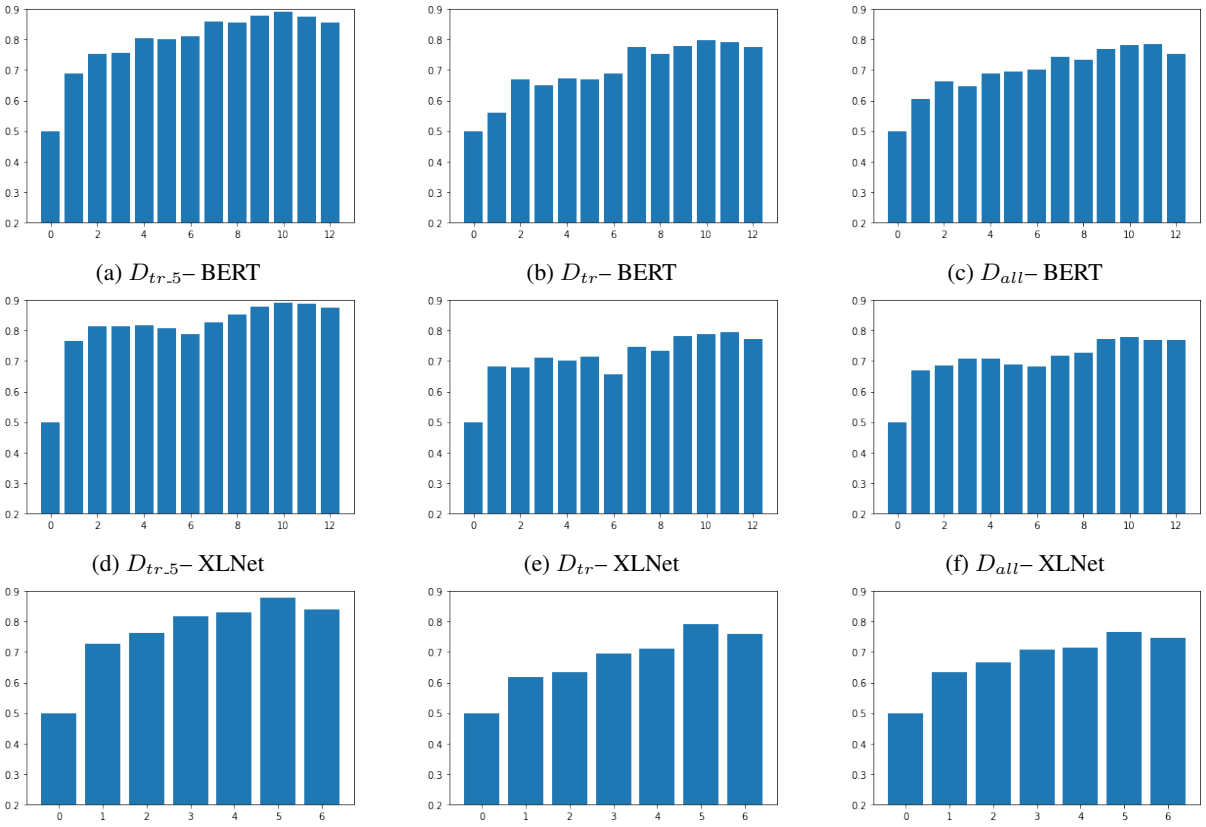
Given salient neurons with respect to our task, we observe their distribution across the model. Figure 2 summarizes the results. Across all models and datasets, the LCA method never selected any neurons from the embedding layer. This is in line with the layer-wise results where the performance using embedding layer representation is similar to random classification, i.e., no causativity information is present.

For BERT and DistilBERT, the distribution of salient neurons is skewed towards higher layers (excluding top layer), i.e., causativity information is more represented at the higher layers. XLNet presents a slightly different picture where the salient neurons selected from the middle layers are substantially lower than most of the other layers. As the task becomes harder, the contribution of lower middle layers (3-4) substantially increases while the last layer contribution drops.

The number of neurons selected from middle layers (5-6 in the case of 12 layer models and 3 in the case of 6 layer models) are substantially lower than the neighbouring layers across all models and data sets. We hypothesize that learning causitivity requires word-level and sentence-level information which is dominating at the lower and higher layers.

6 Discussion

As shown in Table 1, all classifiers performed best on D_{tr_5}. With little syntactic variation between instances in D_{tr_5}, this is the least challenging setting for the task: The verbs and arguments in each sentence are the main indicators for the classifiers to identify causativity. In D_{tr}, all models achieve slightly lower accuracy. Longer sentences are more likely to contain conjunctions or subordinate clauses, which may distract the classifiers from the sentence's (non)causative root verb and its arguments. As expected, the lowest accuracy scores are observed in D_{all}, which includes both transitive and intransitive sentences, as well as alternating verbs whose causativity property changes in these different environments. Table 3 shows that all three models mislabel alternating verbs more often than nonalternating verbs. BERT and XLNet



(a) D_{tr_5}– BERT (b) D_{tr}– BERT (c) D_{all}– BERT

(d) D_{tr_5}– XLNet (e) D_{tr}– XLNet (f) D_{all}– XLNet

(g) D_{tr_5}– DistilBERT (h) D_{tr}– DistilBERT (i) D_{all}– DistilBERT

Figure 1: Layer-wise results: X-axis = Layer number, Y-axis = Classification accuracy

achieved the best accuracy for causative verbs in almost all experiments, while DistilBERT often performed better on noncausative verbs.

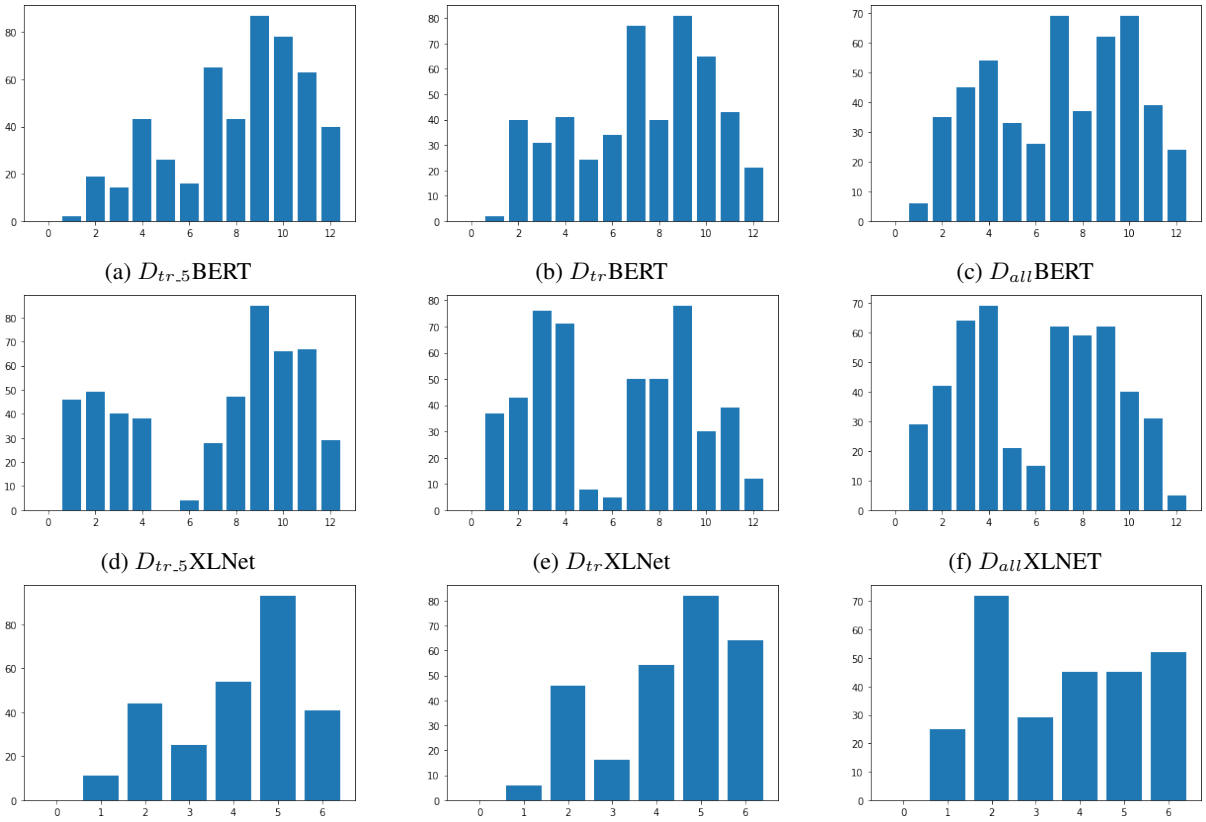
Our datasets are randomly collected from a larger corpus with no regard for verb frequency. This results in datasets where some verbs occur only once or twice, some are never seen in the training data, and some are more common. Our goal is to determine whether the classifiers successfully learn to predict (non)causativity, independently of specific verb lemmas. The results reported so far are all averaged over all verbs in a dataset, illustrating that some models are more successful on the classification task than others (e.g. BERT achieving higher accuracy scores than the other models on the first two datasets). Additionally, it is also worth exploring the accuracy of the classifiers for individual verbs, particularly those that are most likely to be mislabeled by any of the classifiers. Table 4 reports the two most-mislabeled verbs of each type per dataset (across all models). Notably, the XLnet classifier consistently makes more mistakes with noncausative instances than with causative ones, as is also apparent from Table 3.

Broadly, the frequently mislabeled verbs fall in three categories: 1. presumed errors due to parsing mistakes and subsequent errors in the gold data; 2. errors due to incorrect labels of ambiguous verbs in the gold data; 3. errors due to an ambiguity between full verb, light verb, and auxiliary verb.

Presumed errors due to parsing mistakes and subsequent errors in the gold data Most of the frequently-mislabeled verbs in D_{tr_5} fall into this category. These verbs occur only a few times each, indicating that they do not represent a deeper structural issue with the classifiers; for instance, sentences with the root verb *mark* occasionally appear incomplete in ENCOW, as exemplified in (6).

(6) the symptoms marked gr . (ENCOW-02-23709973)

The verb *sound* is labeled as a causative verb in our gold data (e.g., "to sound the bells"), but appears often in another word sense, as exemplified in (7-a). In these sentences, the verb does not have a direct object as expected; the reason for their inclusion in our datasets is an incorrect dependency parse in ENCOW. In other words, the causative



(a) D_{tr_5}BERT (b) D_{tr}BERT (c) D_{all}BERT

(d) D_{tr_5}XLNet (e) D_{tr}XLNet (f) D_{all}XLNET

(g) D_{tr_5}DistilBERT (h) D_{tr}DistilBERT (i) D_{all}DistilBERT

Figure 2: How top neurons spread across different layers for each causativity dataset. X-axis = Layer number, Y-axis = Number of neurons selected from that layer

gold label is assigned by mistake. A similar case is *mean*; as with *sound*, many instances do not involve a direct object at all, as exemplified in (7-b), but are included because of an incorrect parse.

(7) a. that sounds so scary !!! (ENCOW-05-11095175)
b. you mean screw justice ? (ENCOW-14-01839826)

D_{all} also contains incorrect gold labels that are to a large extent due to parsing errors, for instance *bring*. All sentences included in (8) were parsed as having *bring* as their root verb. That the classifiers tended to assign a noncausative label to these sentences suggests that they instead assigned labels for *take for granted*, *love*, or *be*, respectively (which is actually correct).

(8) a. people take for granted what tax money brings . (ENCOW-11-16881058)
b. knowledge is power , and what americans really love is the power knowledge brings . (ENCOW-13-11898010)
c. sugar is a barrow boy with all that epithet brings . (ENCOW-10-21805613)

In future work, we will improve our datasets to minimize the number of this type of errors, using a more recent dependency parser and some manual checking.

Errors due to incorrect gold labels of ambiguous verbs In D_{tr}, *face* is the most mislabeled causative verb. The presumed causative label for this verb comes from the VN class *confront-98*, which contains verbs such as *target* or *combat*. However, the mislabeled examples from the dataset seem to evoke a weaker, more passive sense of *face*, as in (9-a), where human annotators might not assign a causative label. In these cases, the label assigned by the classifier is actually correct, while the gold label is not. The mislabeled instances of *cover* in D_{all} are, similarly to *face*, an artefact of verb polysemy and should in fact not be regarded as causative sentences, as exemplified in (9-b).

(9) a. older mums face similar risks . (ENCOW-05-25724129)

	BERT	DistilBERT	XLNet
causative verbs in D_{tr_5}			
mark	6 (60.00%)	6 (60.00%)	2 (20.00%)
sound	1 (2.00%)	9 (18.00%)	4 (8.00%)
noncausative verbs in D_{tr_5}			
leave	4 (10.00%)	7 (17.50%)	15 (37.50%)
mean	1 (1.37%)	2 (2.74%)	15 (20.55%)
causative verbs in D_{tr}			
face	11 (25.00%)	11 (25.00%)	17 (38.64%)
express	12 (33.33%)	8 (22.22%)	10 (27.78%)
noncausative verbs in D_{tr}			
leave	7 (17.07%)	19 (46.34%)	17 (41.46%)
represent	8 (8.42%)	17 (17.89%)	15 (15.79%)
alternating causative verbs in D_{all}			
set	3 (8.82%)	10 (29.41%)	1 (2.94%)
open	3 (12.00%)	4 (16.00%)	3 (12.00%)
alternating noncausative verbs in D_{all}			
close	4 (57.14%)	5 (71.43%)	6 (85.71%)
open	4 (66.67%)	2 (33.33%)	5 (83.33%)
nonalternating causative verbs in D_{all}			
cover	9 (6.52%)	32 (23.19%)	10 (7.25%)
bring	9 (9.47%)	10 (10.53%)	9 (9.47%)
nonalternating noncausative verbs in D_{all}			
have	25 (4.64%)	19 (3.53%)	43 (7.98%)
be	20 (10.81%)	25 (13.51%)	36 (19.46%)

Table 4: Most mislabeled verbs in all settings. Each cell states the number of instances with the given verb with an incorrect label, giving the absolute number followed by the percentage of all instances with this verb.

b. the manual that comes with the game covers everything you need to know , including the mission editor . (ENCOW-08-06019647)

Sentences with the verb *represent* are frequently labeled as causative by one or more of the classifiers. When the verb is used in a legal or political sense, as in (10), this may in fact be appropriate. Since our verb sets are labeled on the lemma level and we do not perform any word sense disambiguation, these differences are not explicitly marked in our datasets, so these sentences are counted as mislabeled instances.

(10) they represent the voice of over 80,000 students and 62,000 members in 155 countries . (ENCOW-09-01862399)

In D_{tr}, all classifiers occasionally label instances of noncausative *leave* as causative, particularly XLNet. *leave* is a member of the VN classes *become-109.1-1-1*, *escape-51.1-1-1*, *fulfilling-13.4.1*, *future_having-13.3*, *keep-15.2*, and others. While not all of these classes license basic intransitive sentences of the type included in our datasets, this illustrates the polysemy of *leave*, which might be an explanation for the relatively high number of mislabeled instances in our experiments.

Generally, in D_{all}, noncausative alternating verbs are among the most mislabeled verbs. Since the dataset contains different numbers of verbs of each type, this may be a sparsity effect more than an effect of these verbs being more difficult to label. This question will be approached with new datasets in future work.

The reason for most errors of this type is that our datasets were created automatically with the help of a lexical resource. In order to avoid such polysemy issues, a version of the datasets with human annotations would be necessary.

Errors due to an ambiguity between full verb, light verb, and auxiliary verb Finally, the verbs *have* and *be* are the most mislabeled nonalternating noncausative verbs in D_{all}. These verbs appear in light verb constructions, as auxiliary verbs, and in a range of word senses that can be causative or noncausative. The examples in (11) illustrate why the classifiers are struggling to label such sentences as noncausative. Note that in all cases, the MaltParser annotations provided alongside ENCOW mark a form of *have* as the root verb.

(11) a. hi we have just moved house and the house has no tv aerial . (ENCOW-11-17855426)
b. we had a small cup made up not long ago with a very simple design . (ENCOW-06-00570494)
c. local people have the power to stop this by not buying counterfeit products . (ENCOW-08-19775040)

ENCOW was parsed between 2015 and 2018 using the standard *engmalt* model available on the MaltParser website (Roland Schäfer, p.c.) This type of error would be minimized if a more recent dependency parser was used.

To summarize, many of the "errors" of the classifiers are actually not errors but incorrect labels in the gold data. This means that the classifiers might be better in predicting causativity than assessed by our evaluation.

7 Conclusion

We set up a series of classification experiments with a range of datasets to determine whether large language models learn implicit representations of

causativity, a linguistic property that is not necessarily represented syntactically or morphologically in English. We compare classifiers based on BERT, DistilBERT, and XLNet, and find that all learn to predict causativity to a large extent. Differences in classification accuracy are observed across different datasets (see Table 1). As expected, all models achieve the highest accuracy on D_{tr_5} and the lowest accuracy on D_{all}. The latter set, in addition to verbs that are lexically causative or lexically non-causative, also includes verbs that participate in the causative-inchoative alternation, which presents an additional challenge to the classifiers.

We also show that causativity is represented rather in the higher layers of the models and, furthermore, that reducing each model to only the 10% of its neurons that are most correlated with the causativity property only leads to small differences in accuracy, sometimes an increase in accuracy due to the elimination of non-discriminative features.

Our error analysis suggests that many of the classification errors are actually labeling errors in the data, due either to a wrong parse of the sentence in our source corpus ENCOW or to the polysemy of verbs that can be causative in certain readings but are not causative in some of the readings mislabeled in the dataset. Put differently, the classifiers were probably better in identifying causativity than their accuracy scores suggest. While our datasets were created with little manual effort and already led to good results, we are planning on pursuing possible improvements in the future in order to avoid these labeling errors as far as possible.

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

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D. Anthony Bau, and James Glass. 2019a. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI, Oral presentation)*.

Fahim Dalvi, Avery Nortonsmith, D. Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, and James Glass. 2019b. Neurox: A toolkit for analyzing individual neurons in neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 9851–9852.

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.

David R. Dowty. 1979. *Word Meaning and Montague Grammar*. Springer Netherlands.

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.

Richard Futrell and Roger P. Levy. 2019. Do RNNs learn human-like abstract word order preferences? In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 50–59.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-Based Construction of a Verb Lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, page 691–696.

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 4364–4373, Hong Kong, China. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhuoran Liu, Shivali Goel, Mukund Yelahanka Raghuprasad, and Smaranda Muresan. 2019b. Columbia at SemEval-2019 task 7: Multi-task learning for stance classification and rumour verification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1110–1114, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119. Curran Associates, Inc.

Matthew 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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, Turkey. European Language Resources Association (ELRA).

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, Lancaster. UCREL, IDS.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. 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.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations.

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*, Florence, Italy.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

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 *The IEEE International Conference on Computer Vision (ICCV)*.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.

Monotonicity Marking from Universal Dependency Trees

Zeming Chen **Qiyue Gao**
Department of Computer Science and Software Engineering,
Rose-Hulman Institute of Technology
{chenz16, gaoq}@rose-hulman.edu

Abstract

Dependency parsing is a tool widely used in the field of Natural Language Processing and computational linguistics. However, there is hardly any work that connects dependency parsing to monotonicity, which is an essential part of logic and linguistic semantics. In this paper, we present a system that automatically annotates monotonicity information based on Universal Dependency parse trees. Our system utilizes surface-level monotonicity facts about quantifiers, lexical items, and token-level polarity information. We compared our system's performance with existing systems in the literature, including NatLog and ccg2mono, on a small evaluation dataset. Results show that our system outperforms NatLog and ccg2mono.

1 Introduction

The number of computational approaches for Natural Language Inference (NLI) has rapidly grown in recent years. Most of the approaches can be categorized as (1) Systems that translate sentences into first-order logic expressions and then apply theorem proving (Blackburn and Bos, 2005). (2) Systems that use blackbox neural network approaches to learn the inference (Devlin et al., 2019; Liu et al., 2019). (3) Systems that apply natural logic as a tool to make inferences (MacCartney and Manning, 2009; Hu et al., 2020; Angeli et al., 2016; Abzianidze, 2017). Compared to neural network approaches, systems that apply natural logic are more robust, formally more precise, and more explainable. Several systems contributed to the third category (MacCartney and Manning, 2009; Hu et al., 2020; Angeli et al., 2016) to solve the NLI task using monotonicity reasoning, a type of logical inference that is based on word replacement. Below is an example of monotonicity reasoning:

1. (a) **All** students↓ carry a MacBook↑.
   (b) All students carry a laptop.
   (c) All new students carry a MacBook.
2. (a) **Not all** new students↑ carry a laptop.
   (b) Not all students carry a laptop.

As the example shows, the word replacement is based on the polarity mark (arrow) on each word. A monotone polarity (↑) allows an inference from (1a) to (1b), where a more general concept *laptop* replaces the more specific concept *MacBook*. An antitone polarity (↓) allows an inference from (1a) to (1c), where a more specific concept *new students* replaces the more general concept *students*. The direction of the polarity marks can be reversed by adding a downward entailment operator like *Not* which allows an inference from (2a) to (2b). Thus, successful word placement relies on accurate polarity marks. To obtain the polarity mark for each word, an automatic polarity marking system is required to annotate a sentence by placing polarity mark on each word. This is formally called the polarization process. Polarity markings support monotonicity reasoning, and thus are used by systems for Natural Language Inference and data augmentations for language models. (MacCartney and Manning, 2009; Hu et al., 2020; Angeli et al., 2016).

In this paper, we introduce a novel automatic polarity marking system that annotates monotonicity information by applying a polarity algorithm on a universal dependency parse tree. Our system is inspired by ccg2mono, an automatic polarity marking system (Hu and Moss, 2018) used by Hu et al. (2020). In contrast to ccg2mono, which derives monotonicity information from CCG (Lewis and Steedman, 2014) parse trees, our system's polarization algorithm derives monotonicity information using Universal Dependency (Nivre et al., 2016) parse trees. There are several advantages of using UD parsing for polarity marking rather than

Proceedings of the 14th International Conference on Computational Semantics, pages 121–131
June 17–18, 2021.

CCG parsing. First, UD parsing is more accurate since the amount of training data for UD parsing is larger than those of CCG parsing. The high accuracy of UD parsing should lead to more accurate polarity annotation. Second, UD parsing works for more types of text. Overall, our system opens up a new framework for performing inference, semantics, and automated reasoning over UD representations. We will introduce the polarization algorithm's general steps, a set of rules we used to mark polarity on dependency parse trees, and comparisons between our system and some existing polarity marking tools, including NatLog (MacCartney and Manning, 2009; Angeli et al., 2016) and ccg2mono. Our evaluation focuses on a small dataset used to evaluate ccg2mono (Hu and Moss, 2020). Our system outperforms NatLog and ccg2mono. In particular, our system achieves the highest annotation accuracy on both the token level and the sentence level.

2 Related Work

Universal Dependencies (UD) (Nivre et al., 2016) was first designed to handle language tasks for many different languages. The syntactic annotation in UD mostly relies on dependency relations. Words enter into dependency relations, and that is what UD tries to capture. There are 40 grammatical dependency relations between words, such as nominal subject (**nsubj**), relative clause modifier (**acl:relcl**), and determiner (**det**). A dependency relation connects a headword to a modifier. For example, in the dependency parse tree for *All dogs eat food* (figure 1), the dependency relation **nsubj** connects the modifier *dogs* and the headword *eat*. The system presented in this paper utilizes Universal Dependencies to obtain a dependency parse tree from a sentence. We will explain the details of the parsing process in the implementation section.

There are two relevant systems of prior work: (1) The NatLog (MacCartney and Manning, 2009; Angeli et al., 2016) system included in the Stanford CoreNLP library (Manning et al., 2014); (2) The ccg2mono system (Hu and Moss, 2018). The NatLog system is a natural language inference system, a part of the Stanford CoreNLP Library. NatLog marks polarity to each sentence by applying a pattern-based polarization algorithm to the dependency parse tree generated by the Stanford dependency parser. A list of downward-monotone and non-monotone expressions are defined along

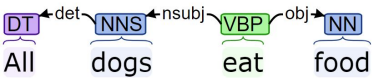


Figure 1: A dependency parse tree for "All dogs eat food."

with an arity and a Tregex pattern for the system to identify if an expression occurred.

The ccg2mono system is a polarity marking tool that annotates a sentence by polarizing a CCG parse tree. The polarization algorithm of ccg2mono is based on van Benthem (1986)'s work and Moss (2012)'s continuation on the soundness of internalized polarity marking. The system uses a marked/order-enriched lexicon and can handle application rules, type-raising, and composition in CCG. The main polarization contains two steps: mark and polarize. For the mark step, the system puts markings on each node in the parse tree from leaf to root. For the polarize step, the system generates polarities to each node from root to leaf. Compared to NatLog, an advantage of ccg2mono is that it polarizes on both the word-level and the constituent level.

3 Universal Dependency to Polarity

3.1 Overview

Our system's polarization algorithm contains three steps: (1) Universal Dependency Parsing, which transforms a sentence to a UD parse tree, (2) Binarization, which converts a UD parse tree to a binary UD parse tree, and (3) Polarization, which places polarity marks on each node in a binary UD parse tree.

3.2 Binarization

To preprocess the dependency parse graph, we designed a binarization algorithm that can map each dependency tree to an s-expression (Reddy et al., 2016). Formally, an s-expression has the form (exp1 exp2 exp3), where exp1 is a dependency label, and both exp2 and exp3 are either (1) a word such as *eat*; or (2) an s-expression such as (**det** *all dogs*). The process of mapping a dependency tree to an s-expression is called binarization. Our system represents an s-expression as a binary tree. A binary tree has a root node, a left child node, and a right child node. In representing an s-expression, the root node can either be a single word or a dependency label. Both the left and the right child nodes can either be a sub-binary-tree, or null. The

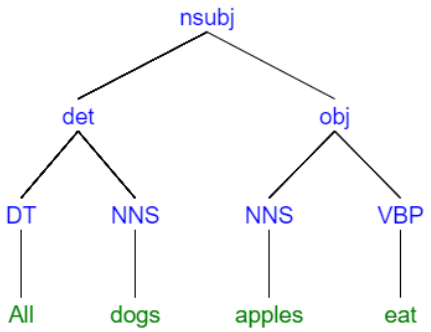


Figure 2: A binarized dependency parse tree for "All dogs eat apples."

system always puts the modifiers on the left and the headwords on the right. For example, the sentence *All dogs eat apples* has an s-expression

(**nsubj** (**det** *All dogs*) (**obj** *eat apples*))

and can be shown as a binary tree in figure 2. In the left sub-tree (*All dogs*), the dependency label **det** will be the root node, the modifier *all* will be the left child, and the headword *dogs* will be the right child.

Our binarization algorithm employs a dependency relation hierarchy to impose a strict traversal order from the root relation to each leaf word. The hierarchy allows for an ordering on the different modifier words. For example, in the binary dependency parse tree (**nsubj** (**det** All dogs) (**obj** eat apples)), the nominal subject (**nsubj**) goes above the determiner (**det**) in the tree because **det** is lower than **nsubj** in the hierarchy. We originally used the binarization hierarchy from Reddy et al. (2016)'s work, and later extended it with additional dependency relations such as oblique nominal (**obl**) and expletive (**expl**). Table 1 shows the complete hierarchy where the level-id indicates a relation's level in the hierarchy. The smaller a relation's level-id is, the higher that relation is in the hierarchy.

relation	level-id	relation	level-id
conj-sent	0	obl:tmod	50
advcl-sent	1	obl:npmod	50
advmod-sent	2	cop	50
case	10	det	55
mark	10	det:predet	55
expl	10	acl	60
discourse	10	acl:relcl	60
nsubj	20	appos	60
csubj	20	conj	60
nsubj:pass	20	conj-np	60
conj-vp	25	conj-adj	60
ccomp	30	obj	60
advcl	30	iobj	60
advmod	30	cc	70
nmod	30	amod	75
nmod:tmod	30	nummod	75
nmod:npmod	30	compound	80
nmod:poss	30	compound:prt	80
xcomp	40	fixed	80
aux	40	conj-n	90
aux:pass	40	conj-vb	90
obl	50	flat	100

Table 1: Universal Dependency relation hierarchy. The smaller a relation's level-id is, the higher that relation is in the hierarchy.

Algorithm 1 Binarization

```
root ← GET_ROOT_NODE(G)
T ← COMPOSE(root)
return T

function COMPOSE(node):
    C ← GET_CHILDREN(node)
    C_s ← SORT_BY_PRIORITY(C)
    if | C_s | == 0 then
        B ← BINARYDEPENDENCYTREE()
        B.val = node
        return B
    else
        top ← C.pop()
        B ← BINARYDEPENDENCYTREE()
        B.val = RELATE(top, node)
        B.left = COMPOSE(top)
        B.right = COMPOSE(node)
        return B
    end if
end function
```

3.3 Polarization

The polarization algorithm places polarities on each node of a UD parse tree based on a lexicon of polarization rules for each dependency relation and some special words. Our polarization algorithm is similar to the algorithms surveyed by Lavalle-Martínez et al. (2018). Like the algorithm of Sanchez (1991), our algorithm computes polarity from leaves to root. One difference our algorithm has is that often, the algorithm computes polarity following a left-to-right inorder traversal (left⟶root⟶right) or a right-to-left inorder traversal (right⟶root⟶left) in additional to the top-down traversal. In our algorithm, each node's polarity depends both on its parent node and its sibling node (left side or right side), which is different from algorithms in Lavalle-Martínez et al. (2018)'s paper. Our algorithm is deterministic, and thus never fails.

The polarization algorithm takes in a binarized UD parse tree $\mathcal{T}$ and a set of polarization rules, both dependency-relation-level ($\mathcal{L}$) and word-level ($\mathcal{W}$). The algorithm outputs a polarized UD parse tree $\mathcal{T}^*$ such that (1) each node is marked with

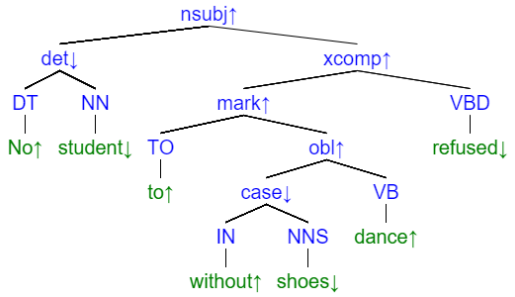


Figure 3: Visualization of a polarized binary dependency parse tree for a triple negation sentence *No student refused to dance without shoes.*

a polarity of either monotone (↑), antitone (↓), or no monotonicity information (=), (2) both $\mathcal{T}$ and $\mathcal{T}^*$ have the same universal dependency structure except the polarity marks. Figure 3 shows a visualization of the binary dependency parse tree after polarization completes. The general steps of the polarization start from the root node of the binary parse tree. The system will get the corresponding polarization rule from the lexicon according to the root node's dependency relation. In each polarization rule, the system applies the polarization rule and then continues the above steps recursively down the left sub-tree and the right sub-tree. Each polarization rule is composed from a set of basic building blocks include rules for negation, equalization, and monotonicity generation. When the recursion reaches a leaf node, which is an individual word in a sentence, a set of word-based polarization rules will be retrieved from the lexicon, and the system polarizes the nodes according to the rule corresponding to a particular word. More details about word-based polarization rules will be covered in section 3.4.2, Polarity Generation. An overview of the polarization algorithm and a general scheme of the implementation for dependency-level polarization rules are shown in Algorithm 2.

3.4 Polarization Rules

Our polarization algorithm contains a lexicon of polarization rules corresponding to each dependency relation. Each polarization rule is composed from a set of building blocks divided into three categories: negation rules, equalization rules, and monotonicity generation rules. The generation rules will generate three types of monotonicity: monotone (↑), antitone (↓), and no monotonicity information (=) either by initialization or based on the words.

Algorithm 2 Polarization

Input: $\mathcal{T}$: binary dependency tree
$\mathcal{L}$: dependency-level polarization rules
$\mathcal{W}$: word-level polarization rules
Output: $\mathcal{T}^*$: polarized binary dependency tree

if $\mathcal{T}$.is_tree **then**
  relation $\leftarrow \mathcal{T}$.val
  POLARIZATION_RULE(.) $\leftarrow \mathcal{L}$[relation]
  POLARIZATION_RULE($\mathcal{T}$)
end if

▷ *General scheme of a polarization rule's implementation for a dependency relation*
function POLARIZATION_RULE($\mathcal{T}$)
  ▷ *Initialize or inherit polarities*
  if $\mathcal{T}$.mark $\neq$ NULL **then**
    $\mathcal{T}$.right.mark = $\mathcal{T}$.mark
    $\mathcal{T}$.left.mark = $\mathcal{T}$.mark
  else
    $\mathcal{T}$.right.mark = ↑
    $\mathcal{T}$.left.mark = ↑
  end if

  ▷ *Polarize sub-trees*
  POLARIZATION($\mathcal{T}$.left)
  POLARIZATION($\mathcal{T}$.right)
  ▷ *Or, for relations like* ***nsubj****:*
  ▷ POLARIZATION*($\mathcal{T}$.right)*
  ▷ POLARIZATION*($\mathcal{T}$.left)*

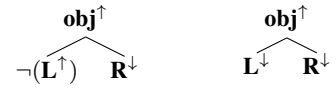
  ▷ *Apply* `negation` *and* `equalization` *rules*
  if NEGATE is applicable **then**
    NEGATE($\mathcal{T}$)
  end if
  if EQUALIZE is applicable **then**
    EQUALIZE($\mathcal{T}$)
  end if

  ▷ *Apply word-level rules*
  if not $\mathcal{T}$.is_tree and $\mathcal{T}$.val $\in \mathcal{W}$.keys **then**
    WORD_RULE(.) $\leftarrow \mathcal{W}$[$\mathcal{T}$.val]
    WORD_RULE($\mathcal{T}$)
  end if
end function

3.4.1 Building Blocks

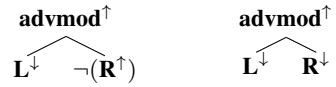
Negation and Equalization The negation rule and the equalization rule are used by several core dependency relations such as **nmod**, **obj**, and **acl:recl**. Both negation and equalization have two ways of application: backward or top-down. A backward negation rule is triggered by a downward polarity (↓) on the right node of the tree (marked below as R), flipping every node's polarity under the left node (marked below as L). Similarly, a backward equalization rule is triggered by a no monotonicity information polarity (=) on the tree's right node, and it marks every node under the left node as =. Examples for trees before and after applying a backward and forward negation and equalization are shown as follows:

- Backward Negation:

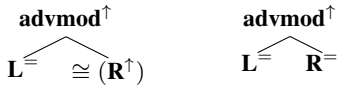


- Backward Equalization:

- Forward Negation:

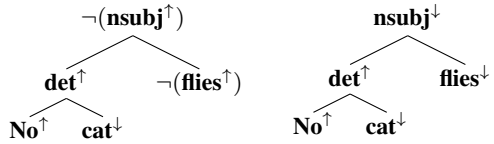


- Forward Equalization:



where ¬ means negation and ≅ means equalization.

A top-down negation is used by the polarization rule like determiner (**det**) and adverbial modifier (**advmod**). It starts at the parent node of the current tree, and flips the arrow on each node under that parent node excluding the current tree. This top-down negation is used by **det**, **case**, and **advmod** when a negation operators like *no*, *not*, or *at-most* appears. Below is an example of a tree before and after applying the top-down negation:



Polarity Generation The polarity is generated by words. During the polarization, the polarity can change based on a particular word that can promote the polarity governing the part of the sentence to which it belongs. These words include quantifiers and verbs. For the monotonicity from quantifiers, we follow the monotonicity profiles listed in the work done by Icard III and Moss (2014) on monotonicity, which built on van Benthem (1986). Additionally, to extend to more quantifiers, we observed polarization results generated by ccg2mono. Overall, we categorized the quantifiers as follows:

- Universal Type

  Every ↓ ↑ Each ↓ ↑ All ↓ ↑

- Negation Type

  No ↓ ↓ Less than ↓ ↓ At most ↓ ↓

- Exact Type

  Exactly n = = The = ↑ This = ↑

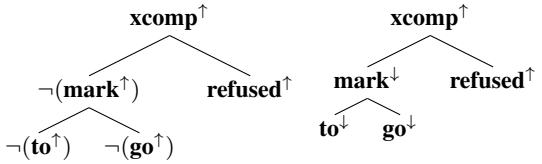
- Existential Type

  Some ↑ ↑ Several ↑ ↑ A, An ↑ ↑

- Other Type

  Most = ↑ Few = ↓

Where the first mark is the monotonicity for the first argument after the quantifier and the second mark is the monotonicity for the second argument after the quantifier. For verbs, there are upward entailment operators and downward entailment operators. Verbs that are downward entailment operators, such as *refuse*, promote an antitone polarity, which will negate its dependents. For example, for the phrase *refused to go*, *refused* will promote an antitone polarity, which negates *to dance*:



In addition to quantifiers and verbs, some other words also change the monotonicity of a sentence. For example, words like *not*, *none*, and *nobody* promote an antitone polarity. Our system also handles material implications with the form *if x then y*. Based on Moss (2012), the word *if* promotes an antitone polarity in the antecedent and positive polarity in the consequent. For background on monotonicity and semantics, see van Benthem (1986), Keenan and Faltz (1984), and also Karttunen (2012).

3.4.2 Dependency Relation Rules

Each dependency relation has a corresponding polarization rule. All the rules start with initializing the starting node as upward monotone polarity (↑). Alternatively, if the starting node has a polarity marked, each child node will inherit the root node's polarity. Each rule's core part is a combination of the default rules and monotonicity generation rules. In this section, we will briefly show three major types of dependency relation rules in the polarization algorithm. The relative clause modifier relation will represent rules for modifier relations. The determiner relation rule will represent rules containing monotonicity generation rules. The Object and open clausal complement rule will represent rules containing word-level polarization rules.

Algorithm 3 Polarize_acl:relcl

Input: $\mathcal{T}$: binary dependency sub-tree
Output: $\mathcal{T}^*$: polarized binary dependency sub-tree

```
if T.mark ≠ NULL then
    T.right.mark = T.mark
else
    T.right.mark = ↑
end if
T.left.mark = ↑

POLARIZE(T.right)
POLARIZE(T.left)

if T.right.mark == ↓ then
    NEGATE(T.left)
else if T.right.mark == = then
    EQUALIZE(T.left)
end if
```

Relative Clause Modifier For the relative clause modifier relation (**acl:relcl**), the relative clause depends on the noun it modifies. First, the polarization will first be performed on both the left and right nodes, and then, depending on the polarity of the right node, a negation or an equalization rule will be applied. The algorithm first applies a top-down inheritance if the root already has its polarity marked; otherwise, it initializes the left and right nodes as monotone. The algorithm polarizes both the left and right nodes. Next, the algorithm checks the right node's polarity. If the right node is marked as antitone, a backward negation is applied. Alternatively, if the right node is marked as no monotonicity information, a backward equalization is applied. During the experiments, we noticed that if the root node is marked antitone, and the left node inherits that, a negation later will cause a double negation, producing incorrect polarity marks. To avoid this double negation, we exclude the left node from the top-down inheritance rule by initializing the left node directly with a monotone mark. The rule for **acl:relcl** also applies to the adverbial clause modifier (**advcl**) and the clausal modifier of noun (**acl**). An overview of the algorithm is shown in Algorithm 3.

Determiner For the determiner relation (**det**), each different determiner can assign a new monotonicity to the noun it modifies. First, the algorithm performs a top-down inheritance on the left node if the root already has polarity marked. Next, the algorithm assigns the polarity for the noun depending on the determiner's type. For example, if the determiner is a universal quantifier, an antitone polarity is assigned to the right node. For negation quantifiers like *no*, its right node also receives an antitone polarity. Thus, a top-down negation is applied at the determiner relation tree's parent. Algorithm 4 shows an overview of the algorithm.

Algorithm 4 Polarize_det

Input: $\mathcal{T}$: binary dependency sub-tree
$\mathcal{D}$: determiner mark dictionary
Output: $\mathcal{T}^*$: polarized binary dependency sub-tree

```
det_type ← GET_DET_TYPE(T.left)
if T.mark ≠ NULL then
    T.left.mark = T.mark
else
    T.left.mark = ↑
end if

T.right.mark = D[det_type]
POLARIZE(T.right)

if det_type == negation then
    NEGATE(T.parent)
end if
```

Object and Open Clausal Complement For the object relation (**obj**) and the open clausal complement relation **xcomp**, both the verb and the noun would inherit the monotonicity from the parent in the majority of cases. The inheritance procedure is the same as the one used in **acl:relcl**'s rule. Similarly, after the inheritance, the rule will polarize both the right sub-tree and the left sub-tree. Differently, since **obj** and **xcomp** both have a verb under the relation, they require a word-level polarization rule that will check the verb determine if the verb is a downward entailment operator, which prompts an antitone monotonicity. The algorithm takes in a dictionary that contains a list of verbs and their

Algorithm 5 Polarize_obj

Input: $\mathcal{T}$: binary dependency sub-tree
Output: $\mathcal{T}^*$: polarized binary dependency sub-tree

```
if T.mark ≠ NULL then
    T.right.mark = T.mark
else
    T.right.mark = ↑
end if
T.left.mark = ↑

POLARIZE(T.right)
POLARIZE(T.left)

▷ Word-level polarization rule for downward entailment operators
if IS_DOWNWARD_OPERATOR(T.right.mark) then
    NEGATE(T.left)
end if
```

implicatives. The dictionary is generated from the implicative verb dataset made by Ross and Pavlick (2019). If a verb is a downward entailment operator, which has a negative implicative, the rule will apply a negation rule on the left sub-tree to flip each node's arrow in the left sub-tree. An overview of the algorithm is shown in Algorithm 5.

4 Comparison to Existing Systems

We conducted several preliminary comparisons to two existing systems. First, we compared to NatLog's monotonicity annotator. Natlog's annotator also uses dependency parsing. The polarization algorithm does pattern-based matching for finding occurrences of downward monotonicity information, and the algorithm only polarizes on word-level. In contrast, our system uses a tree-based polarization algorithm that polarizes both on word-level polarities and constituent level polarities. Our intuition is that the Tregex patterns used in NatLog is not as common or as easily understandable as the binary tree structure, which is a classic data structure wildly used in the filed of computer science.

According to the comparison on a list of sentences, NatLog's annotator does not perform as well as our system. For example, for a phrase *the rabbit*, *rabbit* should have a polarity with no monotonicity information (=). However, NatLog marks *rabbit* as a monotone polarity (↑). NatLog also incorrectly polarizes sentences containing multiple negations. For example, for a triple negation sentence, *No newspapers did not report no bad news*, NatLog gives: $No^{\uparrow}$ $newspapers^{\downarrow}$ $did^{\downarrow}$ $not^{\downarrow}$ $report^{\uparrow}$ $no^{\uparrow}$ $bad^{\uparrow}$ $news^{\uparrow}$. This result has incorrect polarity marks on multiple words, where *report*, *bad*, *news* should be ↓, and *no* should be ↑. Both of the scenarios above can be handled correctly by our system.

Comparing to ccg2mono, our algorithm shares some similarities to its polarization algorithm. Both of the systems polarize on a tree structure and rely on a lexicon of rules, and they both polarize on the word-level and the constituent level. One difference is that ccg2mono's algorithm contains two steps, the first step puts markings on each node, and the second step puts polarities on each node. Our system does not require the step of adding markings and only contains the step of adding polarities on each node.

Our system has multiple advantages over ccg2mono. For parsing, our system uses UD parsing, which is more accurate than CCG parsing used by ccg2mono due to a large amount of training data. Also, our system covers more types of text than ccg2mono because UD parsing works for a variety of text genres such as web texts, emails, reviews, and even informal texts like Twitter tweets. (Silveira et al., 2014; Zeldes, 2017; Liu et al., 2018). Our system can also work for more languages than ccg2mono since UD parsing supports more languages than CCG parsing.

Overall, our system delivers more accurate polarization than ccg2mono. Many times the CCG parser makes mistakes and leads to polarization mistakes later on. For example, in the annotation $The^{\downarrow}$ $market^{\downarrow}$ $is^{\downarrow}$ $not^{\downarrow}$ $impossible^{\downarrow}$ $to^{\downarrow}$ $navigate^{\downarrow}$, ccg2mono incorrectly marks every word as ↓. Our system, on the other hand, uses UD parsing which has higher parsing accuracy than CCG parsing, and thus leads to fewer polarization mistakes compared to ccg2mono. For the expression above, our system correctly polarizes it as $The^{\uparrow}$ $market^{=}$ $is^{\uparrow}$ $not^{\uparrow}$ $impossible^{\downarrow}$ $to^{\uparrow}$ $navigate^{\uparrow}$.

Our system also handles multi-word quantifiers better than ccg2mono. For example, for a multi-word quantifier expression like *all of the dogs*, ccg2mono mistakenly marks *dogs* as =. Our system, however, can correctly mark the expression: $all^{\uparrow}$ $of^{\uparrow}$ $the^{\uparrow}$ $dogs^{\downarrow}$.

Moreover, the core of ccg2mono does not include aspects of verbal semantics of downward-entailing operators like *forgot* and *regret* (Moss and Hu, 2020). For example ccg2mono's polarization for $Every^{\uparrow}$ $member^{\downarrow}$ $forgot^{\uparrow}$ $to^{\uparrow}$ $attend^{\uparrow}$ $the^{\uparrow}$ $meeting^{=}$ is not correct because it fails to flip the polarity of *to attend the*. In contrast, our system produces a correct result: $Every^{\uparrow}$ $member^{\downarrow}$ $forgot^{\uparrow}$ $to^{\downarrow}$ $attend^{\downarrow}$ $the^{\downarrow}$ $meeting^{=}$.

All three systems have difficulty polarizing sentences containing numbers. A scalar number **n**'s monotonicity information is hard to determine because it can presenter different contexts: a single number **n**, without additional quantifiers or adjectives, can either mean *at least* **n**, *at most* **n**, *exactly* **n**, and *around* **n**. These contexts are syntactically hard to identify for a dependency parser or a CCG parser because it would require pragmatics and some background knowledge which the parsers do not have. For example, in the sentence *A dog ate 2 rotten biscuits*, the gold label for *2* is = which indicates that the context is "exactly 2". However, our system marks this as "↓ since it considers the

sentence	type
More$^{\uparrow}$ dogs$^{\uparrow}$ than$^{\uparrow}$ cats$^{\downarrow}$ sit$^{=}$	comparative
Less$^{\uparrow}$ than$^{\uparrow}$ 5$^{\uparrow}$ people$^{\downarrow}$ ran$^{\downarrow}$	less-than
A$^{\uparrow}$ dog$^{\uparrow}$ who$^{\uparrow}$ ate$^{\uparrow}$ two$^{=}$ rotten$^{\uparrow}$ biscuits$^{\uparrow}$ was$^{\uparrow}$ sick$^{\uparrow}$ for$^{\uparrow}$ three$^{\downarrow}$ days$^{\downarrow}$	number
Every$^{\uparrow}$ dog$^{\downarrow}$ who$^{\downarrow}$ likes$^{\downarrow}$ most$^{\downarrow}$ cats$^{=}$ was$^{\uparrow}$ chased$^{\uparrow}$ by$^{\uparrow}$ at$^{\uparrow}$ least$^{\uparrow}$ two$^{\downarrow}$ of$^{\uparrow}$ them$^{\uparrow}$	every:most:at-least
Even$^{\uparrow}$ if$^{\uparrow}$ you$^{\downarrow}$ are$^{\downarrow}$ addicted$^{\downarrow}$ to$^{\downarrow}$ cigarettes$^{\downarrow}$ you$^{\uparrow}$ can$^{\uparrow}$ smoke$^{\uparrow}$two$^{\downarrow}$ a$^{\uparrow}$ day$^{\uparrow}$	conditional:number

Table 2: Example sentences in Hu and Moss (2020)'s evaluation dataset

context as "at least 2", which is different from the gold label.

5 Experiment

Dataset We obtained the small evaluation dataset used in the evaluation of ccg2mono (Hu and Moss, 2020) from its authors. The dataset contains 56 hand-crafted English sentences, each with manually annotated monotonicity information. The sentences cover a wide range of linguistic phenomena such as quantifiers, conditionals, conjunctions, and disjunctions. The dataset also contains hard sentences involving scalar numbers. Some example sentences from the dataset are shown in Table 2.

Dependency Parser In order to obtain a universal dependency parse tree from a sentence, we utilize a parser from Stanza (Qi et al., 2020), a Python natural language analysis package made by Stanford. The neural pipeline in Stanza allow us to use pretrained neural parsing models to generate universal dependency parse trees. To achieve optimal performance, we trained two neural parsing models: one parsing model trained on Universal Dependency English GUM corpus (Zeldes, 2017). The pretrained parsing model achieved 90.0 LAS (Zeman et al., 2018) evaluation score on the testing data.

Experiment Setup We evaluated the polarization accuracy on both the token level and the sentence level, in a similar fashion to the evaluation for part-of-speech tagging (Manning, 2011). For both levels of accuracy, we conducted one evaluation on all tokens (*acc(all-tokens)* in Table 3) and another one on key tokens including content words (nouns, verbs, adjectives, adverbs), determiners, and numbers (*acc(key-tokens)* in Table 3). The key tokens contain most of the useful monotonicity information for inference. In token-level evaluation, we counted the number of correctly annotated tokens for *acc(all-tokens)* or the number of correctly annotated key tokens for *acc(key-tokens)*. In sentence-level evaluation, we counted the number of correct sentences. A correct sentence has all tokens correctly annotated for *acc(all-tokens)* or all key tokens correctly annotated for *acc(key-tokens)*. We also evaluated our system's robustness on the token level. We followed the robustness metric for evaluating multi-class classification tasks, which uses precision, recall, and F1 score to measure a system's robustness. We calculated these three metrics for each polarity label: monotone(↑), antitone(↓), and None or no monotonicity information(=). The robustness evaluation is also done both on all tokens and on key tokens.

	Token-level		
system	NatLog	ccg2mono	ours
acc(all-tokens)	69.9	76.0	**96.5**
acc(key-tokens)	68.1	78.0	**96.5**
	Sentence-level		
system	NatLog	ccg2mono	ours
acc(all-tokens)	28.0	44.6	**87.5**
acc(key-tokens)	28.6	50.0	**89.2**

Table 3: This table shows the polarity annotation accuracy on the token level and the sentence level for three systems: NatLog, ccg2mono, and our system. The token level accuracy counts the number of correctly annotated tokens, and the sentence level accuracy counts the number of correctly annotated sentences. Two types of accuracy are used. For *acc(all-tokens)*, all tokens are evaluated. For *acc(key-tokens)*, only key tokens (content words + determiners + numbers) are evaluated.

6 Evaluation

Table 3 shows the performance of our system, compared with NatLog and ccg2mono. Our evaluation process is the same as Hu and Moss (2020). From Table 3, we first observe that our system consistently outperforms ccg2mono and NatLog on both the token level and the sentence level. For accuracy on the token level, our system has the highest accuracy for the evaluation on all tokens (96.5) and the highest accuracy for the evaluation on key tokens (96.5). Our system's accuracy on key tokens is higher than the accuracy on all tokens, which demonstrates our system's good performance on polarity annotation for tokens that are more signif-

	All Tokens								
system	**NatLog**			**ccg2mono**			**ours**		
Polarity	Monotone	Antitone	None	Monotone	Antitone	None	Monotone	Antitone	None
precision	71.4	43.5	70.7	86.0	75.6	58.0	**97.6**	**96.5**	**91.7**
recall	87.3	15.9	63.9	77.8	78.3	74.6	**97.2**	**89.4**	**87.3**
F1-score	78.6	23.3	67.1	81.7	76.9	65.3	**97.4**	**97.6**	**89.4**
	Key Tokens								
system	**NatLog**			**ccg2mono**			**ours**		
Polarity	Monotone	Antitone	None	Monotone	Antitone	None	Monotone	Antitone	None
precision	68.7	70.9	42.1	85.2	78.7	62.7	**96.9**	**96.4**	**94.2**
recall	88.6	61.5	14.0	80.3	79.3	73.7	**97.9**	**98.5**	**86.0**
F1-score	77.4	65.9	21.1	82.7	79.0	67.7	**97.4**	**97.4**	**89.9**

Table 4: Token level robustness comparison between NatLog, ccg2mono, and our system. The robustness score is evaluated both on all tokens and on key tokens (content words + determiners + numbers). For each of the three polarities: monotone(↑), antitone(↓), and None or no monotonicity information(=), the relative precision, recall and F1 score are calculated.

icant to monotonicity inference. For accuracy on the sentence level, our system again has the highest accuracy for the evaluation on all tokens (87.5) and the highest accuracy for the evaluation on key tokens (89.2). Such results suggest that our system can achieve good performance on determining the monotonicity of the sentence constituents. Overall, the evaluation validates that our system has higher polarity annotation accuracy than existing systems. We compared our annotations to ccg2mono's annotation and observed that of all the tokens in the 56 sentences, if ccg2mono annotates it correctly, then our system also does so. This means, our system's polarization covers more linguistic phenomena than ccg2mono. Table 4 shows the robustness score of our system and the two existing systems. Our systems has much higher precision and recall on all three polarity labels than the other two systems. For the F1 score, our system again has the highest points over the other two systems. The consistent and high robustness scores show that our system's performance is much more robust on the given dataset than existing systems.

7 Conclusion and Future Work

In this paper, we have demonstrated our system's ability to automatically annotate monotonicity information (polarity) for a sentence by conducting polarization on a universal dependency parse tree. The system operates by first converting the parse tree to a binary parse tree and then marking polarity on each node according to a lexicon of polarization rules. The system produces accurate annotations on sentences involving many different linguistic phenomena such as quantifiers, double negation, relative clauses, and conditionals. Our system had better performance on polarity marking than existing systems including ccg2mono (Hu and Moss, 2018) and NatLog (MacCartney and Manning, 2009; Angeli et al., 2016). Additionally, by using UD parsing, our system offers many advantages. Our system supports a variety of text genres and can be applied to many languages. In general, this paper opens up a new framework for performing inference, semantics, and automated reasoning over UD representations.

For future work, an inference system can be made that utilizes the monotonicity information annotated by our system, which is similar to the MonaLog system (Hu et al., 2020). Several improvements can be made to the system to obtain more accurate annotations. One improvement would be to incorporate pragmatics to help determine the monotonicity of a scalar number.

Acknowledgements

This research is advised by Dr. Lawrence Moss from Indiana University and Dr. Michael Wollowski from Rose-hulman Institute of Technology. We thank their helpful advises and feedback on this research. We also thank the anonymous reviewers for their insightful comments.

References

Lasha Abzianidze. 2017. LangPro: Natural language theorem prover. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 115–120, Copenhagen, Denmark. Association for Computational Linguistics.

Gabor Angeli, Neha Nayak, and Christopher D. Manning. 2016. Combining natural logic and shallow

reasoning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 442–452, Berlin, Germany. Association for Computational Linguistics.

Johan van Benthem. 1986. *Essays in Logical Semantics*, volume 29 of *Studies in Linguistics and Philosophy*. D. Reidel Publishing Co., Dordrecht.

P. Blackburn and Johan Bos. 2005. Representation and inference for natural language - a first course in computational semantics. In *CSLI Studies in 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.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S Moss, and Sandra Kübler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*, pages 319–329.

Hai Hu and Larry Moss. 2018. Polarity computations in flexible categorial grammar. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 124–129, New Orleans, Louisiana. Association for Computational Linguistics.

Hai Hu and Lawrence S. Moss. 2020. An automatic monotonicity annotation tool based on ccg trees. In *Second Tsinghua Interdisciplinary Workshop on Logic, Language, and Meaning: Monotonicity in Logic and Language*.

Thomas F. Icard III and Lawrence S. Moss. 2014. Recent progress on monotonicity. In *Linguistic Issues in Language Technology, Volume 9, 2014 - Perspectives on Semantic Representations for Textual Inference*. CSLI Publications.

Lauri Karttunen. 2012. Simple and phrasal implicatives. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, page 124–131, USA. Association for Computational Linguistics.

Edward L. Keenan and Leonard M. Faltz. 1984. *Boolean Semantics for Natural Language*. Springer.

J. Lavalle-Martínez, M. Montes y Gómez, L. Pineda, Héctor Jiménez-Salazar, and Ismael Everardo Bárcenas Patiño. 2018. Equivalences among polarity algorithms. *Studia Logica*, 106:371–395.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar. Association for Computational Linguistics.

Wei Liu, Lei Li, Zuying Huang, and Yinan Liu. 2019. Multi-lingual Wikipedia summarization and title generation on low resource corpus. In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, pages 17–25, Varna, Bulgaria. INCOMA Ltd.

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. 2018. Parsing tweets into Universal Dependencies. 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 965–975, New Orleans, Louisiana. Association for Computational Linguistics.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Christopher D. Manning. 2011. Part-of-speech tagging from 97linguistics? In *CICLing*.

L. Moss. 2012. The soundness of internalized polarity marking. *Studia Logica*, 100:683–704.

Lawrence S. Moss and Hai Hu. 2020. Syllogistic logics with comparative adjectives. Unpublished ms., Indiana University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational*

Linguistics: System Demonstrations, pages 101–108, Online. Association for Computational Linguistics.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Alexis Ross and Ellie Pavlick. 2019. How well do NLI models capture verb veridicality? 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 2230–2240, Hong Kong, China. Association for Computational Linguistics.

V. Sanchez. 1991. Studies on natural logic and categorial grammar.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Is that really a question? Going beyond factoid questions in NLP

Aikaterini-Lida Kalouli and **Rebecca Kehlbeck** and **Rita Sevastjanova** and
Oliver Deussen and **Daniel Keim** and **Miriam Butt**
University of Konstanz
firstname.lastname@uni-konstanz.de

Abstract

Research in NLP has mainly focused on factoid questions, with the goal of finding quick and reliable ways of matching a query to an answer. However, human discourse involves more than that: it contains non-canonical questions deployed to achieve specific communicative goals. In this paper, we investigate this under-studied aspect of NLP by introducing a targeted task, creating an appropriate corpus for the task and providing baseline models of diverse nature. With this, we are also able to generate useful insights on the task and open the way for future research in this direction.

1 Introduction

Recently, the field of human-machine interaction has seen ground-breaking progress, with the tasks of Question-Answering (QA) and Dialog achieving even human-like performance. The probably most popular example is *Watson* (Ferrucci et al., 2013), IBM's QA system which was able to compete on the US TV program *Jeopardy!* and beat the best players of the show. Since then and particularly with the rise of Neural Networks (NN), various high-performance QA and Dialog systems have emerged. For example, on the QQP task of the GLUE benchmark (Wang et al., 2018), the currently best performing system achieves an accuracy of 90.8%. Despite this success, current QA and Dialog systems cannot be claimed to be on a par with human communication. In this paper we address one core aspect of human discourse that is underresearched within NLP: non-canonical questions.

Research in NLP has mainly focused on factoid questions, e.g., *When was Mozart born?*, with the goal of finding quick and reliable ways of matching a query to terms found in a given text collection. There has been less focus on understanding the structure of questions per se and the communicative goal they aim to achieve. State-of-the-art parsers are mainly trained on Wikipedia entries or newspaper texts, e.g., the Wall Street Journal, genres which do not contain many questions. Thus, the tools trained on them are not effective in dealing with questions, let alone distinguishing between different types. Even within more computational settings that include deep linguistic knowledge, e.g., PARC's Bridge QA system (Bobrow et al., 2007) which uses a sophisticated LFG parser and semantic analysis, the actual nature and structure of different types of questions is not studied in detail.

However, if we are aiming at human-like NLP systems, it is essential to be able to efficiently deal with the fine nuances of non-factoid questions (Dayal, 2016). Questions might be posed

- as a (sarcastic, playful) comment, e.g., *Have you ever cooked an egg?* (rhetorical)
- to repeat what was said or to express incredulity/surprise, e.g., *He went where?* (echo)
- to make a decision, e.g., *What shall we have for dinner?* (deliberative)
- to deliberate rather than ask or to rather ask oneself than others, e.g., *Do I even want to go out?* (self-addressed)
- to request or order something, e.g., *Can you pass me the salt?* (ability/inclination)
- to suggest that a certain answer should be given in reply, e.g., *Don't you think that calling names is wrong?* (suggestive)
- to assert something, e.g., *You are coming, aren't you?* (tag)
- to quote the words of somebody else, e.g., *And he said, "Why do you bother?"* (quoted)
- to structure the discourse, e.g., *What has this taught us? It ...* (discourse-structuring)
- etc.

Proceedings of the 14th International Conference on Computational Semantics, pages 132–143
June 17–18, 2021.

The importance of these communicative goals in everyday discourse can be seen in systems like personal assistants, chatbots and social media. For example, personal assistants like Siri, Alexa and Google should be able to distinguish an ability question of the kind *Can you play XYZ?* from a rhetorical question such as *Can you be even more stupid?* Similarly, chatbots offering psychotherapeutic help (Ly et al., 2017; Håvik et al., 2019) should be able to differentiate between a factoid question such as *Is this a symptom for my condition?* and a self-addressed question, e.g., *Why can't I do anything right?* In social media platforms like Twitter, apart from the canonical questions of the type *Do you know how to tell if a brachiopod is alive?*, we also find non-canonical ones like *why am I lucky?* Paul et al. (2011) show that 42% of all questions on English Twitter are rhetorical.

To enable NLP systems to capture non-factoid uses of questions, we propose the task of *Question-Type Identification* (QTI). The task can be defined as follows: given a question, determine whether it is an information-seeking question (ISQ) or a non information-seeking question (NISQ). The former type of question, also known as a canonical or factoid question, is posed to elicit information, e.g., *What will the weather be like tomorrow?* In contrast, questions that achieve other communicative goals are considered non-canonical, non-information-seeking. NISQs do not constitute a homogeneous class, but are heterogeneous, comprising sub-types that are sometimes difficult to keep apart (Dayal, 2016). But even at the coarse-grained level of distinguishing ISQs from NISQs, the task is difficult: surface forms and structural cues are not particularly helpful; instead, Bartels (1999) and Dayal (2016) find that prosody and context are key factors in question classification.

Our ultimate objective in this paper is to provide an empirical evaluation of learning-centered approaches to QTI, setting baselines for the task and proposing it as a tool for the evaluation of QA and Dialog systems. However, to the best of our knowledge, there are currently no openly available QTI corpora that can permit such an assessment. The little previous research on the task has not contributed suitable corpora, leading to comparability issues. To address this, this paper introduces RQueT (*rocket*), the Resource of Question Types, a collection of questions in-the-wild labeled for their ISQ-NISQ type. As the first of its kind, the resource of 2000 annotated questions allows for initial machine-/deep-learning experimentation and opens the way for more research in this direction.

In this paper, we use this corpus to evaluate a variety of models in a wide range of settings, including simple linear classifiers, language models and other neural network architectures. We find that simple linear classifiers can compete with state-of-the-art transformer models like BERT (Devlin et al., 2019), while a neural network model, combining features from BERT and the simple classifiers, can outperform the rest of the settings.

Our contributions in this paper are three-fold. First, we provide the first openly-available QTI corpus, aiming at introducing the task and comprising an initial benchmark. Second, we establish suitable baselines for QTI, comparing systems of very different nature. Finally, we generate linguistic insights on the task and set the scene for future research in this area.

2 Relevant Work

Within modern theoretical linguistics, a large body of research exists on questions. Some first analyses focused on the most well-known types, i.e., deliberative, rhetorical and tag questions (Wheatley, 1955; Sadock, 1971; Cattell, 1973; Bolinger, 1978, to name only a few). Recently, researchers have studied the effect of prosody on the type of question as well as the interaction of prosody and semantics on the different types (Bartels, 1999; Dayal, 2016; Biezma and Rawlins, 2017; Beltrama et al., 2019; Eckardt, 2020, to name a few). It should also be noted that research in developing detailed pragmatic annotation schemes for human dialogs, thus also addressing questions, has a long tradition, e.g., Jurafsky et al. (1997); Novielli and Strapparava (2009); Bunt et al. (2016); Asher et al. (2016). However, most of this work is too broad and at the same time too fine-grained for our purposes: on the one hand, it does not focus on questions and thus these are not studied in the desired depth and on the other, the annotation performed is sometimes too fine-grained for computational approaches. Thus, we do not report further on this literature.

In computational linguistics, questions have mainly been studied within QA/Dialog systems, (e.g., Alloatti et al. (2019); Su et al. (2019)), and within Question Generation, (e.g., Sasazawa et al. (2019); Chan and Fan (2019)). Only a limited amount of research has focused on (versions of)

the QTI task. One strand of research has used social media data – mostly Twitter – training simple classifier models (Harper et al., 2009; Li et al., 2011; Zhao and Mei, 2013; Ranganath et al., 2016). Although this body of work reports on interesting methods and findings, the research does not follow a consistent task definition, analysing slightly different things that range from "distinguishing informational and conversational questions", "analysis of information needs on Twitter" to the identification of rhetorical questions. Additionally, they do not evaluate on a common dataset, making comparisons difficult. Furthermore, they all deal with social media data, which, despite its own challenges (e.g., shortness, ungrammaticality, typos), is enriched with further markers like usernames, hashtags and urls, which can be successfully used for the classification. A different approach to the task is pursued by Paul et al. (2011), who crowdsources human annotations for a large amount of Twitter questions, without applying any automatic recognition. More recently, the efforts by Zymla (2014), Bhattasali et al. (2015) and Kalouli et al. (2018) are more reproducible. The former develops a rule-based approach to identify rhetorical questions in German Twitter data, while Bhattasali et al. (2015) implements a machine-learning system to identify rhetorical questions in the Switchboard Dialogue Act Corpus. In Kalouli et al. (2018) a rule-based multilingual approach is applied on a parallel corpus based on the Bible.

3 RQueT: a New Corpus for QTI

The above overview of relevant work indicates that creating suitable training datasets is challenging, mainly due to the sparsity of available data. Social media data can be found in large numbers and contains questions of both types (Wang and Chua, 2010), but often the context in which the questions are found is missing or very limited, making their classification difficult even for humans. On the other hand, corpora with well-edited text such as newspapers, books and speeches are generally less suitable, as questions, in particular NISQs, tend to appear more often in spontaneous, unedited communication. Thus, to create a suitable benchmark, we need to devise a corpus fulfilling three desiderata: a) containing naturally-occurring data, b) featuring enough questions of both types, and c) providing enough context for disambiguation.

3.1 Data Collection

To this end, we find that the CNN transcripts[1] fulfill all three desiderata. We randomly sampled 2000 questions of the years 2006–2015, from settings featuring a live discussion/interview between the host of a show and guests. Questions are detected based on the presence of a question mark; this method misses the so-called "declarative" questions (Beun, 1989), which neither end with a question mark nor have the syntactic structure of a question, but this compromise is necessary for this first attempt on a larger-scale corpus. Given the importance of the context for the distinction of the question types (Dayal, 2016), along with the question, we also extracted two sentences before and two sentences after the question as context. For each of these sentences as well as for the question itself, we additionally collected speaker information. Table 1 shows an excerpt of our corpus. Unfortunately, due to copyright reasons, we can only provide a shortened version of this corpus containing only 1768 questions; this can be gained via the CNN transcripts corpus made available by Sood (2017).[2] The results reported here concern this subcorpus, but we also provide the results of the entire corpus of 2000 questions in Appendix A. Our corpus is split in a 80/20 fashion, with a training set of 1588 and a test set of 180 questions (or 1800/200 for the entire corpus, respectively).

3.2 Data Annotation

The RQueT corpus is annotated with a binary scheme of ISQ/NISQ and does not contain a finer-grained annotation of the specific sub-type of NISQ. We find it necessary to first establish the task in its binary formulation. Each question of our corpus was annotated by three graduate students of computational linguistics. The annotators were only given the definition of each type of question and an example, as presented in Section 1, and no further instructions. The lack of more detailed instructions was deliberate: for one, we wanted to see how easy and intuitive the task is for humans given that they perform it in daily communication. For another, to the best of our knowledge, there are no previous annotation guidelines or best-practices available.

The final label of each question was determined by majority vote, with an inter-annotator agreement of 89.3% and Fleiss Kappa at 0.58. This moderate

[1] http://transcripts.cnn.com/TRANSCRIPTS/

[2] See https://github.com/kkalouli/RQueT

Sentence	Text	Speaker	QT
Ctx 2 Before	*This is humor.*	S. BAXTER	
Ctx 1 Before	*I think women, female candidates, have to be able to take those shots.*	S. BAXTER	
Question	*John Edwards got joked at for his $400 hair cut, was it?*	S. BAXTER	NISQ
Ctx 1 After	*And you know, he was called a Brett Girl.*	S. BAXTER	
Ctx 2 After	This, is you know, the cut and thrust of politics.	S. BAXTER	

Table 1: Sample of the corpus format. Each row contains a sentence and its context before and after. The question and its context also hold the speaker information. Each question is separately annotated for its type.

agreement reflects the difficulty of the task even for humans and hints at the improvement potential of the corpus through further context, e.g., in the form of intonation and prosody (see e.g., Bartels 1999). The resulting corpus is an (almost) balanced set of 944 (1076 for the entire corpus) ISQ and 824 (924 for the entire corpus) NISQ. The same balance is also preserved in the training and test splits. Table 2 gives an overview of RQueT.

	ISQ	NISQ	All
Train	847 (969)	741 (831)	1588 (1800)
Test	97 (107)	83 (93)	180 (200)
Total	944 (1076)	824 (924)	1768 (2000)

Table 2: Distribution of question type in the shortened and the entire RQueT corpus, respectively.

4 RQueT as a Benchmarking Platform

We used the RQueT corpus to evaluate a variety of models,[3] establishing appropriate baselines and generating insights about the nature and peculiarities of the task.

4.1 Lexicalized and Unlexicalized Features

Following previous literature (Harper et al., 2009; Li et al., 2011; Zymla, 2014; Bhattasali et al., 2015; Ranganath et al., 2016) and our own intuitions, we extracted 6 kinds of features, 2 lexicalized and 4 unlexicalized, a total of 16 distinct features:

1. lexicalized: bigrams and trigrams of the surface forms of the question itself (*Q*), of the context-before (*ctxB1* and *ctxB2*, for the first and second sentence before the question, respectively) and of the context-after (*ctxA1* and *ctxA2*, for the first and second sentence after the question, respectively)
2. lexicalized: bigrams and trigrams of the POS tags of the surface forms of the question itself (*Q*), of the context-before (*ctxB1*, *ctxB2*) and of the context-after (*ctxA1* and *ctxA2*)
3. unlexicalized: the length difference between the question and its first context-before (*lenDiffQB*) and the question and its first context-after (*lenDiffQA*), as real-valued features
4. unlexicalized: the overlap between the words in the question and its first context-before/after, both as an absolute count (*wOverBAbs* and *wOverAAbs* for context before/after, respectively) and as a percentage (*wOverBPerc* and *wOverAPerc* for context before/after, respectively)
5. unlexicalized: a binary feature capturing whether the speaker of the question is the same as the speaker of the context-before/after (*speakerB* and *speakerA*, respectively)
6. unlexicalized: the cosine similarity of the InferSent (Conneau et al., 2017) embedding of the question to the embedding of the first context-before/after[4] (*similQB* and *similQA*, respectively).

We used these feature combinations to train three linear classifiers for each setting: a Naive Bayes classifier (NB), a Support Vector Machine (SVM) and a Decision Tree (DT). These traditional classifiers were trained with the *LightSide* workbecnh.[5] The Stanford CoreNLP toolkit (Toutanova et al., 2003) was used for POS tagging.

4.2 Fine-tuning Pretrained BERT

Given the success of contextualized language models and their efficient modeling of semantic information, e.g., Jawahar et al. (2019); Lin et al. (2019), we experiment with BERT (Devlin et al., 2019) for this task. Since the semantic relations between the question and its context are considered the most significant predictors of QT, contextualized models

[3] https://github.com/kkalouli/RQueT

[4] Here we opt for the non-contextualized InferSent embeddings because contextualized embeddings like BERT inherently exhibit high similarities (Devlin et al., 2019).

[5] http://ankara.lti.cs.cmu.edu/side/

should be able to establish a clear baseline. The QTI task can be largely seen as a sequence classification task, much as Natural Language Inference and QA. Thus, we format the corpus into appropriate BERT sequences, i.e., question-only sequence or question – context-before or question – context-after sequence, and fine-tune the pretrained BERT (base) model on that input. We explicitly fine-tune the parameters recommended by the authors. The best models train for 2 epochs, have a batch size of 32 and a learning rate of 2e-5. By fine-tuning the embeddings, we simultaneously solve the QTI task, which is the performance we report on in this setting. The fine-tuning is conducted through *HuggingFace*.[6]

4.3 BERT Embeddings as Fixed Features

The fine-tuned BERT embeddings of Section 4.2 can be extracted as fixed features to initialize further classifier models (cf. Devlin et al. 2019). We input them to the same linear classifiers used in section 4.1, i.e., NB, SVM and DT, but also use them for neural net (NN) classifiers because such architectures are particularly efficient in capturing the high-dimensionality of these inputs. To utilize the most representative fine-tuned BERT embeddings, we experiment with the average token embeddings of layer 11 and the *[CLS]* embedding of layer 11. We chose layer 11 as the higher layers of BERT have been shown to mostly capture semantic aspects, while the last layer has been found to be very close to the actual classification task and thus less suitable (Jawahar et al., 2019; Lin et al., 2019). We found that the *[CLS]* embedding performs better and thus, we only report on this setting.

Moreover, as shown in Section 5, some of the unlexicalized features of Section 4.1 lead to competitive performance with the pretrained BERT models. Thus, we decided to investigate whether the most predictive unlexicalized feature can be efficiently combined with the BERT fine-tuned embeddings and lead to an even higher performance. To this end, each linear classifier and NN model was also trained on an *extended* vector, comprising the CLS-layer11 fine-tuned BERT embedding of the respective model, i.e., only of the question (*Q-Embedding*), of the question and its (first) context-before (*Q-ctxB-Embedding*) and of the question and its (first) context-after (*Q-ctxA-Embedding*) as a fixed vector, and an additional dimension for the binary encoded unlexicalized feature.

We experimented with three NN architectures and NN-specific parameters were determined via a grid search separately for each model. Each NN was optimized through a held-out validation set (20% of the training set). First, we trained a Multi-Layer Perceptron (MLP) with a ReLU activation and the Adam optimizer. Second, we trained a feed-forward (FF) NN with 5 dense hidden layers and the RMSprop optimizer. Last, we trained an LSTM with 2 hidden layers and the RMSprop optimizer. Both the FF and the LSTM use a sigmoid activation for the output layer, suitable for the binary classification. All NNs were trained with *sklearn*.

5 Results and Analysis

5.1 Quantitative Observations

The results of the training settings are presented in Table 3. Recall that these results concern the corpus of 1768 questions. The results on the entire corpus can be found in Appendix A. For space reasons, we only present the most significant settings and results. For the lexicalized features, all models use both the surface and the POS n-grams as their combination proved best — the separate settings are omitted for brevity, so e.g., *Q tokens/POS* stands for a) the question's bigrams and trigrams and b) the question's POS bigrams and trigrams. All performance reported in Table 3 represents the accuracy of the models.

The careful benchmarking presented in Table 3 allows for various observations. We start off with the diverse combinations of lexicalized and unlexicalized features. First, we see that training only on the question, i.e., on its n-grams and POS tags, can serve as a suitable baseline with an accuracy of 62.7% for NB. Adding the first context-before improves performance and further adding the second context-before improves it even further at 72.7% for NB. A similar performance leap is observed when the first context-after is added to the question (73.3% for NB), while further adding the second context-after does not change the picture. Since adding the first context-before and -after to the question increases accuracy, we also report on the setting where both first context-before and -after are added to the question. This does indeed boost the performance even more, reaching an accuracy of 75% for NB. Given that the second context-before is beneficial for the *Q+ctxB1+ctxB2* setting, we add it to the previously best model of 75%

[6] https://huggingface.co/

Lexicalized					Unlexicalized				BERT Embeds			Classifiers						
Q tokens/POS	**ctxB1 tokens/POS**	**ctxB2 tokens/POS**	**ctxA1 tokens/POS**	**ctxA2 tokens/POS**	**speakerB**	**speakerA**	**similQA**	**lenDiffA**	**Q-Embed**	**Q-ctxB-Embed**	**Q-ctxA-Embed**	**NB**	**SVM**	**DT**	**MLP**	**FF**	**LSTM**	**BERT Fine-Tuning**
✓												62.7	61.1	63.3	-	-	-	-
✓	✓											68.8	69.4	58.5	-	-	-	-
✓	✓	✓										72.7	70	61.1	-	-	-	-
✓			✓									73.3	65	66.1	-	-	-	-
✓			✓	✓								68.8	68.8	63.3	-	-	-	-
✓	✓		✓									**75**	62.7	62.7	-	-	-	-
✓	✓	✓	✓									66.1	66.6	58.5	-	-	-	-
✓	✓	✓	✓	✓								65	67.2	58.8	-	-	-	-
					✓							57.2	57.2	57.2	57.2	57.2	56.9	-
						✓						**77.7**	**77.7**	**77.7**	**77.7**	**77.7**	**77.7**	-
					✓	✓						77.7	77.7	77.7	77.7	77.7	77.7	-
						✓	✓	✓				77.7	77.7	77.7	77.7	77.7	77.7	-
✓	✓	✓			✓							73.3	69.4	61.1	-	-	-	-
✓			✓			✓						75	73.3	**76.1**	-	-	-	-
✓			✓		✓	✓						75.5	72.7	**76.1**	-	-	-	-
✓	✓		✓			✓						74.4	71.6	62.7	-	-	-	-
✓	✓	✓	✓			✓						67.2	76.1	75.5	-	-	-	-
✓	✓		✓			✓	✓					74.4	71.6	75.5	-	-	-	-
✓	✓		✓			✓		✓				74.4	71.6	75.5	-	-	-	-
									PT			-	-	-	-	-	-	76.1
										PT		-	-	-	-	-	-	78.3
											PT	-	-	-	-	-	-	80.1
									FN			**77.7**	72.7	72.7	71.1	75.5	75.5	-
						✓			FN			77.7	80	**83.8**	80	78.8	80	-
										FN		76.6	**82.2**	72.2	77.2	80	81.1	-
					✓					FN		76.6	**81.6**	72.2	81.1	80	78.3	-
											FN	83.3	**83.3**	77.7	81.1	82.7	76.6	-
						✓					FN	83.3	83.3	81.1	82.2	**84.4**	80	-
✓	✓		✓			✓			✓	✓	✓	**Ensemble: *88.3***						

Table 3: Accuracy of the various classifiers and feature combinations (settings). A checkmark means that this feature was present in this setting. *PT* stands for the pretrained BERT embeddings and *FN* for the fine-tuned ones. Bolded figures are the best performances across types of classifiers. The stared figure is the best performing ensemble model across settings. *wOverAbs* and *wOverPerc* are omitted for brevity.

and find out that their combination rather harms the accuracy. Experimenting with both contexts-before and -after and the question does not lead to any improvements either. The combinations of the lexicalized features show that the best setting is the one where the question is enriched by its first context-before and -after (75%).

We make a striking observation with respect to the unlexicalized features. Training only on the speaker-after, i.e., on whether the speaker of the question is the same as the speaker of the first context-after, and ignoring entirely the question and context representation is able to correctly predict the QT in 77.7% of the cases. This even outperforms the best setting of the lexicalized features. The speaker-before does not seem to have the same expressive power and training on both speaker features does not benefit performance either. We also find that the rest of the unlexicalized features do not have any impact on performance because training on each of them alone hardly outperforms the simple *Q tokens/POS* baseline, while by training on all unlexicalized features together we do not achieve better results than simply training on speaker-after.[7]

Based on the finding that the speaker-after is so powerful, we trained hybrid combinations of lexicalized features and the speaker information. First, the speaker-before is added to the *Q+ctxB1+ctxB2*, which is the best setting of contexts-before, but we do not observe any significant performance change. This is expected given that speaker-before alone does not have a strong performance. Then, the speaker-after is added to the setting *Q+ctxA1* and

[7]These settings are omitted from the table for brevity.

the performance reaches 76.1% (for DT), approaching the best score of speaker-after. The addition of speaker-before to this last setting does not improve performance. On the other hand, adding the speaker-after information to the best lexicalized setting (*Q+ctxB1+ctxA1*) does not have an effect, probably due to a complex interaction between the context-before and the speaker. This performance does not benefit either from adding the second context-before (which proved beneficial before) or adding the other unlexicalized features.[8]

Moving on, we employ the pretrained BERT embeddings to solve the QTI task. Here, we can see that the model containing the question and the context-after (*Q-ctxA-Embedding*) is the best one with 80.1%, followed by the model containing the question and the context-before (*Q-ctxB-Embedding*, 78.3). Worst-performing is the model based only on the question (*Q-Embedding*). This simple fine-tuning task shows that contextualized embeddings like BERT are able to capture the QT more efficiently than lexicalized and unlexicalized features – they even slightly outperform the powerful speaker feature. This means that utilizing these fine-tuned embeddings as fixed input vectors for further classifiers can lead to even better results, and especially, their combination with the predictive speaker information can prove beneficial.

In this last classification setting, we observe that the classifiers trained only on the fine-tuned BERT embeddings deliver similar performance to the fine-tuning task itself. This finding reproduces what is reported by Devlin et al. (2019). However, the real value of using this feature-based approach is highlighted through the addition of the speaker information to the contextualized vectors. The speaker information boosts performance both in the setting of *fine-tuned Q-Embedding* and in the setting *fine-tuned Q-ctxA-Embedding*. In fact, the latter is the best performing model of all with an accuracy of 84.4%. Adding the speaker-before information to the *fine-tuned Q-ctxB-Embedding* does not have an impact on performance due to the low impact of the speaker-before feature itself.

5.2 Qualitative Interpretation

The results presented offer us interesting insights for this novel task. First, they confirm the previous finding of the theoretical and computational literature that context is essential in determining the question type. Both the lexicalized and the embeddings settings improve when context is added. Concerning the lexicalized settings, we conclude that the surface and syntactic cues present within the question and its first context-after are more powerful than the cues present within the question and the first context-before. This is consistent with the intuition that whatever follows a question tends to have a more similar structure to the question itself than whatever precedes it: no matter if the utterer of the question continues talking or if another person addresses the question, the attempt is to stay as close to the question as possible, to either achieve a specific communication goal or to actually answer the question, respectively. However, our experiments also show that combining the first context-before and -after with the question does indeed capture the most structural cues, generating the insight that one sentence before and after the question is sufficient context for the task at hand. Interestingly, we can confirm that the second context-after is not useful to the classification of the QT, probably being too dissimilar to the question itself. Table 4 shows examples of the most predictive structural cues for the best setting of the lexicalized classifiers (*Q+ctxB1+ctxA1*).

ISQ	*you_feel, what_do_you, do_you_agree, make_of_that, you_expect, me_ask_you, why_did_you, how_did_you*
NISQ	*why_arent't, and_should_we, COMMA_how_about, how_could, do_we_want, can_we*

Table 4: Structural features with the most influence in the model *Q+ctxB1+ctxA1*.

Training on non-linguistic unlexicalized features does not boost performance. However, our work provides strong evidence that the speaker meta-information is of significant importance for the classification. This does not seem to be a peculiarity of this dataset as later experimentation with a further English dataset and with a German corpus shows that the speaker information is consistently a powerful predictor. Additionally, we can confirm from Appendix A that the speaker feature has the same behavior, when trained and tested on the entire corpus. To the best of our knowledge, previous literature has not detected the strength of this feature. From the prediction power of this feature, it

[8] Although the unlexicalized features had shown no significant performance, they were added here to check for interaction effects between them and the lexicalized features.

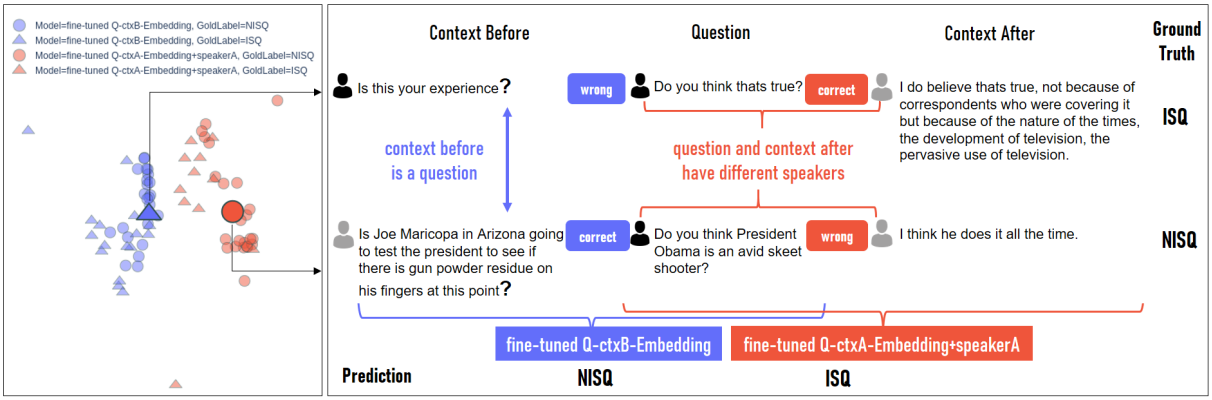


Figure 1: Interactive visualization of the wrongly predicted instances of the models *fine-tuned Q-ctxB-Embedding* and *fine-tuned Q-ctxA-Embedding+speakerA*. Based on this visualization, we can observe sentences with similar patterns and how these are learned from the models. Some sentences are ambiguous having both patterns; thus, we need a third model for our ensemble.

might seem that information on the question and its context is not necessary at all. However, we show that the addition of the linguistic information of the question and its context through the fine-tuned embeddings provides a clear boost for the performance. The importance of similar linguistic unlexicalized features has to be investigated in future work. In fact, for the current work, we also experimented with the topic information, i.e., based on topic modeling, we extracted a binary feature capturing whether the topic of the question and the context-after is the same or not. However, this feature did not prove useful in any of the settings and was thus omitted from the analysis. Future work will have to investigate whether a better topic model leads to a more expressive binary feature and whether other such features, such as sentiment extracted from a sentiment classification model, can prove powerful predictors.

Concerning the distributional and NN methods, this is the first work employing such techniques for the task and confirming the findings of the more traditional machine learning settings. Fine-tuning the pretrained BERT embeddings reproduces what we showed for the standard classifiers: the context and especially the context-after boosts the performance. This finding is also confirmed when treating the fine-tuned BERT embeddings as standard feature vectors and further training on them. Most importantly, this setting allows for the expansion of the feature vector with the speaker information: this then leads to the best performance. Unsurprisingly, the speaker-before is not beneficial for the classification, as it was not itself a strong predictor. Finally, we also observe that the results reported for this smaller corpus are parallel to the results reported for the entire corpus (see Appendix A).

5.3 Further Extension & Optimization

By studying Table 3 the question arises whether our best-performing model of *fine-tuned Q-ctxA-Embedding+speakerA* can be further improved and crucially, whether the context-before can be of value. With our lexicalized models, we show that the best models are those exploiting the information of the context-before, in addition to the question and the context-after. However, all of our BERT-based models have been trained either on the combination of question and context-before or on the combination of question and context-after, but never the combination of all three. The inherent nature of the BERT model, which requires the input sequence to consist of a pair, i.e., at most two distinct sentences separated by the special token *[SEP]*, is not optimized for a triple input. On the other hand, "tricking" BERT into considering the context-before and the question as one sentence delivers poor results. Thus, we decided to exploit the power of visualization to see whether an ensemble model combining our so far best performing model of *fine-tuned Q-ctxA-Embedding+speakerA* with our context-before BERT-based model *fine-tuned Q-ctxB-Embedding* would be beneficial.

To this end, we created a small interactive Python visualization to compare the two models, using UMAP (McInnes et al., 2018) as a dimensionality reduction technique and visualizing the datapoints in a 2D scatter plot. We computed positions jointly for both models and projected them into the same 2D space using cosine similariy as the

distance measure. As we are interested in potential common wrong predictions between the models, we only visualize wrongly classified samples, and group them by two criteria: the model used (color-encoded) and the gold label (symbol-encoded).

Examining the visualization of Figure 1 (left) we observe that there is no overlap between the wrongly predicted labels of the two models. This means that training an ensemble model is a promising way forward. Additionally, through the interactive visualization, we are guided to the most suitable ensemble model. Particularly, we see some common patterns for the wrongly predicted labels for each of the models. The *fine-tuned Q-ctxA-Embedding+speakerA* has a better performance in predicting ISQ, whereby the decision seems to be influenced by the speaker feature (i.e., if the question and context-after have different speakers, the model predicts ISQ). However, the *fine-tuned Q-ctxB-Embedding* model seems to learn a pattern of a context-before being a question; in such cases, the target question is predicted as NISQ. In the ground truth we have ambiguous cases though, where questions have both patterns. Thus, although it seems that the two models fail on different instances and that they could thus be combined in an ensemble, they would alone likely fail in predicting the ambiguous/controversial question instances. Instead, surface and POS features of the questions and their contexts should be able to differentiate between some of the controversial cases. To test this, we created an ensemble model consisting of the two models and the best lexicalized model holding such features (*Q+ctxB1+ctxA1*). First, this ensemble model checks whether *fine-tuned Q-ctxA-Embedding+speakerA* and *fine-tuned Q-ctxB-Embedding* predict the same label. If so, it adopts this label too. Otherwise, it picks up the prediction of *Q+ctxB1+ctxA1*. With this ensemble approach, we are indeed able to improve our so-far best model by 4%, reaching an accuracy of 88.3%, as shown in the last entry of Table 3.

At this point, two questions arise. First, the reader might wonder whether this result means that the task is virtually "solved". Recall that the inter-annotator agreement was measured at 89.3% and thus, it might seem that our ensemble model is able to be competitive with that. However, this is not the case: if we observe the Fleiss Kappa, we see that it only demonstrates moderate agreement. This could be due to the difficulty of the task, as mentioned before, but it also shows that the task formulation has room for improvement. In a post-annotation session, our annotators reported that some of the uncertainty and disagreement could be tackled with multi-modal data, where also audio or video data of the corresponding questions is provided. Additionally, higher agreement could have been achieved with more annotators. Thus, our current work offers room for improvement, while providing strong baselines. Second, the question is raised whether this feature combination is indeed the best setting for all purposes of this task; the answer to this depends on what the ultimate goal of this task is. If the ultimate goal is application-based, where a model needs to determine whether a question requires a factoid answer (or not) in a real-life conversation, the trained model should not include the context-after as a feature as this would exactly be what we want to determine based on the model's decision. However, if the goal is to automatically classify questions of a given corpus to generate linguistic insights, then the trained model can include all features. The evaluation undertaken here serves both these purposes by detailing all settings. On the one hand, we show that the models achieve high performance even when removing the context-after and that therefore an application-based setting is possible. On the other hand, we also discover which feature combination will lead to the best predictions, generating theoretical insights and enabling more research in this direction.

6 Conclusion

In this paper, we argued for the need of the Question-Type Identification task, in which questions are distinguished based on the communicative goals they are set to achieve. We also provided the first corpus to be used as a benchmark. Additionally, we studied the impact of different features and established diverse baselines, highlighting the peculiarities of the task. Finally, we were able to generate new insights, which we aim to take up on in our future work.

Acknowledgements

We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for funding within project BU 1806/10-2 "Questions Visualized" of the FOR2111 "Questions at the Interfaces". We also thank our annotators, as well as the anonymous reviewers for their helpful comments.

References

Francesca Alloatti, Luigi Di Caro, and Gianpiero Sportelli. 2019. Real Life Application of a Question Answering System Using BERT Language Model. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 250–253, Stockholm, Sweden. Association for Computational Linguistics.

Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727, Portorož, Slovenia. European Language Resources Association (ELRA).

Christine Bartels. 1999. *The intonation of English statements and questions*. New York: Garland Publishing.

Andrea Beltrama, Erlinde Meertens, and Maribel Romero. 2019. Decomposing cornering effects: an experimental study. *Proceedings of Sinn und Bedeutung*, 22(1):175–190.

Robbert-Jan Beun. 1989. *The recognition of declarative questions in information dialogues*. Ph.D. thesis, Tilburg University. Pagination: 139.

Shohini Bhattasali, Jeremy Cytryn, Elana Feldman, and Joonsuk Park. 2015. Automatic Identification of Rhetorical Questions. 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 743–749, Beijing, China. Association for Computational Linguistics.

María Biezma and Kyle Rawlins. 2017. Rhetorical questions: Severing asking from questioning. *Semantics and Linguistic Theory*, 27:302.

Daniel G. Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007. PARC's Bridge and Question Answering System. In *Proceedings of the Grammar Engineering Across Frameworks Workshop (GEAF 2007)*, pages 46–66, Stanford, California, USA. CSLI Publications.

Dwight Bolinger. 1978. Yes—no questions are not alternative questions. In *Questions*, pages 87–105. Springer.

Harry Bunt, Volha Petukhova, Andrei Malchanau, Kars Wijnhoven, and Alex Fang. 2016. The DialogBank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3151–3158, Portorož, Slovenia. European Language Resources Association (ELRA).

Ray Cattell. 1973. Negative Transportation and Tag Questions. *Language*, 49(3):612–639.

Ying-Hong Chan and Yao-Chung Fan. 2019. BERT for Question Generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 173–177, Tokyo, Japan. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Veneeta Dayal. 2016. *Questions*. Oxford University Press, Oxford.

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.

Regine Eckardt. 2020. Conjectural questions: The case of German verb-final wohl questions. *Semantics and Pragmatics*, 13:1–17.

David Ferrucci, Anthony Levas, Sugato Bagchi, David Gondek, and Erik T. Mueller. 2013. Watson: Beyond Jeopardy! *Artificial Intelligence*, 199–200(1):93–105.

F.M. Harper, D. Moy, and J. A. Konstan. 2009. Facts or Friends? Distinguishing Informational and Conversational Questions in Social Q&A Sites. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2009)*, pages 759–768.

Robin Håvik, Jo Dugstad Wake, Eivind Flobak, Astri Lundervold, and Frode Guribye. 2019. A conversational interface for self-screening for adhd in adults. In *Internet Science*, pages 133–144, Cham. Springer International Publishing.

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.

D. Jurafsky, E. Shriberg, and D. Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual. Technical Report Draft 13, University of Colorado, Institute of Cognitive Science.

Aikaterini-Lida Kalouli, Katharina Kaiser, Annette Hautli-Janisz, Georg A. Kaiser, and Miriam Butt. 2018. A Multilingual Approach to Question Classification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Baichuan Li, Xiance Si, Michael R. Lyu, Irwin King, and Edward Y. Chang. 2011. Question Identification on Twitter. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 2477–2480, New York, NY, USA. Association for Computing Machinery.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting inside BERT's Linguistic Knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Kien Hoa Ly, Ann-Marie Ly, and Gerhard Andersson. 2017. A fully automated conversational agent for promoting mental well-being: A pilot RCT using mixed methods. *Internet Interventions*, 10:39–46.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software*, 3(29):861.

Nicole Novielli and Carlo Strapparava. 2009. Towards unsupervised recognition of dialogue acts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 84–89, Boulder, Colorado. Association for Computational Linguistics.

Sharoda A. Paul, Lichan Hong, and Ed H. Chi. 2011. What is a question? Crowdsourcing tweet categorization. In *CHI 2011, Workshop on Crowdsourcing and Human Computation*.

Suhas Ranganath, Xia Hu, Jiliang Tang, SuhangWang, and Huan Liu. 2016. Identifying Rhetorical Questions in Social Media. In *Proceedings of the 10th International AAAI Conference on Web and Social Media (ICWSM 2016)*.

Jerrold Sadock. 1971. Queclaratives. In *Papers from the 7th Regional Meeting of the Chicago Linguistic Society*, pages 223–232.

Yuichi Sasazawa, Sho Takase, and Naoaki Okazaki. 2019. Neural Question Generation using Interrogative Phrases. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 106–111, Tokyo, Japan. Association for Computational Linguistics.

Gaurav Sood. 2017. CNN Transcripts 2000–2014. Published by Harvard Dataverse, retrieved from https://doi.org/10.7910/DVN/ISDPJU.

Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 2019. Generalizing Question Answering System with Pre-trained Language Model Fine-tuning. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 203–211, Hong Kong, China. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

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.

Kai Wang and Tat-Seng Chua. 2010. Exploiting salient patterns for question detection and question retrieval in community based question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING10)*, page 1155–1163.

J. M. O. Wheatley. 1955. Deliberative questions. *Analysis*, 15(3):49–60.

Zhe Zhao and Qiaozhu Mei. 2013. Questions about Questions: An Empirical Analysis of Information Needs on Twitter. In *Proceedings of the International World Wide Web Conference Committee (IW3C2)*, pages 1545–1555.

Mark-Matthias Zymla. 2014. Extraction and Analysis of non-canonical Questions from a Twitter-Corpus. Master's thesis, University of Konstanz.

Appendix A: Performance Results on the entire RQueT

The following table collects all performance results when training on the entire RQueT corpus of 2000 questions. Although we cannot make this whole corpus available, we would like to report on the performance to show how our findings are parallel in both variants of the corpus and that the smaller size of the corpus we make available does not obscure the overall picture.

Lexicalized					Unlexicalized				BERT Embeds			Classifiers						
Q tokens/POS	**ctxB1 tokens/POS**	**ctxB2 tokens/POS**	**ctxA1 tokens/POS**	**ctxA2 tokens/POS**	**speakerB**	**speakerA**	**similQA**	**lenDiffA**	**Q-Embed**	**Q-ctxB-Embed**	**Q-ctxA-Embed**	**NB**	**SVM**	**DT**	**MLP**	**FF**	**LSTM**	**BERT Fine-Tuning**
✓												64.5	61.5	59	-	-	-	-
✓	✓											67.5	67	62	-	-	-	-
✓	✓	✓										72.5	62.5	56.5	-	-	-	-
✓			✓									73	62	63	-	-	-	-
✓			✓	✓								71	65.5	62.5	-	-	-	-
✓	✓		✓									**75.5**	65.5	61	-	-	-	-
✓	✓	✓	✓									67.5	62	60	-	-	-	-
✓	✓	✓	✓	✓								66.5	61	58	-	-	-	-
					✓							57	57	57	57	56.9	56.9	-
						✓						**78.5**	**78.5**	**78.5**	**78.5**	**78.5**	**78.5**	-
					✓	✓						78.5	78.5	78.5	78.5	78.5	78.5	-
						✓	✓	✓				78.5	77.5	78.5	-	-	-	-
✓	✓	✓			✓							72	65	56.5	-	-	-	-
✓			✓			✓						77.5	69	75	-	-	-	-
✓			✓		✓	✓						76	70	75	-	-	-	-
✓	✓		✓			✓						**78**	73	77	-	-	-	-
✓	✓	✓	✓			✓						69	71	75.5	-	-	-	-
✓	✓		✓			✓	✓					78	74.5	76.5	-	-	-	-
✓	✓		✓			✓		✓				78	72	77	-	-	-	-
									PT			-	-	-	-	-	-	76.4
										PT		-	-	-	-	-	-	77.4
											PT	-	-	-	-	-	-	79.4
									FN			76.5	76	72.5	**77**	76.4	72.5	-
						✓			FN			77	78.5	73	**80**	79.5	76.4	-
										FN		76	78.5	78.5	**80**	79.5	79.5	-
					✓					FN		76	79	78.5	78.5	**80**	79	-
											FN	78.5	78	**79.5**	78.5	**79.5**	76.4	-
						✓					FN	78.5	80	80	81.5	**82.4**	78.5	-
✓	✓		✓			✓			✓	✓	✓	**Ensemble: *85***						

Table 5: Accuracy of the various classifiers and feature combinations (settings) on the entire RQueT corpus of 2000 questions. A checkmark means that this feature was present in this setting. *PT* stands for the pretrained BERT embeddings and *FN* for the fine-tuned ones. Bolded figures are the best performances across types of classifiers. The stared figure is the best performing ensemble model across settings. *wOverAbs* and *wOverPerc* are omitted for brevity.

New Domain, Major Effort? How Much Data is Necessary to Adapt a Temporal Tagger to the Voice Assistant Domain

Touhidul Alam
Liquid Studio, Accenture
Kronberg, Germany
touhidul.alam@accenture.com

Alessandra Zarcone
HumAIn Labs, Fraunhofer IIS
Erlangen, Germany
zce@iis.fraunhofer.de

Sebastian Padó
IMS, Universität Stuttgart
Stuttgart, Germany
pado@ims.uni-stuttgart.de

Abstract

Reliable tagging of Temporal Expressions (TEs, e.g., *Book a table at L'Osteria for Sunday evening*) is a central requirement for Voice Assistants (VAs). However, there is a dearth of resources and systems for the VA domain, since publicly-available temporal taggers are trained only on substantially different domains, such as news and clinical text.

Since the cost of annotating large datasets is prohibitive, we investigate the trade-off between in-domain data and performance in DA-Time, a hybrid temporal tagger for the English VA domain which combines a neural architecture for robust TE recognition, with a parser-based TE normalizer. We find that transfer learning goes a long way even with as little as 25 in-domain sentences: DA-Time performs at the state of the art on the news domain, and substantially outperforms it on the VA domain.

1 Introduction

Many Natural Language Processing (NLP) applications rely on a temporal tagger to successfully identify and normalize temporal expressions (TEs: e.g. *seven in the evening* → *T19:00*). Examples include question answering, summarization, and information extraction (Strötgen and Gertz, 2016). Temporal tagging serves to anchor events on the temporal axis and contributes to event ordering sequences (UzZaman and Allen, 2010). This is particularly useful for Voice Assistants (VAs), that is software agents such as Apple's Siri or Amazon's Alexa, which are able to interpret spoken human queries (commands) and help their users perform simple tasks, including scheduling tasks such as *setting reminders* or *creating and editing* calendar events. For example, given the query *Delete my Monday's meeting*, a VA might have to retrieve information from a calendar corresponding to the day the user is referring to as *Monday*. In order to succeed in such tasks, VAs require a reliable temporal tagger, which can identify TEs and classify them into categories (TE recognition, for example, DATE vs. TIME) and then convert them into machine-readable canonical values (TE normalization, e.g. *seven in the evening* → *T19:00*).

The major shortcoming of current temporal taggers is arguably their domain dependence, as it is well known that NLP tools degrade on out-of-domain data. The publicly available temporal taggers (Chang and Manning, 2012; Filannino et al., 2013; Strötgen and Gertz, 2013; Lee et al., 2014) have been developed and evaluated on domain-specific datasets annotated according to the TimeML standard (Pustejovsky et al., 2003a), notably the news (Pustejovsky et al., 2003b), social media (Zhong et al., 2017), narrative (Mazur and Dale, 2010), or clinical domain (Galescu and Blaylock, 2012). In contrast, to our best knowledge, there is no existing temporal tagger optimized for the VA domain, which differs considerably from other domains: it is dominated by concise stand-alone commands, typically referring to single future events (e.g., *Add yoga to my calendar tomorrow at 6*), often outside disambiguating discourse. As a result, coreference and event ordering play a smaller role than in other domains. Also, VA queries, compared to the news domain, contain more references to the time of an event (*at 6*) and to regular event repetitions (*Wake me up every day at 7*), as well as more underspecified or vague time expressions (*Remind me to call mom later this evening*) (Rong et al., 2017; Tissot et al., 2019).

A possible solution to overcome the problem of the scarcity of tagged training data for the VA domain is to adopt a transfer learning approach (Bengio, 2011). However, this leaves open the question of what the training curve looks like: how

Proceedings of the 14th International Conference on Computational Semantics, pages 144–154
June 17–18, 2021.

```
Add my appointment at Varin Salon on
<TIMEX3 tid="t1" type="DATE" value="
    2020-04-27"> April 27th </TIMEX3>
from
<TIMEX3 tid="t2" type="TIME" value="
    2020-04-27T10:30" anchorTimeID="t1">
10:30 am </TIMEX3>
to
<TIMEX3 tid="t3" type="TIME" value="
    2020-04-27T11:30" anchorTimeID="t1">
11:30 am </TIMEX3>
<TIMEX3 tid="t4" type="DURATION"  value=
    "PT1H" beginPoint="t2" endPoint="t3"
     />
to the calendar.
```

Figure 1: TimeML example from Zarcone et al. (2020).

much data is necessary until performance "flattens out"? We investigate the performance of a temporal tagger pre-trained on news and fine-tuned on the VA domain and find that a surprisingly small amount of data (less than 100 in-domain sentences) is sufficient to achieve reasonable performance on the low-resource target domain, substantially outperforming existing systems on the VA domain.

Paper structure. We first contrast annotated data in the news and VA domain (Sec. 2). After an overview of related work (Sec. 3), we introduce DA-Time, a hybrid temporal tagger for the VA domain, which uses a neural model for TE recognition and a parsing-based model for TE normalization (Sec. 4). After describing the experimental setup (Sec. 5), we present a detailed evaluation for varying amount of target domain annotations (Sec. 6).

2 Annotation and Data

2.1 The TimeML Markup Standard

TimeML is a widely-adopted framework for annotating time, events and event relations in text following the ISO 8601 standard[1] (Pustejovsky et al., 2003a). TimeML has also been used for the influential TempEval competitions (Verhagen et al., 2007, 2010; UzZaman et al., 2013) which form the basis for most work on temporal tagging. TimeML specifies four major data structures: EVENT, TIMEX3, SIGNAL, and LINK. Among these, TIMEX3 describes TEs; EVENT, SIGNAL, and LINK describe relations among TEs. For the purposes of this study, we focus on TIMEX3 and do not take relations among events into account, as motivated by the lower significance of such relations for VAs.

[1]ISO 8601 is an international standard covering the exchange of date- and time-related data

TE	Value Pattern (*type*)	Unit
Last summer	YYYY-SS (DATE)	Season
Last year	YYYY (DATE)	Year
This month	YYYY-MM (DATE)	Month
Next week	YYYY-WXX (DATE)	Week
Sunday the 5th	YYYY-MM-DD (DATE)	Day
7 pm tonight	YYYY-MM-DDTHH (TIME)	Hour
15 minutes later	YYYY-MM-DDTHH:MM (TIME)	Minute
At 3:07:15	YYYY-MM-DDTHH:MM:SS (TIME)	Second

Table 1: Examples of temporal units, with corresponding TE examples and their value patterns.

TEs in TIMEX3 are classified into four *types*: DATE (e.g., *May 2nd*), TIME (e.g., *tomorrow morning*), DURATION (e.g., *an hour*), SET (e.g., *every Monday*). An example is given in Figure 1. Each TE in TIMEX3 is identified by a unique ID (*tid* attribute). TEs are assigned *values* in a normalized machine-readable format following the ISO 8601 standard. Reference date information is also included on TIME type, which refers to the date to which the TE is anchored. TEs of *type* DURATION are also tagged with a *beginPoint* and *endPoint*, corresponding to the *tid* of the two TEs the DURATION *type* expression is anchored to. As Figure 1 shows, sometimes the range of a duration remains underspecified. In this case, an *empty tag* of *type* DURATION is added. Similarly, if only the duration range and either the beginning or end point are mentioned (e.g. *Book the room from 10:30 am for two hours*), then an empty TIME *type* tag is added to indicate the missing TE. If the value of a TE is derived from the value of another one, the *anchorTimeID* attribute indicates which TE the tagged TIMEX3 is anchored to.

On a more fine-grained level, TEs can be described using *temporal units* at different levels of granularity (Strötgen and Gertz, 2016), e.g. *the 2nd week of February*, *the 2nd day of February*), *next February* (month). These units are not explicitly annotated in TIMEX3, but they can be used to identify different value patterns (see Table 1).

2.2 Datasets

We now introduce the TimeML-annotated English datasets in the source (news) and target domain (VA). Descriptive statistics are reported in Table 2.

News domain The news domain is widely studied because of the vast availability of news text, and

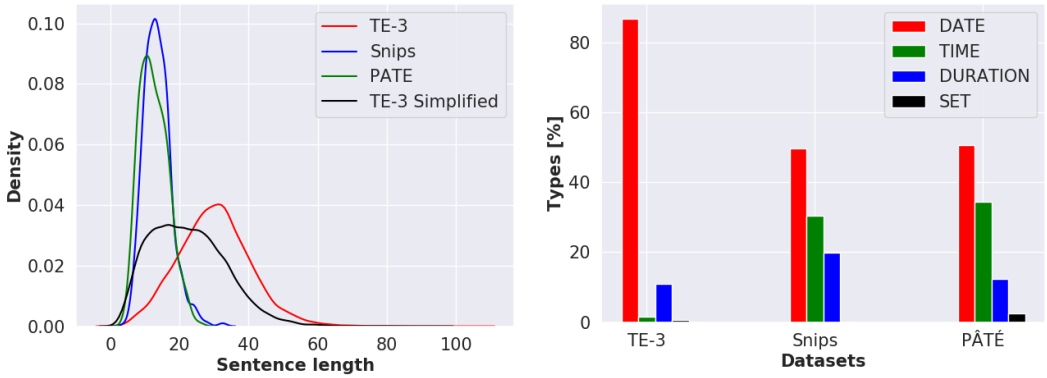


Figure 2: Comparison between the news and VA corpora on Sentence length distribution across datasets (left) and TIMEX3 type distribution (right). The Figure on the right includes empty tags for Snips and PÂTÉ.

		Tokens	Sent.s w/ TIMEXes	# of TIMEXes
News	TBAQ	99420	1469	1822
	TE-3 Silver	713091	10020	12739
	TE-3 (TBAQ+Silver)	812511	11489	14561
	TE-3 Simplified	289897	12897	14561
	TE-3 Platinum	7009	106	138
VA	Snips	9677	697	947
	PÂTÉ	5633	353	767

Table 2: Statistics on datasets for two domains (TE-3: TempEval-3[2]). TE-3 Simplified is described in 5.1.

the importance of TEs for relationships between reported events. In TempEval-3 (UzZaman et al., 2013), the manually annotated TBAQ corpus, consisting of TimeBank and AQUAINT corpus, was used as a training set (99K tokens) (Pustejovsky et al., 2003b). Additionally, a 700K-token machine-annotated corpus (TE-3 Silver) was created from Gigaword (Parker et al., 2011). Furthermore, a *platinum* set (TE-3 Platinum) was provided for evaluation, which had a higher inter-annotator agreement than existing TimeML corpora (hence the name).

Voice Assistant domain Two datasets have recently become available for the VA domain: Snips (Coucke et al., 2018) and PÂTÉ (Zarcone et al., 2020). Snips is a widely-adopted dataset for benchmarking intent and entity classification in the VA domain. No details are provided on how Snips was created. A subset of Snips was annotated with TimeML/TIMEX3 tags by Zarcone et al. (2020). PÂTÉ is a TE-rich crowdsourced dataset for the VA domain, whose collection effort was specifically focused on eliciting naturally-sounding commands containing a wide variety of TEs. As such, we focus on PÂTÉ for our final evaluation.

2.3 Cross-domain Comparison

A comparison between the news and VA domains on the basis of the abovementioned corpora is shown in Figure 2. News texts are typically grammatical and coherent reports of past events that took place at a certain moment in time. The news datasets contain longer sentences (Figure 2, left), with longer-distance relationships between events (e.g. *After that year*) that pose a challenge for normalization. VA commands, on the other hand, are comparatively shorter, and they do not provide a large sentence context nor do they typically contain references to previous event mentions. Typically, TEs in VA domain are used to refer to future events. In some cases, VA commands can contain multiple TEs, posing a challenge to the normalizer in identifying the relations among them (e.g., *Move yoga from Monday at 8 pm to Sunday at 7).*

Figure 2 (right) shows the distribution of TIMEX3 types in the datasets. It is skewed towards DATE throughout, but DATE is even more dominant in TempEval. TIME *type* TEs are substantially underrepresented in the news domain compared to the VA domain: news are generally reported on a daily level of granularity, whereas scheduling tasks require more fine-grained temporal descriptions. Granularity differences are also reflected in the *unit* distribution: the news domain mostly contains units of type DAY (48%), while in the VA domain HOUR and DAY are equally represented as the most frequent *units* (52% DAY, 40% HOUR).

Another difference between the datasets in Figure 2 is that the VA domain datasets contain a substantial number of empty tags, which are typ-

[2]TempEval-3 Task: https://www.cs.york.ac.uk/semeval-2013/task1/index.html

ical of VA interactions where temporal information can be inferred from context (e.g., *Remind me in two hours* where the inferred absolute time information can be used to set a reminder). Snips and PÂTÉ contain around 20% and 10% empty tags respectively. In Snips, 18.6% of the DATE tags and 25.4% of the TIME (but none of the DURATION tags) are empty tags. In PÂTÉ, 91% of the DURATION tags are empty tags but only 1% of the DATE tags and 1.8% of the TIME tags are empty tags. Most of the empty tags in PÂTÉ (90%) are DURATION tags, while in Snips, they are either DATE (43%) or TIME (57%) tags. Meanwhile, the news datasets do not use empty tags in their annotation at all, so a comparison is not possible.

In sum, we can expect temporal taggers that are optimized on news to perform worse on the VA domain given the differences in distribution of types, units, and domain-specific features they rely on.

3 Related Work

The first TempEval challenge (Verhagen et al., 2007) focused on the automatic extraction of temporal relations given a TimeML-annotated dataset. TempEval-2 (Verhagen et al., 2010) introduced the task of temporal tagging of TEs for the English news domain, consisting in their recognition and normalization, and as a prerequisite for temporal information extraction, which also includes the extraction of events and of their temporal relations. TempEval-3 (UzZaman et al., 2013) extended the task to multilingual settings providing TIMEX3 annotation in English and Spanish. More recent TempEval challenges (Bethard et al., 2015, 2016, 2017) also branched out to the clinical domain. As to temporal tagging in different domains (e.g., news, narrative, colloquial, autonomic), Strötgen and Gertz (2016) addressed potential challenges, observing that existing temporal taggers work sufficiently well only in the domain they were developed for. This is probably why, to the best of our knowledge, work on temporal tagging has so far only been considered in within-domain settings.

TempEval-3 can serve as a showcase of approaches to temporal tagging. The nine participants tackled the task either with rule-based, data-driven, or hybrid methods (UzZaman et al., 2013). HeidelTime (Strötgen et al., 2013), a rule-based system, obtained the top rank. The system used regular expression-based rules to identify and normalize time expressions in multilingual settings (Strötgen and Gertz, 2015). Later, they extended their rules to cover different domains (e.g., narrative, colloquial) (Strötgen and Gertz, 2016). When TEs were underspecified (e.g. *January 6th*), domain-sensitive strategies (such as searching for contextual cues or identifying a reference time) were adopted to normalize them (e.g. to normalize *January 6th* as the previous January 6th or the forthcoming one). As rule-based systems are typically crafted to work for their reference domain, HeidelTime is not able to identify and normalize expressions that are more typical of concise commands to a VA, such as *Book a slot for the 5th*, where the month is not mentioned. UW-Time (Lee et al., 2014) is a hybrid semantic parsing-based tagger using Combinatory Categorial Grammar (Steedman and Baldridge, 2011). Compared to HeidelTime, UW-Time successfully combines hand-engineered and trained rules, showing the benefit of context-handling over rule-based approach. UW-Time can use features such as the tense of a verb to determine if the TEs refer to either the past or the future, or can determine if a four-digit number in a text refer to a year or not depending on the context. UW-Time was evaluated on the news and narrative domain and set the current state-of-the-art of temporal tagging on the TempEval-3 evaluation set, working exceptionally well but with a high degree of domain specificity.

4 DA-Time

We now present a hybrid system for temporal tagging, which we use to investigate domain adaptation of temporal tagging: DA-Time (for Domain-Adapted Time Tagger). DA-Time is a pipeline of a neural TE recognizer and a rule-based normalizer[3].

4.1 TE Recognizer

We frame TE recognition as a joint TE *type* and *unit* classification tasks. As argued in Tissot et al. (2019), temporal unit or granularity is a key feature of TEs, and can be expected to improve TE recognition, in particular for imprecise TEs[4], for example those formed by a temporal unit of a specific degree of granularity and a fuzzy quantifier (e.g., *some days, several weeks, years after*). We adopt a sequence-labelling architecture influenced by the neural NER model of Lample et al. (2016).

[3]The implementation of the TE recognizer is available at this Github repository under an academic use license: https://github.com/audiolabs/DA-Time/

[4]Since temporal unit is not an explicit part of TIMEX3, we derive it from the normalized value (details in Section 5.1).

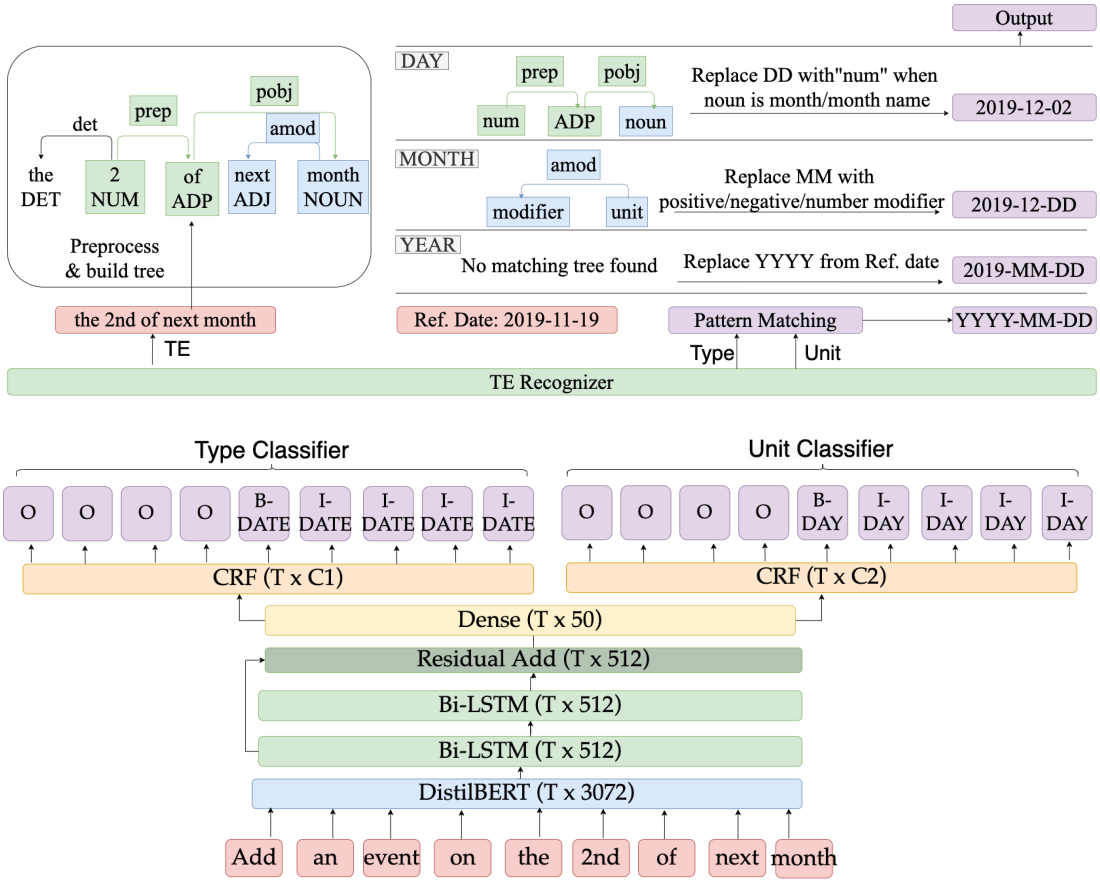


Figure 3: TE recognizer (bottom) and TE normalizer (top). Example input: *Add an event on the 2nd of next month*. Recognizer output (bottom): *the 2nd of next month* as DATE *type* and DAY *unit*. Normalizer (top): Given the recognizer output, reference date, and dependency analysis of TE, the rules are checked sequentially. The output is a normalized value for the TE.

The model takes a sentence as an input sequence and predicts *type* and *unit* in a BIO labeling scheme, as shown in Figure 3 (bottom). We use a contextualized embedding model, DistilBERT[5] (Sanh et al., 2019), as an embedding layer. DistilBERT is a smaller and faster version of BERT (Devlin et al., 2019) which is compressed during pre-training by using knowledge distillation. This improves on the inference speed compared to BERT. The embedding layer is followed by two Bi-LSTM layers. An add layer after the second Bi-LSTM which acts as a *residual add* or *skip connection* layer to improve learning (He et al., 2016). Finally, a dense layer followed by two different Conditional Random Field (CRF) layers on top is added.

Baseline model No other neural model is available as a baseline for the task of full temporal tagging of the PÂTÉ dataset, and due to its size the dataset would not be suitable for training a neural model on it. However a reasonable alternative is to adopt a pre-trained language model (Peters et al., 2018; Howard and Ruder, 2018). We propose a DistilBERT + CRF based model as a baseline, where DistilBERT is used as a pre-trained model and CRF is used to extract the labels (*type* and *unit*).

Transfer learning We apply the two approaches proposed by Felbo et al. (2017). The first method, *chain*, fine-tunes each layer sequentially (except the embedding layer in our experiment), freezing all the other layers. The second method, *full*, fine-tunes the whole network together. They found the *chain* method to perform well for sentiment analysis, as individual layers are learned with a reduced risk of overfitting. Since we observed the same pattern in preliminary experiments, we only report results from fine-tuning with the *chain* method.

For our target domain, we further apply a rule-based post-processing step to predict empty tags. Our approach consists in (1) identifying patterns of one DATE or TIME *type* begin-point (identifiable

[5]DistilBERT uncased: https://huggingface.co/distilbert-base-uncased

by tokens such as *from, between, etc.*) and one end-point (*to, and, etc.*.), where no DURATION tag is present, and (2) adding an empty DURATION tag anchored to the begin-point and end-point TEs. For example, in a command, *Set a meeting* FROM *5* TO *6 pm*, the neural model predicts *5* and *6 pm* as two TIME *type* and further post-processing identifies an additional DURATION *type*.

4.2 TE Normalizer

For the normalization task, we propose a rule-based model using a dependency parser sketched in Figure 3 (top). TEs are fed into the parser[6]. Based on the extracted *type* and temporal *unit*, the normalizer identifies a valid normalization pattern (out of 11 expected patterns, cf. Section 5.1) for that *type* and *unit*. For example, given a DATE *type* and a WEEK *unit*, the normalizer expects to find an output pattern of YYYY-WXX. If the pattern predictions from the *type* and *unit* are incompatible (e.g., a DATE *type* with an HOUR *unit*), the normalizer uses the next most probable *unit* from the recognizer model to find as pattern that is compatible with the *unit* (e.g. a TIME *type*). This permits a more robust choice of normalization pattern and reduces the need for iterating over non-relevant rules. After identifying the pattern, each sub-unit in the pattern is normalized sequentially using parsing-based rules. In the case of YYYY-WXX, first the *value* of YEAR and then WEEK is normalized. For every pattern, we define a set of at least four rules: rules for *explicit* TEs (*12th Jan 2020*), *relative* TEs (*tomorrow morning*), *relative with modifier* (*three hours ago*), for *underspecified* TEs (*the 5th*), as well as some pattern-specific rules (e.g. for *weekly*). For each TE, the normalizer iterates over rules for each sub-unit of the pattern. Additionally, we define a gazetteer, containing the values for weekdays, times of the day, etc.

In our domain-specific settings, our normalizer assumes that underspecified expressions (e.g., *June 5*, underspecified year) refer to the past (the previous year's *June 5*) in the news domain and to the future (next year's *June 5*) in the VA domain. This hierarchically-structured rule-based model (which first identifies a pattern and then pattern-specific rules) can easily be adapted to other domains by defining different pattern-specific rules for every type of expression (relative, underspecified, etc.).

[6]We use the SpaCy dependency parser (v.2.3.0): https://spacy.io/api/dependencyparser

5 Experimental Setup

5.1 Data Preprocessing

We perform two data preprocessing steps: sentence simplification and inference of temporal units.

Sentence simplification As mentioned in Section 2.2, the news and VA domains greatly differ with regard to the distribution of sentence length. To reduce this discrepancy, we experiment with a parsing-based[7] text simplification method to preprocess news sentences. For each TE, it extracts the minimal complete sentence containing it (phrase type *S*). For example, in "*Washington said he will argue to save his client's life when the sentencing phase of the trial begins next Wednesday*", the underlined sub-sentence was extracted. This reduces the average length of news domain sentences from 24 to 16.

Temporal unit inference As described above, we need to access the granularity of temporal units as supervision for our model. However, temporal units are not explicitly annotated in TIMEX3: for example, *February* and *2nd week* both have type DATE but not MONTH or WEEK, respectively. However, the unit is reflected in the value pattern (XXXX-02 and XXXX-W06). Thus, we infer the TE's unit from their TimeML *value* fields using the patterns in Table 1. To cover TimeML *values* outside those mentioned in the ISO 8601, we introduce three additional *units*: QUARTER, a sub-unit of YEAR (*first quarter of 2020*); REF, which is used for reference time points (*currently*); and OTHER, which includes a number of infrequent value patterns, values for entities of type SET, and units less relevant for VAs such as century or decade.

5.2 Experiments

First, we train our DA-Time models on the news domain: DA-Time$_1$ (trained with TE-3), DA-Time$_2$ (trained with TE-3 Simplified), DA-Time$_{BL}$ (baseline model trained with TE-3). We split the dataset for our target VA domain, PÂTÉ, into a train/test set with an 80:20 ratio, keeping the class distribution constant between partitions. We perform two experiments[8]: (1) in-domain evaluation of news-trained models on the TE-3 platinum test set (all 3 DA-Time models); (2) out-of-domain evaluation

[7]Stanford CoreNLP parser: https://stanfordnlp.github.io/CoreNLP/

[8](Hyper-)parameters are described in the Appendix.

Model	Training data	$\text{Extent}_{\text{strict}}$	$\text{Extent}_{\text{relax}}$	$\text{Unit}_{\text{relax}}$	$\text{Type}_{\text{relax}}$	$\text{Value}_{\text{relax}}$
HeidelTime	(rule-based)	81.8	90.7	-	83.3	78.1
UW-Time	TBAQ	83.1	91.4	-	**85.4**	**82.4**
DA-Time$_{BL}$	TE-3 (TBAQ+Silver)	81.3±1.3	87.5±1.0	74.0±0.5	74.9±2.3	59.6±1.9
DA-Time$_1$	TE-3 (TBAQ+Silver)	**86.6**±0.4	**91.4**±0.8	78.2±1.5	80.7±2.3	71.7±2.2
DA-Time$_2$	TE-3 Simplified	85.1±0.8	90.0±1.3	77.4±2.7	81.1±2.1	71.3±3.0

Table 3: Experiment 1: F1 Evaluation scores on the news domain (TempEval-3 platinum). DA-Time scores are averages of 5 runs with standard deviations.

of news-trained models on the PÂTÉ test set (DA-Time$_2$, for better comparison with the VA domain, where sentences are shorter). For our second experiment, we compare three settings: (a) direct evaluation of the news model to obtain a lower bound; (b) fine-tuning the news model on PÂTÉ-train and Snips (using Felbo et al. (2017)) and evaluating on PÂTÉ-test to obtain an upper bound; (c), repeating (b) with smaller amounts of VA data (10-100% of PÂTÉ-train with a step size of 10%, i.e., about 50 sentences) to quantify the importance of target domain data. For comparison, we report results for two existing systems, UW-Time and HeidelTime. For news, we report results from the literature, and for PÂTÉ, we evaluate the publicly available UW-Time[9] and HeidelTime[10] systems.

5.3 Evaluation Metrics

We report the F-score metrics from TempEval-3. These include (a) two measures of the overlap between the predicted and gold TE spans (*extent*), computed both in a strict (TEs are exactly matched) and a relaxed condition (TEs are partially matched); and (b) scores for attribute values (*type* and *value*) as well as *unit*. For our own system, scores are reported averages of 5 runs with standard deviations.

6 Results and Analysis

6.1 Experiment 1: In-Domain Evaluation

Table 3 shows results on the TE-3 platinum test set. For extent recognition, DA-Time$_1$ outperforms the other models, as its neural architecture benefits from the large training set. However, we also see that using the noisy silver corpus affects the *type*, and consequently the *value* scores adversely. The best-performing models for *value* scores are the rule-based HeidelTime and UW-Time, which rely on comprehensive domain-specific knowledge. The scores from the DA-Time$_{BL}$ baseline are relatively poor, which is expected here. The extension of the Bi-LSTM and residual layers in the DA-Time$_1$ allows the model to learn task-specific features. The performance of DA-Time$_2$, which uses simplified sentences, is slightly reduced - unsurprisingly, given that the test set is not simplified.

Error analysis. We observe that most errors arise from missing DURATION *type* TEs and from wrong predictions of DATE instead of DURATION. In some cases, mismatches are due to incorrect annotations in the evaluation set (e.g. a TE *2008* is annotated as DURATION but with a value of *2008*). In a few cases, DA-Time falsely predicts modifiers (e.g., *the day before*) as being part of a TE. Such modifiers are handled in the TimeML annotation by tagging them as SIGNAL - however, SIGNAL tags are out of the scope of our current work. Normalization can be further improved by leveraging on the tense of the verbs. Currently, DA-Time is built on the assumption that news texts refer to past events. In several cases the TE is underspecified, but the tense reveals it refers to a future point in time (e.g., *The event will take place on March 15*). Besides, the normalizer of DA-Time is designed to handle TEs in the VA domain. Thus, *units* like decades and centuries cannot be normalized by DA-Time.

6.2 Experiment 2: Cross-Domain Evaluation

Figure 4 shows the results for evaluating DA-Time$_2$ on the PÂTÉ test set without and with fine-tuning on various amounts of PÂTÉ and Snips data. The horizontal lines are for DA-Time$_2$ and literature models without domain adaptation.

As expected, results on PÂTÉ for models without domain adaptation are substantially worse than on the news domain. As the *Extent* and *Type* evaluations show, the strongly data-driven DA-Time$_2$

[9]UW-Time: https://bitbucket.org/kentonl/uwtime/src/master/

[10]HeidelTime (news domain): https://heideltime.ifi.uni-heidelberg.de/

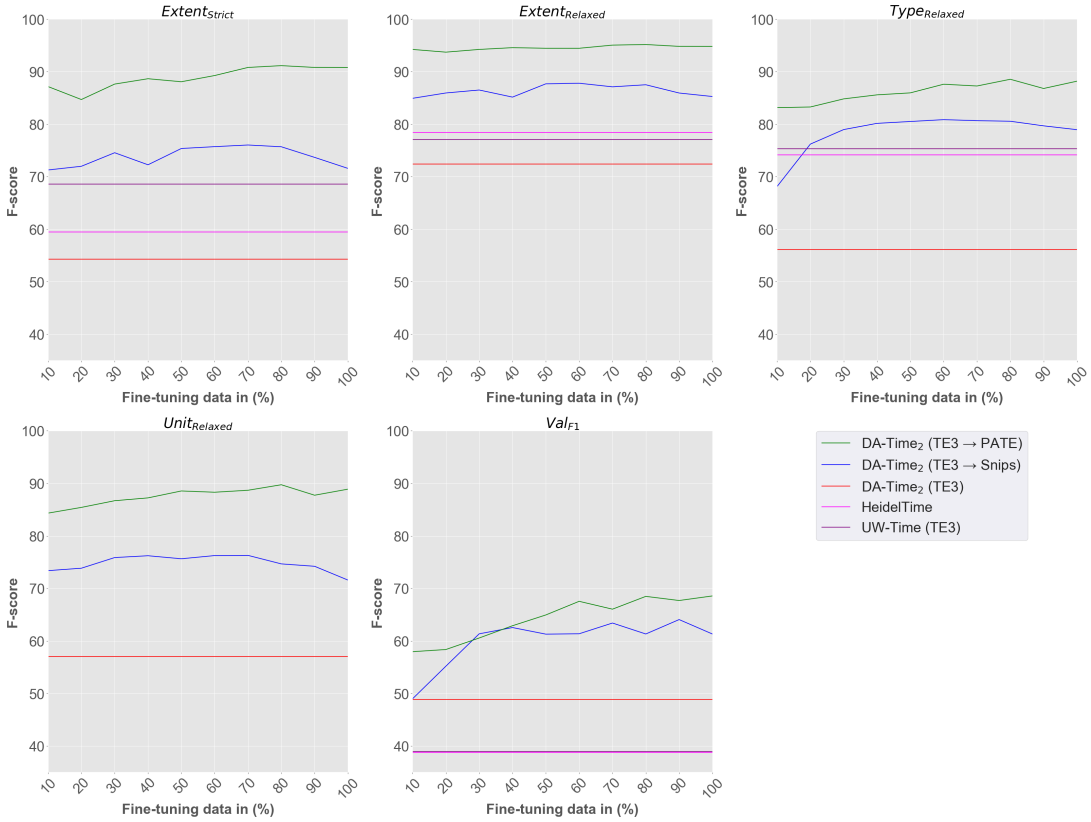


Figure 4: Experiment 2: Evaluation on PÂTÉ-test. X axis indicates the percentage of fine-tuning data used. Scores are an average of 3 runs. Horizontal lines are for models without domain adaptation. The arrows in the legend indicate which datasets were used for training and for fine-tuning, e.g. DA-Time$_2$ (TE3 → Snips) was trained with the TE3 corpus and fine-tuned with the Snips corpus. If only one dataset is indicated, the model was not fine-tuned.

TE recognizer (without fine-tuning - DA-Time$_2$ (TE3) in the figure) performs rather badly compared to HeidelTime and UWTime, presumably due to the changed properties of the input. Nevertheless, it manages to outperform both competitors in the *Value* evaluation, due to the domain-specific TE normalization component. This underlines the importance of domain specific knowledge.

Fine-tuning on Snips (DA-Time$_2$ (TE3 → Snips)) brings about notable improvement for *Extent*, *Type* and *Unit*, which also translate into an improvement for *Value*. However, the improvements flatten out after using ≈ 30% of Snips. We believe that this is due to the differences between Snips and PÂTÉ, even if the two datasets contain data from the same domain.

In comparison, fine-tuning on PÂTÉ (DA-Time$_2$ (TE3 → PÂTÉ)) yields the best results. Strikingly, the biggest jump occurs for just adding 10% of the data or about 25 sentences (strict extent: +30%, relaxed metrics (extent, *type* and *unit*): +≈ 20%, value: +10%). The figures keep improving to some extent with more data, with a final value F1 score of 68% compared to 49% without domain adaptation, and 38% for UW-Time and HeidelTime.

Error analysis. Domain adaptation improves performance in particular on minority classes. Table 4 shows a detailed class breakdown for *type* classification for one run of the model from Section 6.2. Fine-tuning with 10% of the data increases the F-score for the TIME *type* from 0 to 75%, as precision and recall increase by 70% and 79% respectively. The F-score for TIME further increases by 12 extra points after fine-tuning with the full amount of data (75% to 87%): The major difference between news and VA is the difference in class distribution which we have already seen in Figure 2. DURATION *type* expressions, which often contain empty tags and are thus dependent on TIME or DATE *type* TEs, also improve substantially.

Table 4 also shows a corresponding breakdown for *unit* classification. Among the two major *units* (DAY and HOUR), F-score of HOUR *unit* shows an increment of 71 and 80 points when fine-tuning with 10% and 100% of the data respectively. This is expected, as the class distribution difference influ-

Type (freq.)	F-score w/o fine-tuning	Δ after fine-tuning w/ 10% data	Δ after fine-tuning w/ 100% data	Unit (freq.)	F-score w/o fine-tuning	Δ after fine-tuning w/ 10% data	Δ after fine-tuning w/ 100% data
DATE (68)	64.0	+20	+30	DAY (61)	66.0	+9	+26
TIME (48)	0.0	**+75**	**+87**	HOUR (44)	7.0	**+71**	**+80**
DURATION (21)	32.0	+36	**+40**	WEEK (5)	44.0	+0	-4
SET (3)	50.0	+30	+30	MONTH (3)	55.0	-5	+12

Table 4: Per-$type_{\text{relaxed}}$ and per-$unit_{\text{relaxed}}$ evaluation of DA-Time$_2$ on PÂTÉ test: F-scores without fine-tuning (TE3) and Δ after fine-tuning with 10% and 100% of the data (TE3 → PÂTÉ).

enced the *unit* distribution too. Other minor classes are again too infrequent for a reliable analysis.

The rule-based empty tag recognition in DA-Time$_2$ identifies some false positive TEs. This happens when two different TEs are present, which do not denote the beginning and end of an event but rather a change in schedule (BOOK *a schedule from 3 to 5 pm* VS. MOVE *a schedule from 3 to 5 pm*). Domain adaptation however makes a difference compared to out-of-domain scenarios by correctly recognizing a singular numerical token as a TE (*Book a hotel reservation from May 3 to 5* or, *Set a reminder on May 3 at 5*) as they are quite common in the VA domain commands. But this is still a challenge when normalizing multiple TEs without identifying the relations among the TEs (e.g., *Change Star wars 9 from the 25th to the same time on the 24th*). We also find that our parsing-based normalizer provides a particular benefit for handling long TEs (e.g., *the 15th of next month* or *the day before last Tuesday*, etc.).

7 Conclusion

Identifying time expressions (TEs) is a crucial part of the interaction between a voice assistant (VA) and a user, but only small annotated TE corpora exist in the TE domain. In this paper, we have presented DA-Time, a hybrid model combining a neural TE recognizer with a rule-based TE normalizer, and assessed how much data is necessary to fine-tune DA-Time on the VA domain after pre-training on the much better resourced news domain.

We find that our DA-Time model, which performs competitively with the state of the art on news, can be fine-tuned very effectively on the VA domain. While, unsurprisingly, the best performance is achieved with the full target domain training set, already 10% of that dataset – some 25 sentences – is sufficient to achieve major improvements over the news-trained model. Particularly relevant is the improvement on the *Value F1* metric, i.e., the quality of the normalized TEs.

To our best knowledge, this is also the first approach to consider the granularity of temporal *unit* following the TimeML annotation and ISO 8601 standard, and to leverage it to recognize TEs in parallel with TIMEX3 *types* in a parallel setting. TIMEX3 *type* and *unit* are both crucial inputs for our hybrid normalizer. Our normalizer encodes some domain-specific assumptions (e.g., about underspecified TEs). These are particularly important in handling long TEs. While our normalizer is domain-specific, leveraging on temporal units can ease domain adaptation to new domains.

We believe that the small amount of necessary data for fine-tuning is promising for the generalization of temporal taggers for other specific domains. In the future, further improvement may be brought by leveraging anchored time information to identify relations among TEs. Taking into consideration of other TimeML tags (EVENT, SIGNAL) can improve some of the current limitations of the model (for example by identifying event-time relationships or prepositional modifiers). More generally speaking, training temporal taggers in a more end-to-end fashion is a promising direction that appears particularly feasible in the Voice Assistant domain. Considering DA-Time as a baseline model could lead to further neural-based research in the VA domain or for other application domains where identification of temporal information is important.

Acknowledgements

This research was carried out while the first author was affiliated with the Fraunhofer IIS. It was funded in part by the German Federal Ministry for Economic Affairs and Energy (BMWi) through the SPEAKER project (FKZ 01MK19011).

References

Yoshua Bengio. 2011. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, page 17–37. JMLR.org.

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814. Association for Computational Linguistics.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062. Association for Computational Linguistics.

Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. SemEval-2017 task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada. Association for Computational Linguistics.

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Michele Filannino, Gavin Brown, and Goran Nenadic. 2013. ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 53–57. Association for Computational Linguistics.

Lucian Galescu and Nate Blaylock. 2012. A corpus of clinical narratives annotated with temporal information. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 715–720. ACM.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1437–1447, Baltimore, Maryland. Association for Computational Linguistics.

Pawel Mazur and Robert Dale. 2010. Wikiwars: A new corpus for research on temporal expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 913–922.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition LDC2011T07. Web Download. *Linguistic Data Consortium*.

Matthew 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*, pages 2227–2237. Association for Computational Linguistics.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust specification of event and temporal expressions in text.

In *Proceedings of IWCS-5, fifth International Workshop on Computational Semantics*.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, et al. 2003b. The TimeBank corpus. *Corpus Linguistics*, 2003:40.

Xin Rong, Adam Fourney, Robin N Brewer, Meredith Ringel Morris, and Paul N Bennett. 2017. Managing uncertainty in time expressions for virtual assistants. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 568–579. ACM.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *The 7th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC^2)*.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and explicit models of grammar*, pages 181–224.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 541–547. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2016. *Domain-Sensitive Temporal Tagging*. Morgan & Claypool Publishers.

Jannik Strötgen, Julian Zell, and Michael Gertz. 2013. HeidelTime: Tuning English and developing Spanish resources for TempEval-3. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 15–19, Atlanta, Georgia, USA. Association for Computational Linguistics.

Hegler Tissot, Marcos Didonet Del Fabro, Leon Derczynski, and Angus Roberts. 2019. Normalisation of imprecise temporal expressions extracted from text. *Knowledge and Information Systems*, 61(3):1361–1394.

Naushad UzZaman and James F Allen. 2010. Event and temporal expression extraction from raw text: First step towards a temporally aware system. *International Journal of Semantic Computing*, 4(04):487–508.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. Association for Computational Linguistics.

Alessandra Zarcone, Touhidul Alam, and Zahra Kolagar. 2020. PÂTÉ: A corpus of temporal expressions for the in-car voice assistant domain. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 516–523. ELRA.

Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 420–429.

A TE Recognizer

Parameter	**Value**
DA-Time$_1$ input maximum length	50
DA-Time$_2$ input maximum length	30
Batch size	32
Training epochs	30
Fine-tuning epochs	20
Initial learning rate	0.001
Fine-tuning learning rate	0.0001
Bi-LSTM dropout rate	0.5
Bi-LSTM recurrent dropout rate	0.5
DistilBERT dimensions	3072
Recurrent unit	256
Dense layer unit	50
Dense layer activation	ReLu
Optimizer	Adam
Early stopping patience	5
Validation split	0.1

Table 5: Training hyper-parameters for TE Recognizer

Breeding Fillmore's Chickens and Hatching the Eggs: Recombining Frames and Roles in Frame-Semantic Parsing

Gosse Minnema and Malvina Nissim
Center for Language and Cognition
University of Groningen, The Netherlands
{g.f.minnema, m.nissim}@rug.nl

Abstract

Frame-semantic parsers traditionally predict predicates, frames, and semantic roles in a fixed order. This paper explores the 'chicken-or-egg' problem of interdependencies between these components theoretically and practically. We introduce a flexible BERT-based sequence labeling architecture that allows for predicting frames and roles independently from each other or combining them in several ways. Our results show that our setups can approximate more complex traditional models' performance, while allowing for a clearer view of the interdependencies between the pipeline's components, and of how frame and role prediction models make different use of BERT's layers.

1 Introduction

FrameNet (Baker et al., 2003) is a computational framework implementing the theory of frame semantics (Fillmore, 2006). At its core is the notion of *linguistic frames*, which are used both for classifying word senses and defining semantic roles. For example, in (1), "bought" is said to evoke the COMMERCE_BUY frame, and "Chuck", "some eggs", and "yesterday" instantiate its associated roles.

(1) COMMERCE_BUY
[$_{\text{Buyer}}$ Chuck] $_{\odot}$bought [$_{\text{Goods}}$ some eggs]
[$_{\text{Time}}$ yesterday]

In NLP, frame-semantic parsing is the task of automatically analyzing sentences in terms of FrameNet frames and roles. It is a form of semantic role labeling (SRL) which defines semantic roles (called *frame elements*) relative to frames (Gildea and Jurafsky, 2002). Canonically (Baker et al., 2007), frame-semantic parsing has been split up into a three-component pipeline: `targetID` (find frame-evoking predicates), then `frameID` (map each predicate to a frame), and lastly `argID` (given a predicate-frame pair, find and label its arguments). Some recent systems, such as the LSTM-based Open-SESAME and (Swayamdipta et al., 2017) or the classical-statistical SEMAFOR (Das et al., 2014), implement the full pipeline, but with a strong focus specifically on `argID`. Other models implement some subset of the components (Tan, 2007; Hartmann et al., 2017; Yang and Mitchell, 2017; Peng et al., 2018), while still implicitly adopting the pipeline's philosophy.[1] However, little focus has been given to frame-semantic parsing as an *end-to-end* task, which entails not only implementing the separate components of the pipeline, but also looking at their interdependencies.

We highlight such interdependencies from a theoretical perspective, and investigate them empirically. Specifically, we propose a BERT-based (Devlin et al., 2019) sequence labeling system that allows for exploring frame and role prediction independently, sequentially, or jointly. Our results (i) suggest that the traditional pipeline is meaningful but only one of several viable approaches to end-to-end SRL, (ii) highlight the importance of the `frameID` component, and (iii) show that, despite their interdependence, frame and role prediction need different kinds of linguistic information.

Contributions The main contributions of this paper are the following:

- We identify theoretical and practical challenges in the traditional FrameNet SRL pipeline (§2);
- We introduce a flexible, BERT-based sequence-labeling architecture, and experiment with predicting parts of the pipeline separately (§3);
- We explore four methods for re-composing an end-to-end system (§4);

[1] Yang and Mitchell (2017) and Peng et al. (2018) learn frames and arguments jointly, but still need `targetID` as a separate step.

Proceedings of the 14th International Conference on Computational Semantics, pages 155–165
June 17–18, 2021.

- Through two evaluation metrics, we empirically show the relative contribution of the single components and their reciprocal impact (§5-6).

All of our source code and instructions for how to reproduce the experiments is publicly available at https://gitlab.com/gosseminnema/bert-for-framenet.

2 On pipelines, chickens, and eggs

According to Fillmore (2006), an essential feature of a frame is that it is "any system of concepts related in such a way that to understand any one of them you have to understand the whole structure in which it fits." In particular, linguistic frames are systems of semantic roles, possible predicates, and other semantic information. In this section, we discuss the relationship between these concepts in the context of frame-semantic parsing and highlight interdependencies between the various components.

2.1 Challenges for parsers

The following artificial examples display some of the challenges that frame-semantic parsers face:

(2) SELF_MOTION
[$_{\text{Self\_mover}}$ Angela] $_{\odot}$ran [$_{\text{Goal}}$ to school]

(3) FLUIDIC_MOTION
[$_{\text{Fluid}}$ A tear] $_{\odot}$ran [$_{\text{Path}}$ down my cheek]

(4) EXPEND_RESOURCE
[$_{\text{Agent}}$ We] $_{\odot}$ran $_{\odot}$out [$_{\text{Resource}}$ of cookies]

(5) $\emptyset$
His girlfriend ran him home.[2]

In each example, the predicate contains "ran", but used in different frames. In (2) and (3), the predicate is the verb "run", but used in two different senses (running of a person vs. running of a liquid), corresponding to two different frames. Here, the main parsing challenge is resolving this ambiguity and choosing the correct frame (`frameID`). By contrast, in (4), the predicate is "run out". This complex verb is not ambiguous, so the main challenge in this sentence would be `targetID` (i.e. identifying that the target consists of the two tokens "ran" and "out"). Similarly, in (5), "run" is used in a sense not listed in FrameNet, so the challenge here is to make sure nothing is tagged at all.

[2]See sense #14 of "run" in https://www.oxfordlearnersdictionaries.com/definition/english/run_1?q=run

The *roles-make-the-frame* problem In (2-3), given the target ("ran"), the task is to find the correct frame and its corresponding roles. In the traditional pipeline, we would do this by first predicting a frame, and then labeling the dependents of "ran" with roles from this frame. However, the question is what kinds of patterns a frame-finding model needs to learn in order to be successful. It is clearly not sufficient to learn a one-to-one mapping between word forms and frames, not just because of known ambiguous cases ("Angela runs" vs. "a tear runs"), but also because of gaps in FrameNet that conceal unknown ambiguities, such as in (5).

To distinguish between "ran" in (2) and (3), a model has to to take into account the sentential context in some way, which is exactly what LSTM-based models or BERT-based models can do. But what kind of contextual information exactly do we need? SELF_MOTION and FLUIDIC_MOTION have a very similar syntax and semantics, the crucial difference being the semantic category of the "mover". Concretely, this means that in (2-3), we would benefit from recognizing that "Angela" denotes an animate entity while "a tear" denotes a fluid. Doing so would amount to doing partial semantic role labeling, since we are looking at the predicate's syntactic arguments and their semantic properties, which is exactly the information an `argID` model needs to tag "Angela" with "Self_mover" and "a tear" with "Fluid". While it is possible to use contextual information without knowledge of dependency structure (perhaps simple co-occurrence is enough), we hypothesize that such knowledge would be helpful, and thus, that doing `frameID` and `argID` simultaneously, or even predicting `frameID` after `argID`.

The *frames-make-the-targets* problem In the literature, `targetID` has received even less attention than `frameID` — all models we are aware of use gold `targetID` inputs — but is crucial to the success of any end-to-end model. Theoretically speaking, the `targetID` problem is less interesting than `frameID`: since as almost any content word can evoke a frame, assuming a fully complete FrameNet (containing all possible predicates), doing `targetID` would amount to a (simplified) POS-tagging task where content words are labeled as "yes", and (most) function words as "no".

However, in practice, FrameNet is far from complete, so that doing `targetID` means identifying all wordforms that correspond to some pred-

icate evoking a frame present in FrameNet, making `targetID` dependent on `frameID`.[3] For example, to find the target in (2-3), it would suffice to lemmatize "ran" to "run", and check if "run" is listed under any FrameNet frames. But this strategy would fail in (4-5): in those cases, 'ran' is not the full target, but either only a part of it (4), or not at all (5). In order to predict this, we would need to recognize that "run out" is part of the EXPEND_RESOURCE frame, and that "run someone somewhere" is a different sense of "run" that does not match either FLUIDIC_MOTION or SELF_MOTION. Hence, `targetID` seems to presuppose (partial) `frameID` in some cases.

2.2 Pipelines: NLP vs. SRL

The type of problem that we identified in this section is not unique to frame-semantic parsing but also occurs in the standard NLP pipeline of tokenization, POS-tagging, lemmatization, etc. For example, for POS-tagging "run" as either a verb or a noun (as in "we run" vs. "a long run"), one (theoretically speaking) needs access to dependency information (i.e. is there a subject, adjectival modification, etc.). Conversely, dependency parsing benefits from access to POS tags. This would imply that a traditional pipeline might need a lot of redudancy; e.g., a perfect POS-tagging model would also learn some dependency parsing. For (amongst others) this reason, the problem of pipelines versus joint prediction has been extensively studied in NLP in general and SRL in particular. For example, Toutanova et al. (2005) found that predicting all PropBank semantic roles together produced better results than predicting each role separately, Finkel and Manning (2009) proposed a joint model for syntactic parsing and named entity recognition as an alternative to separate prediction or a pipeline-based approach, and He et al. (2018) propose predicting PropBank predicates and semantic roles together instead of sequentially. However, as far as we are aware, no work so far has systematically addressed the frame semantic parsing pipeline and the possible ways for arranging its different components.

In modern NLP, traditional pipelines have largely been replaced by neural models performing several tasks at once. However, a line of work initiated by Tenney et al. (2019); Jawahar et al. (2019) shows

[3] It also makes the task somewhat arbitrary (since it depends on what happens to be annotated in FrameNet), leading some researchers to ignore the problem altogether (Das, 2014).

Input	Sequence labels	Frame structures
Boris	R:Self_mover|R:Sound_source	⟨⟨MAKE_NOISE, ⟨screamed⟩⟩,
screamed	F:Make_noise	⟨⟨Sound_source, ⟨Boris⟩⟩⟩
and		⟨⟨SELF_MOTION, ⟨ran⟩⟩,
ran	F:Self_motion	⟨⟨Self_mover,⟨Boris⟩⟩,
home	R:Direction	⟨Direction, ⟨home⟩⟩⟩

Figure 1: Frame structures and sequence labels (N.B.: color added for illustrative purposes only)

that neural models like BERT implicitly learn to reproduce the classical NLP pipeline, with different layers specializing in specific components of the pipeline, and the possibility for later layers to dynamically resolve ambiguities found in earlier layers. For the BERT-based models we propose, we study the relationship between different layers and the traditional FrameNet pipeline (cf. §6.2).

3 Dissecting the pipeline

We argued that the different components of the frame-semantic parsing task are mutually dependent on each other. Here, we take a more practical view and re-define the parsing problem in a way that allows for experimenting with individual parts of the pipeline and different combinations of them.

3.1 Strip the parser: just sequence labels

For our purposes, a crucial limitation of existing frame-semantic parsing models is that they are relatively complex and not very flexible: the different components have to be executed in a fixed order and depend on each other in a fixed way, leaving no room for experimenting with different orders or alternative ways to combine the components.

By contrast, we propose a maximally flexible architecture by redefining frame-semantic parsing as a sequence labeling task: given a tokenized sentence $S = \langle t_1, \ldots, t_n \rangle$, we predict a frame label sequence $FL = \langle l_1, \ldots, l_n \rangle$, where every $l_i \in (FID \cup \{\emptyset\}) \times 2^{AID}$ is a pair of zero or one frame labels in $FID = \{F_{\text{Abandonment}}, \ldots, F_{\text{Worry}}\}$ and zero or more role labels in $AID = \{A_{\text{Abandonment@Agent}}, \ldots, A_{\text{Worry@Result}}\}$. Note that there can be more than one frame in every sentence, and the spans of different roles can overlap. This is illustrated in Figure 1: *Boris* has two RID labels, each of which is associated to a different frame (Self_mover belongs with SELF_MOTION, while Sound_source belongs to MAKE_NOISE.

This problem definition comprises several simplifications. First of all, we integrate `targetID` and `frameID` into a single component. Moreover,

we 'flatten' the role labels, discarding predicate-role dependency information, and assume that most of this information can be recovered during post-processing (see §5.2). We further simplify the role labels by removing frame names from argument labels, as in $AID' = \{A_{\text{Agent}}, \ldots, A_{\text{Result}}\}$. While this complicates recovering structural information, it also greatly condenses the label space and might improve generalization across frames: many frames share roles with identical names (e.g., Time, Location, or Agent), which we assume are likely to share at least some semantic properties. It should be noted that this assumption is not trivial, given that there is a long and controversial literature on the generalizability of semantic (proto-)roles (Reisinger et al., 2015); we will make it here nonetheless, especially since initial experiments on the development set showed a clear advantage of removing frame names from argument labels.

We implement our architecture using a BERT-based sequence labeler: given a sentence, we tokenize it into byte-pairs, compute BERT embeddings for every token, feed these (one-by-one) to a simple feed-forward neural network, and predict a label representation. By having BERT handle all preprocessing, we avoid making design choices (e.g. POS-tagging, dependency parsing) that can have a large impact on performance (cf. Kabbach et al., 2018), and make our approach easier to adapt to other languages and datasets.

3.2 Strip the tasks: just frames, just roles

Having maximally 'stripped down' the architecture of our parsing model, we can now define the two most basic tasks: frame prediction (equivalent to `targetID` plus `frameID` in the traditional pipeline), or role prediction (equivalent to `argID`, but without needing frames as input). We can then perform the tasks separately, but also jointly, or combine them in any desired way.

FRAMESONLY The first basic task is predicting, given a token, whether this token 'evokes' a FrameNet frame, and if so, which one. We experiment with two types of label representation settings: **Sparse**, which represents each frame (and the empty symbol) as a one-hot vector, while **Embedding** defines dense embeddings for frames. The embedding of a frame F is defined as the centroid of the embeddings of all predicates in F, which in turn are taken from a pre-trained GloVe model (Pennington et al., 2014).[4] This is very similar to the approach taken by Alhoshan et al. (2019). Prediction is done by regressing to the frame embedding space, and selecting the frame with the smallest cosine distance to the predicted embedding. The empty symbol is predicted if the cosine similarity to the best frame is below a threshold t_f.

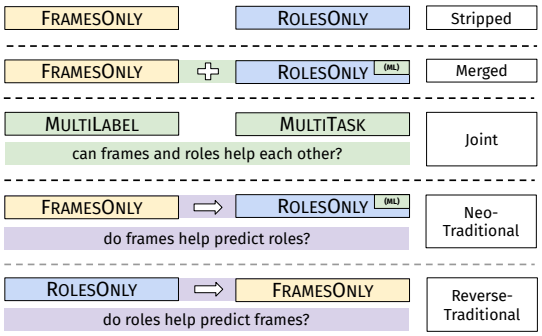


Figure 2: Overview of possible end-to-end systems *(N.B.: boxes marked with (ML) use role predictions from* MULTILABEL*)*

ROLESONLY The other basic task predicts zero or more bare role labels for every token in the input. These labels are encoded in a binary vector that represents which roles are active for a given token. During decoding, tokens with an activation value exceeding a threshold t_r are kept as the final output.

4 Re-composing an end-to-end system

Having defined a basic setup for experimenting with predicting frames and roles alone, we can now design experiments for investigating any interactions between frames and roles. Figure 2 provides an overview of the possible ways for combining the FRAMESONLY and ROLESONLY models: simply merging the outputs, predicting the two tasks jointly, or using a sequential pipeline.

4.1 Do-it-together: multilabel or multitask

Given the overlap between the frame and role prediction tasks, we test whether predicting frames and roles jointly might help the two models mutually inform each other and learn more efficiently.

Joint(MULTILABEL) The first 'joint' approach is to predict, for every token in the input, a binary vector representing any frame target, as well as any role labels carried by the token. Hence, there is only one decoder and all parameters are shared.

[4] We use the model `glove.42B.300d` from https://nlp.stanford.edu/projects/glove/.

Joint(MULTITASK) As an alternative, we also try a setup with separate decoders for roles and frames, without any shared parameters (except for the BERT encoder). Backpropagation is done based on a weighted sum of the losses of the two decoders, where the 'loss weights' are learned.

4.2 Do-it-sequentially: what comes first?

Neo-traditional In the *Neo-traditional* experiment, we test the traditional pipeline structure: i.e., learning frames first, and using these to (explicitly) inform role learning. In order to do this, we make two modifications to ROLESONLY: 1) we split the target role labels by frame, i.e. we ask the model to predict only one frame's roles at any given time, and 2) a representation of the 'active' frame is concatenated to the BERT embeddings as input to the model. This representation could be either **Sparse** or **Embedding** (see above). After role prediction, any roles that did not match the frame inputs are filtered out, and the predictions are merged with the frame model's output.

Neo-traditional+MULTILABEL Following preliminary results, we repeat the experiment using MULTILABEL instead of ROLESONLY. In a final merging step, we keep all role predictions from MULTILABEL and any frame predictions that do not clash with the outputs of FRAMESONLY.

Reverse-traditional In this setup, we invert the traditional pipeline: given a sentence, we first predict role labels (using ROLESONLY), which are then used as input for the frame prediction model.[5]

4.3 Do-it-separately: copy-and-paste

Finally, we tried an approach assuming no interaction between frame and role prediction at all.

Merged In the *Merged* experiment, we simply merge the outputs of FRAMESONLY and ROLESONLY. In this scenario, both models are completely independent, without any possibility for frames and roles to inform each other.

Merged+MULTILABEL Based on initial results showing that MULTILABEL beats ROLESONLY on roles while FRAMESONLY wins on frames, we also experiment with simply merging the output of these two 'winning' models.

[5]Other setups, e.g. using MULTILABEL for role predictions, might give better performance, but would obfuscate the effect of predicting roles before frames.

5 Evaluation: tokens vs. structures

Since our setup diverges significantly from previous systems, testing our models is not trivial. Here, we propose two evaluation methods: a token-based metric that can directly score our models' output (§5.1), and an algorithm for 'recovering' full frame structures that can be checked using the standard SemEval 2007 method (Baker et al., 2007) (§5.2).

5.1 Sequence-label evaluation

The simplest way of evaluating our models' performance is to simply count the number of correct frame and role labels per token. We compute this given a token sequence $\langle t_1, \ldots, t_n \rangle$, a sequence of gold labels $\langle G_1, \ldots, G_n \rangle$ and a sequence of predicted labels $\langle P_1, \ldots, P_n \rangle$, where every G_i and P_i is either a set of frame or role labels, or the empty label $\{\emptyset\}$. We can now define: $\textit{true\_positive} = \sum_{i=1}^{n} |P_i \cap G_i|$, $\textit{false\_positive} = \sum_{i=1}^{n} |P_i \setminus G_i|$, and $\textit{false\_negative} = \sum_{i=1}^{n} |G_i \setminus P_i|$. Finally, we calculate micro-averaged precision, recall, and F-scores in the usual way.

Consistency scoring A limitation of our sequence-labeling approach is that there are no explicit constraints on the predicted role labels and it is not guaranteed that the set of role predictions for a given sentence will be compatible with the set of frame predictions. Hence, we need to evaluate not just the respective accuracy, but also the mutual consistency, of predicted roles and frames. We define this as $\sum_{s \in S} \sum_{t \in \textit{tok}(s)} |\{r \in R_{s,t} | r \in \textit{allowed}(F_s)\}|$, where S is the set of sentences in the evaluation set, $\textit{tok}(s)$ returns the sequence of tokens in a sentence, $R_{s,t}$ is the set of predicted role labels for a particular token, F_s is the set of all predicted frame labels in the sentence, and $\textit{allowed}(F)$ returns the set of role labels that are consistent with a particular set of frame labels. For example, $\textit{allowed}(\{\text{KILLING}, \text{USING}\})$ gives $\{\text{Killer}, \text{Victim}, \ldots, \text{Agent}, \ldots\}$. The number of consistent roles is then divided by the total number of predicted roles $\sum_s \sum_t |r \in R_{s,t}|$ to yield a global consistency score.

5.2 Recovering frame structures

For comparing our models to existing work in frame-semantic parsing, and validating the assumptions underlying our sequence-labeling setup, we need to recover full frame structures from the output of our models. Formally,

Experiment	DEV						TEST					
	frames			roles			frames			roles		
	R	P	F	R	P	F	R	P	F	R	P	F
Open-SESAME	*0.66*	*0.68*	*0.67*	*0.41*	*0.54*	*0.47*	*0.58*	*0.62*	*0.60*	*0.38*	*0.41*	*0.39*
Joint(MULTILABEL)	0.58	0.65	0.61	0.39	0.48	0.43	0.55	0.44	0.49	0.36	0.27	0.31
Joint(MULTITASK)	0.69	0.73	0.71	0.24	0.35	0.28	0.65	0.49	0.56	0.24	0.21	0.21
Neo-traditional(MULTILABEL)*†	0.66	0.73	0.69	0.41	0.54	0.47	0.64	0.51	0.57	0.40	0.30	0.34
Reverse-traditional(ROLESONLY)‡	0.68	0.72	0.70	0.32	0.46	0.38	0.65	0.49	0.56	0.31	0.27	0.28
Merged(MULTILABEL)*	0.72	0.68	0.70	0.39	0.49	0.43	0.69	0.48	0.57	0.36	0.28	0.31
Stripped(FRAMESONLY, Embedding)	0.68	0.69	0.69	-	-	-	0.65	0.46	0.54	-	-	-
Stripped(FRAMESONLY, Sparse)	0.65	0.75	0.70	-	-	-	0.63	0.52	0.57	-	-	-
Stripped(ROLESONLY)	-	-	-	0.32	0.46	0.38	-	-	-	0.31	0.27	0.28

Table 1: Sequence labeling scores (avg. over three runs). *N.B: *For brevity reasons, for Merged and Neo-traditional, we only give results for the* MULTILABEL *setting, which performs better on role prediction than* ROLESONLY. † *Results on Neo-traditional are using Sparse frame inputs.* ‡ *Results on Reverse-traditional are using Sparse frame outputs.*

given a tokenized sentence $\langle t_1, \ldots, t_n \rangle$, and a sequence $\langle l_1, \ldots l_n \rangle$ of frame and role labels, we want to find the set of frame structures $\{\langle TI_1, RI_1 \rangle, \ldots, \langle TI_n, RI_n \rangle\}$. Here, every target instance $TI_i = \langle FT_i, \langle t_j, \ldots, t_k \rangle \rangle$ is a pairing of a frame type $FT \in \{F_{\text{Abandonment}}, \ldots F_{\text{Worry}}\}$ and a sequence of tokens containing the lexical material that evokes the frame. Similarly, we define every role instance $RI_i = \{\langle RT_{i_1}, \langle t_{j_1}, \ldots t_{k_1} \rangle \rangle, \ldots, \langle RT_{i_n}, \langle t_{j_n}, \ldots, t_{k_n} \rangle \rangle\}$ as a set of pairs of role types $RT \in \{A_{\text{Abandonment@Agent}}, \ldots, A_{\text{Worry@Result}}\}$ and token spans instantiating these role types. See Figure 1 (§3) for an example sentence with sequence labels and corresponding frame structures.

Recovery algorithm We propose a simple rule-based approach. First, we find the set of target instances in the sentence, and the corresponding set of frame types.[6] Next, we find the set of (bare) role labels that can be associated to each of the predicted frame types, e.g. WORRY $\mapsto$ {Experiencer, . . . , Result}. Next, for each of the the predicted role spans $\langle t_i, \ldots, t_j \rangle$ in the sentence, we find all of the compatible frame target instances. If there is more than one compatible target, we select the target that is closest in the sentence to the role span. Note that our algorithm would miss cases of more than one frame instance 'sharing' a role (i.e. all having a role with the same label and span), but we assume that such cases are rare. In cases where it is already known which role labels are associated to which frame types (i.e., in the *Neo-traditional* setup), we allow the algorithm to take this information into account, but we found that this has little impact on performance.

SemEval'07 scoring Having recovered the set of predicted frame structures, we can evaluate our models using the standard SemEval 2007 scoring method (Baker et al., 2007). During evaluation on the development set, we noticed that our models frequently seem to make minor mistakes on role spans (i.e. erroneously missing or including an extra token). Since the SemEval script does not take into account partially matching role spans, we propose a modification to the script that gives partial credit for these role spans, and report this in addition to the scores from the original script.

6 Experiments

6.1 Setups

All experiments were run using a pre-trained `bert-base-cased` model, fine-tuned with a simple feedforward network decoder. Loss functions depend on the setup: we optimize Mean Squared Error Loss for ROLESONLY and MULTILABEL, Sequence Cross-Entropy Loss for FRAMESONLY/Sparse, and Cosine Embedding Loss for FRAMESONLY/Embedding. We found best performance using Adam optimization (Kingma and Ba, 2014) with $lr = 5e^{-5}$, training for 12 epochs, with a single hidden layer of size 1000 in the decoder. Unless specified otherwise, the BERT embeddings are an automatically weighted sum ("Scalar Mix") of BERT's hidden layers. For implementation, we used AllenNLP (Gardner et al., 2017) and PyTorch (Paszke et al.,

[6] If more than one frame target label is predicted for a given token, we only keep the label with the highest probability.

Experiment	DEV						TEST					
	strict			modified			strict			modified		
	R	P	F	R	P	F	R	P	F	R	P	F
Open-SESAME (true)	*0.47*	*0.52*	*0.49*	*0.51*	*0.57*	*0.54*	*0.44*	*0.50*	*0.47*	*0.47*	*0.53*	*0.50*
Open-SESAME (recovered)	*0.46*	*0.51*	*0.48*	*0.50*	*0.56*	*0.53*	*0.43*	*0.49*	*0.46*	*0.46*	*0.53*	*0.49*
Joint(MULTILABEL)	0.31	0.40	0.35	0.37	0.48	0.42	0.32	0.42	0.36	0.36	0.48	0.41
Joint(MULTITASK)	0.34	0.44	0.38	0.38	0.50	0.43	0.36	0.43	0.39	0.40	0.48	0.43
Neo-traditional(MULTILABEL)*†	0.33	0.38	0.35	0.42	0.49	0.45	0.34	0.37	0.35	0.42	0.46	0.44
Reverse-traditional(ROLESONLY)‡	0.33	0.48	0.39	0.38	0.55	0.45	0.34	0.47	0.40	0.38	0.53	0.45
Merged(MULTILABEL)*	0.37	0.44	0.40	0.44	0.52	0.47	0.40	0.44	0.42	0.45	0.50	0.47

Table 2: SemEval'07 scores (avg. over three runs)

2019). All models were trained and tested on the standard FrameNet 1.7 fulltext corpus (see Appendix B for more details on the data).

While our main aim remains a deeper understanding of the components of frame-semantic parsing and their interdependencies, we still need to put our scores into perspective and legitimize our sequence labeling approach. Thus, we took Open-SESAME, the only existing, open-source model that we are aware of that is capable of producing end-to-end predictions, as our baseline.[7] We used default settings (i.e., without scaffolding and ensembling) for better comparability with our own models. Hence, note that the results reported here for Open-SESAME are *not* the state-of-the-art results reported by Swayamdipta et al. (2017).

6.2 Results

Sequence labeling Table 1 reports the results on the sequence labeling task.[8] On the test set, Open-SESAME is the best model for both tasks. While best performance is not the core goal of this work, the fact that our best models perform in a similar range shows that our setup is sound to serve as a tool for comparing different pipeline variations.

Comparing our own models, we see that frame prediction performance is similar across setups: except for MULTILABEL, all F1-scores are within 3 points of each other. On role prediction, the setups that use MULTILABEL outperform the others. *Neo-traditional* performs the best on roles overall, whereas MULTITASK scores the worst. For frame prediction, performance does not seem to be boosted by joint role prediction. In fact, in MULTILABEL, performance on frames is very poor.

Similarly, adding roles as input for frame predictions (as in *Reverse-traditional*) does not help performance. Additional experiments to test the theoretical effectiveness of this strategy, using gold role labels as input, showed a slight improvement over FRAMESONLY (increasing F1 to 0.58 on test). However, when using predicted roles, we find no improvement and even see a small detrimental effect due to the poor performance of ROLESONLY. By contrast, *Neo-traditional* and *Merged*, when combining FRAMESONLY and MULTILABEL, perform well on both frames and roles. Lastly, MULTITASK does well on frames (but only slightly better than FRAMESONLY), but very poorly on roles.

Structural evaluation SemEval'07 scores are shown in Table 2. Note that two separate scores are reported for Open-SESAME: "true" and "recovered". For "true", we converted Open-SESAME predictions to SemEval format using all available structural information (i.e., links between roles, frames, and predicates); for "recovered", we first removed structural information and then attempted to recover it using our algorithm (see §5.2). The small difference between these scores suggests that recovery usually succeeds.

In any case, Open-SESAME consistently outperforms our models, and the difference is, overall, larger on the SemEval task than on the sequence labeling task. On the test set, *Merged* is our best model and has an F1-score within 0.05 of Open-SESAME using strict evaluation, and within 0.03 using partial span scoring. Interestingly, whereas the sequence-labeling performance of all models drops dramatically on the test set compared to the development set, SemEval task scores are more sta-

[7] Note that the Open-SESAME paper only treats `argID`, but models published at `https://github.com/swabhs/open-sesame` use the same architecture for doing `targetID` and `frameID` and are discussed at `https://github.com/swabhs/coling18tutorial`.

[8] For checking stability, all experiments were repeated three times and the scores averaged across runs. Overall, the models were quite stable and have F1-scores with standard deviations of ≤ 0.03. See the Appendix for full stability scores.

	Stripped						MULTILABEL					
	frames			roles			frames			roles		
	R	P	F	R	P	F	R	P	F	R	P	F
L02	0.65	0.72	0.68	0.36	0.43	0.39	0.55	0.64	0.59	0.34	0.45	0.39
L04	0.66	0.75	0.70	0.36	0.52	0.43	0.60	0.63	0.62	0.38	0.46	0.42
L06	0.62	0.77	0.69	0.39	0.47	0.42	0.59	0.67	0.62	0.39	0.58	0.47
L08	0.68	0.73	0.71	0.42	0.55	0.47	0.56	0.66	0.61	0.42	0.54	0.47
L10	0.71	0.72	0.71	0.45	0.52	0.48	0.56	0.62	0.59	0.47	0.55	0.51
L12	0.68	0.74	0.71	0.46	0.56	0.51	0.58	0.70	0.63	0.46	0.60	0.52
Mix	0.65	0.75	0.72	0.32	0.46	0.38	0.58	0.65	0.61	0.39	0.48	0.43

Table 3: Sequence-label scores (DEV) by BERT layer

ble. Finally, as expected, both Open-SESAME and our models get higher scores when partial credit is given to incomplete role spans, but our models benefit more from this than Open-SESAME does.

Out of our own models, *Merged* clearly wins, with a five points' difference to MULTILABEL, the worst-scoring model. A possible explanation for this difference is that MULTILABEL has poor recall for frame prediction: since frame structures always need to have a frame target, missing many frames is likely to cause low SemEval scores. However, good frames are not enough: while *Merged* beats Open-SESAME on frames on the development set, it has lower SemEval scores. More generally, it is interesting to note that good sequence labeling scores do not guarantee good SemEval performance. On one hand, we find that *Reverse-traditional* has good SemEval scores, especially for precision, even though it has poor sequence labeling scores on roles. On the other hand, *Neo-traditional* has good sequence labeling scores, but disappointing SemEval scores.

Consistency A factor that would be expected to lead to better SemEval scores is consistency between role and frame prediction: predicting many correct frames, but also many roles inconsistent with these frames, might lead to overall worse structures. Table 4 gives consistency scores (see §5.1) for all setups except *Stripped*. Open-SESAME and *Neo-traditional* score perfectly because frames are known at role prediction time, so that inconsistent roles are filtered out. There are large differences between the other setups: *Merged* has nearly 80% 'legal' roles, whereas *Joint*(MULTITASK) scores only 62%. Moreover, *Merged* outperforms MULTILABEL, despite getting its roles from MULTILABEL. We speculate that this is caused by MULTILABEL predicting 'orphaned' roles (i.e., correct roles lacking a matching frame) that are 're-parented' in *Merged*, which adds 'extra' frames from FRAMESONLY. Finally, *Reverse-traditional*'s consistency is lower than would be expected given that frame prediction is constrained by information about roles, which we attribute to poor role prediction in ROLESONLY. Still, *Reverse-traditional* performs quite well on SemEval, meaning that role coherence alone does not predict structural quality.

BERT layer analysis Analyzing the contributions of different BERT layers helps us better understand the implicit 'pipeline' learned by the model. Table 3 shows sequence labeling scores for the *Stripped* and MULTILABEL models, retrained using embeddings from individual layers. For comparison, the last row shows scores from Table 1.

We see an interesting discrepancy between frames and roles: role prediction clearly improves when using higher layers, but frame prediction is mostly stable, suggesting that the latter benefits from lexical information more than the former. This is true for both the Stripped and MULTILABEL models. Another interesting pattern is that role prediction is better for individual layers than for the "ScalarMix" setup, whereas this is not the case for frame prediction. This means that it is difficult to learn automatically which layers to use for role prediction, but it is yet unclear why.

7 Conclusions

We examined the frame-semantic parsing pipeline theoretically and practically, identifying 'chicken-or-egg' issues in the dependencies between subtasks, and studying them empirically within a BERT-based sequence-labeling framework.

We found that joint frame and role prediction works well, but not always better than using frames as input to roles. By contrast, previous studies (Yang and Mitchell, 2017; Peng et al., 2018) found substantial improvements from joint prediction. However, these systems use gold targets as input,

Experiment	DEV		TEST	
	score	stdev	score	stdev
Open-SESAME	*1.00*	*0.00*	*1.00*	*0.00*
Joint(MULTILABEL)	0.77	0.01	0.74	0.02
Joint(MULTITASK)	0.62	0.05	0.62	0.06
Neo-traditional(MULTILABEL)*†	1.00	0.00	1.00	0.00
Reverse-traditional(ROLESONLY)*†	0.71	0.06	0.70	0.04
Merged(MULTILABEL)*	0.80	0.01	0.78	0.02

Table 4: Consistency scores and deviance across runs

differ in architecture, and (partially) use different datasets, making direct comparison hard.

The main advantage of our sequence-labeling setup is the possibility to investigate frame and role prediction independently, as well as their mutual dependency. We found substantial benefits for role prediction from access to frame information through joint prediction or by receiving frames as input. For frame prediction, instead, the picture is less clear: while we found a theoretical benefit of using (gold) roles as input, this benefit disappears when using predicted roles. Similarly, when jointly predicting frames and roles, the MULTITASK setup yielded a slight improvement for frame prediction, whereas MULTILABEL deteriorated it. These results can be taken as supporting the traditional pipeline approach, but our results using SemEval evaluation, which looks at full frame structures, do not unequivocally confirm this: Open-SESAME performs best, but amongst our models, *Reverse-traditional* and *Merged* outperform the others, including *Neo-traditional*. This suggests that there might be valid alternatives to the standard pipeline, and exploring these might lead to a deeper understanding of frame semantic parsing task itself.

Our setup also allows for investigating which BERT layers both components use. Role prediction strongly prefers high BERT layers, while frame prediction is less picky, suggesting that the tasks use different linguistic information.

We see several logical extensions of our work. First, qualitative analysis of the overlaps in predictions from different models could shed light on the discrepancies between sequence labeling scores, consistency scores, and SemEval scores. A second direction would be to explore how our observations about the relationship between different components of the frame semantic parsing pipeline and BERT layers could be used to improve models. Finally, one could try more sophisticated architectures for sequence-labeling models, in particular by enforcing frame-role consistency within the model itself rather than during post-processing.

Acknowledgements

The research reported in this article was funded by the Dutch National Science organisation (NWO) through the project *Framing situations in the Dutch language*, VC.GW17.083/6215. We would also like to thank our anonymous reviewers for their helpful comments.

References

Waad Alhoshan, Riza Batista-Navarro, and Liping Zhao. 2019. Semantic frame embeddings for detecting relations between software requirements. In *Proceedings of the 13th International Conference on Computational Semantics - Student Papers*, pages 44–51, Gothenburg, Sweden. Association for Computational Linguistics.

Collin Baker, Michael Ellsworth, and Katrin Erk. 2007. SemEval-2007 task 19: Frame semantic structure extraction. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 99–104, Prague, Czech Republic. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and Beau Cronin. 2003. The structure of the FrameNet database. *International Journal of Lexicography*, 16(3):281–296.

Dipanjan Das. 2014. Statistical models for frame-semantic parsing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 26–29, Baltimore, MD, USA. Association for Computational Linguistics.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

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.

C. Fillmore. 2006. Frame semantics. In D. Geeraerts, editor, *Cognitive Linguistics: Basic Readings*, pages 373–400. De Gruyter Mouton, Berlin, Boston. Originally published in 1982.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334, Boulder, Colorado. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. 2017. Out-of-domain FrameNet semantic role labeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 471–482, Valencia, Spain. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. 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.

Alexandre Kabbach, Corentin Ribeyre, and Aurélie Herbelot. 2018. Butterfly effects in frame semantic parsing: impact of data processing on model ranking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3158–3169, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary 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. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. 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 1492–1502, New Orleans, Louisiana. 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.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *CoRR*, abs/1706.09528.

Songbo Tan. 2007. Using error-correcting output codes with model-refinement to boost centroid text classifier. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 81–84, Prague, Czech Republic. Association for Computational Linguistics.

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.

Kristina Toutanova, Aria Haghighi, and Christopher Manning. 2005. Joint learning improves semantic role labeling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 589–596, Ann Arbor, Michigan. 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.

A Supplemental material

A.1 Model stability

In order to check for model stability, we repeated all our experiments (excl. the experiments for BERT layer analysis) three times. The standard deviations of the sequence-labeling scores are shown in Table A.1. F1-scores seem quite stable overall, but in a few cases there are larger deviations in precision and/or recall, especially in the *Joint*(MULTITASK) model.

Experiment	frames R	frames P	frames F	roles R	roles P	roles F
Joint(MULTILABEL)	01	00	00	00	01	01
Joint(MULTITASK)	02	02	00	04	07	01
Neotrad.(MULTILABEL, Sp.)	02	01	01	02	04	01
Merged(MULTILABEL)	01	01	00	00	01	01
Stripped(FRAMESONLY, Emb.)	02	04	01	-	-	-
Stripped(FRAMESONLY, Sp.)	03	02	01	-	-	-
Stripped(ROLESONLY)	-	-	-	02	04	03

Table A.1: Model stability: standard deviation (%) across runs of sequence-labeling scores (on DEV)

A.2 FrameNet data

Corpus We used the standard FrameNet corpus (release 1.7) for all experiments. We used the `fulltext.train` split for training, the `dev` split for validation and evaluation, and the `test` split for final evaluation. Table A.2 shows the relative sizes of these splits.

Distribution of roles One of the key simplifications of our sequence labeling setup is 'decoupling' frames and roles. This reduces the label space since some roles occur in many different frames. Figure A.1 shows the most frequent role names with the number of different frames that they occur in. As can be seen from the graph, most frequent roles are very general ones such as 'Time', 'Place', 'Manner', etc. Although roles are, in the FrameNet philosophy, strictly defined relative to frames, we expect that roles sharing a name across frames will have a very similar semantics.

Split	#Sentences	#Frame structures
`train`	3,413	19,391
`dev`	387	2,272
`test`	2,420	6,714

Table A.2: FrameNet corpus stats

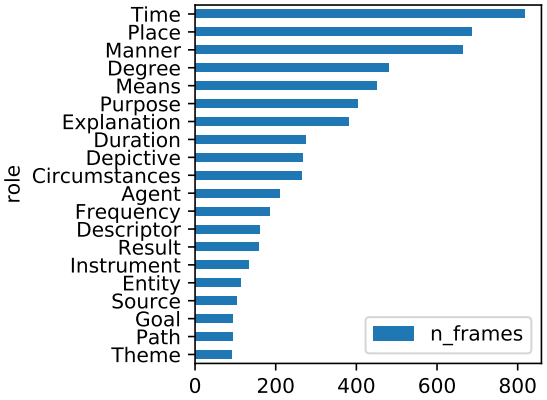


Figure A.1: Top-20 role names by number of frames

Large-scale text pre-training helps with dialogue act recognition, but not without fine-tuning

Bill Noble and **Vladislav Maraev**
Centre for Linguistic Theory and Studies in Probability
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
{bill.noble,vladislav.maraev}@gu.se

Abstract

We use dialogue act recognition (DAR) to investigate how well BERT represents utterances in dialogue, and how fine-tuning and large-scale pre-training contribute to its performance. We find that while both the standard BERT pre-training and pretraining on dialogue-like data are useful, task-specific fine-tuning is essential for good performance.

Large-scale neural language models trained on massive corpora of text data have achieved state-of-the-art results on a variety of traditional NLP tasks. Given that dialogue, especially spoken dialogue, is radically different from the kind of data these language models are pre-trained on, it is uncertain whether they would be useful for dialogue-oriented tasks. In the example from the Switchboard corpus, shown in Table 1, it is evident that the structure of dialogue is quite different from that of written text. Not only is the internal structure of contributions different—with features such as disfluencies, repair, incomplete sentences, and various vocal sounds—but the sequential structure of the discourse is different as well.

In this paper, we investigate how well one such large-scale language model, BERT (Devlin et al., 2019), represents utterances in dialogue. We use dialogue act recognition (DAR) as a proxy task, since both the internal content and the sequential structure of utterances has bearing on this task

We have two main contributions. First we find that while standard BERT pre-training is useful, the model performs poorly without fine-tuning (§3.1). Second, we find that further pre-training with data from the target domain shows promise for dialogue, but the results are mixed when pre-training with a larger corpus of dialogical data from outside the target domain (§3.2).

Speaker	DA	Utterance
A	sd	Well, I'm the kind of cook that I don't normally measure things,
A	sd	I just kind of throw them in
A	sd	and, you know, I don't to the point of, you know, measuring down to the exact amount that they say.
B	sv	That means you're a real cook.
A	bd	<Laughter> Oh, is that what it means.
A	b	Uh-huh.
A	x	<Laughter>.

Table 1: Example from the SWDA corpus (sw2827). Dialogue acts: *sd*—Statement-non-opinion, *sv*—Statement-opinion, *bd*—Downplayer, *b*—Backchannel, *x*—Non-verbal.

1 Background

1.1 Dialogue Act Recognition

The concept of a dialogue act is based on that of speech acts (Austin and Urmson, 2009). Breaking with classical semantic theory, speech act theory considers not only the propositional content of an utterance but also the actions, such as *promising* or *apologizing*, it carries out. Dialogue acts extend the concept of the speech act, with a focus on the interactional nature of most speech.

DAR is the task of labeling utterances with the dialogue act they perform from a given set of dialogue act tags. As with other sequence labeling tasks in NLP, some notion of context is helpful in DAR. One of the first performant machine learning models for DAR was a Hidden Markov Model that used various lexical and prosodic features as input (Stolcke et al., 2000). Most successful neural approaches also model some notion of context (e.g., Kalchbrenner and Blunsom, 2013; Tran et al., 2017a; Bothe et al., 2018b,a; Zhao and Kawahara, 2018).

Proceedings of the 14th International Conference on Computational Semantics, pages 166–172
June 17–18, 2021.

1.2 Transfer learning for NLP

Transfer learning techniques allow a model trained on one task—often unsupervised—to be applied to another. Since annotating natural language data is expensive, there is a lot of interest in transfer learning for natural language processing. Word vectors (e.g., Mikolov et al., 2013; Pennington et al., 2014) are a ubiquitous example of transfer learning in NLP. We note, however, that pre-trained word vectors are not always useful when applied to dialogue (Cerisara et al., 2017).

BERT, a multi-layer transformer model (Devlin et al., 2019), is pre-trained on two unsupervised tasks: *masked token prediction* and *next sentence prediction*. In masked token prediction, some percentage of words are randomly replaced with a mask token. The model is trained to predict the identity of the original token based on the context sentence. In next sentence prediction, the model is given two sentences and trained to predict whether the second sentence follows the first in the original text or if it was randomly chosen from elsewhere in the corpus. After pre-training, BERT can be applied to a supervised task by adding additional un-trained layers that take the hidden state of one or more of BERT's layers as input.

There is some previous work applying BERT to dialogue. Bao et al. (2020) and Chen et al. (2019) both use BERT for dialogue generation tasks. Similarly, Vig and Ramea (2019) find BERT useful for selecting a response from a list of candidate responses in a dialogue. Mehri et al. (2019) evaluate BERT in various dialogue tasks including DAR, and find that a model incorporating BERT outperforms a baseline model. Finally, Chakravarty et al. (2019) use BERT for dialogue act classification for a proprietary domain and achieves promising results, and Ribeiro et al. (2019) surpass the previous state-of-the-art on generic dialogue act recognition for Switchboard and MRDA corpora. This paper aims to supplement the findings of previous work by investigating how much of BERT's success for dialogue tasks is due to its extensive pre-training and how much is due to task-specific fine-tuning.

Fine-tuning vs. further in-domain pre-training We experiment with the following two transfer learning strategies (Sun et al., 2019): *further pre-training*, in which the model is trained in an un-supervised way, similar to its initial training scheme, but on data that is in-domain for the target task; and *single-task fine-tuning*, in which the model's encoder layers are optimized during training for the target task.

Whether or not the encoder model has undergone further in-domain pre-training, there remains a choice of whether to fine-tune during task training, or simply extract features from the encoder model without training it (i.e., *freezing*). Freezing the encoder model is more efficient, since the gradient of the loss function need only be computed for the task-specific layers. However, fine-tuning can lead to better performance since the encoding itself is adapted to the target task and domain.

Peters et al. (2019) investigate when it is best to fine-tune BERT for sentence classification tasks and find that when the target task is very similar to the pre-training task, fine-tuning provides less of a performance boost. We note that there is some conceptual relationship between DAR and next sentence prediction, since the dialogue act constrains (or at least is predictive of) the dialogue act that follows it. That said, the discourse strucutre of the encyclopedia and book data that makes up BERT's pre-training corpus is probably quite different from that of natural dialogue.

Switchboard	**AMI Corpus**
Dyadic	Multi-party
Casual conversation	Mock business meeting
Telephone	In-person & video
English	English
Native speakers	Native & non-native speakers
early '90s	2000s
2200 conversations 1155 in SWDA	171 meetings 139 in AMI-DA
400k utterances	118k utterances
3M tokens	1.2M tokens

Table 2: Comparison between Switchboard and the AMI Meeting Corpus

2 Data

We perform experiments on the Switchboard Dialogue Act Corpus (SWDA), which is a subset of the larger Switchboard corpus, and the dialogue act-tagged portion of the AMI Meeting Corpus (AMI-DA). SWDA is tagged with a set of 220 dialogue act tags which, following Jurafsky et al. (1997), we cluster into a smaller set of 42 tags. AMI uses a smaller tagset of 16 dialogue acts (Carletta, 2007). See Table 2 for details.

Preprocessing We make an effort to normalize transcription conventions across SWDA and AMI.

We remove disfluency annotations and slashes from the end of utterances in SWDA. In both corpora, acronyms are tokenized as individual letters. All utterances are lower-cased.

Utterances are tokenized with BERT's word piece tokenizer with a vocabulary of 30,000. To this vocabulary we added five speaker tokens and prepend each utterance with a speaker token that uniquely identifies the corresponding speaker within that dialogue.

2.1 Pre-training corpora

We also experiment with three unlabeled dialogue corpora, which we use to provide further pre-training for the BERT encoder.

The first two corpora are constructed from the same source as the dialogue act corpora. We use the SWDA portion of the un-labeled Switchboard corpus (SWBD) and the entire AMI corpus (including the 32 dialogues with no human-annotated DA tags that are not included in the DAR training set). In both cases, we exclude dialogues that are reserved for DAR testing.

We also experiment with a much larger a corpus (350M tokens) constructed from OpenSubtitles (Lison and Tiedemann, 2016). Because utterances are not labeled with speaker, we randomly assigned a speaker token to each utterance to maintain the format of the other dialogue corpora.

The pre-training corpora were prepared for the combined masked language modeling and next sentence (utterance) prediction task, as described by Devlin et al. (2019). For the smaller SWBD and AMI corpora, we generate and train on multiple epochs of data. Since there is randomness in the data preparation (e.g., which distractor sentences are chosen and which tokens are masked), we generate each training epoch separately.[1]

3 Model

We use a simple neural architecture with two components: an encoder that vectorizes utterances (BERT), and single-layer RNN sequence model that takes the utterance representations as input.[2] At each time step, the RNN takes the encoded utterance as input and its hidden state is passed to a linear layer with softmax over dialogue act tags.[3]

Conceptually, the encoded utterance represents the context-agnostic features of the utterance, and the hidden state of the RNN represents the full discourse context.

For the BERT utterance encoder, we use the $\text{BERT}_{\text{BASE}}$ model with hidden size of 768 and 12 transformer layers and self-attention heads (Devlin et al., 2019, §3.1). In our implementation, we use the un-cased model provided by Wolf et al. (2020). The RNN has a hidden layer size of 100.

3.1 Pre-training vs. fine-tuning

First, we analyze how pre-training affects BERT's performance as an utterance encoder. To do so, we consider the performance of DAR models with three different utterance encoders:

- `BERT-FT` – pre-trained + DAR fine-tuning
- `BERT-FZ` – pre-trained, frozen during DAR
- `BERT-RI` – random init. + DAR fine-tuning

`BERT-FT` is more accurate than `BERT-RI` by several percentage points on both DA corpora, suggesting that BERT's extensive pre-training does provide some useful information for DAR (Table 3). This performance boost is much more pronounced in the macro-averaged F1 score,[4] which is explained by the fact that at the tag level, pre-training has a larger impact on less frequent tags (see Figure 1 in the supplementary materials).

The `BERT-FZ` performs very poorly compared to either `BERT-FT` or `BERT-RI`, however. It is heavily biased towards the most frequent tags, which explains its especially poor macro-F1 score (Table 3). In SWDA, for example, the model with a frozen encoder predicts one of the two most common tags (Statement-non-opinion or Acknowledge) 86% of the time, whereas those two tags account for only 51% of the ground truth tags. `BERT-FT` is much less biased; it predicts the two most common tags only 59% of the time.

3.2 Impact of dialogue pre-training

Next, we assess the effect of additional dialogue pre-training on BERT's performance as an utter-

[1] For details, see the finetuning example from Hugging Face.

[2] We have experimented with LSTM as the sequence model, but the accuracy was not significantly different compared to RNN. It can be explained by the absence of longer distance dependencies on this level of our model.

[3] Other work has shown that DAR benefits from more sophisticated decoding, such as conditional random field (Chen et al., 2018) and uncertainty propagation (Tran et al., 2017b).

[4] We report both *accuracy* (which is equal to micro-averaged or class-weighted F1) and *macro-F1*, which is the unweighted average of the F1 scores of each class.

ance encoder.[5] Sun et al. (2019) has reported that performing additional pre-training on unlabeled in-domain data improves performance on classification tasks. We want to see if BERT can benefit from pre-training on dialogue data, including from data outside the immediate target domain.

For each of the target corpora (SWDA and AMI-DA), we compare four different pre-training conditions: The in-domain corpus (ID), consisting of the AMI pre-training corpus for the AMI-DA model and the SWBD pre-traning corpus for the SWDA model; the cross-domain corpus (CC), consisting of both the AMI and SWBD pre-training corpora; and finally the OpenSubtitles corpus (OS). As before, we experiment with both frozen and fine-tuned models at the task training stage.

We performed 10 epochs of pre-training on the in-domain models and 5 epochs of pre-training on the cross-domain models so that the total amount of training data was comparable. The OpenSubtitles models were trained for only one epoch but with much more total training time.

In the fine-tuned condition, additional pre-training offers a modest boost in overall accuracy and a substantial boost to the macro-F1 scores, with the cross-domain corpus providing the largest boost. In the frozen condition, only the very large OpenSubtitles corpus is helpful, suggesting that when adapting BERT to dialogue, the size of the corpus is more important than its quality or fidelity to the target domain. Still, pre-training provides nowhere near the performance improvement achieved by fine-tuning on the target task.

4 Discussion

A key aspiration of transfer learning is to expose the model to phenomena that are too infrequent to learn from labeled training data alone. We show some evidence of that here. Pre-trained BERT-FT performs better on infrequent dialogue acts than BERT-RI, suggesting it draws on the extensive pre-training to represent infrequent features of those utterances. Indeed, a simple lexical probe supports this explanation: in utterances where the pre-trained model is correct and the randomly initialized model is not, the rarest word is 1.9 times rarer on average than is typical of corpus as a whole.

[5] In-domain pre-training is sometimes referred to as *fine-tuning*, but we reserve that term for task-specific training on labeled data.

[6] Kozareva and Ravi (2019)

	SWDA		AMI-DA	
	F1	acc.	F1	acc.
BERT-FT	36.75	76.60	43.42	64.93
BERT+ID-FT	43.63	77.01	46.70	**68.88**
BERT+CC-FT	**47.78**	**77.35**	**48.86**	68.79
BERT+OS-FT	41.42	76.95	48.65	68.07
BERT-FZ	7.75	55.61	14.86	48.34
BERT+ID-FZ	6.46	52.30	14.48	48.18
BERT+CC-FZ	5.76	51.14	11.34	40.48
BERT+OS-FZ	**9.60**	**57.67**	**17.03**	**51.03**
BERT-RI	32.18	73.80	34.88	60.89
Majority class	0.78	33.56	1.88	28.27
SotA	-	83.1[6]	-	-

Table 3: Comparison of macro-F1 and accuracy with further in-domain (ID), cross-domain corpus (CC), and OpenSubtitles (OS) dialogue pre-training, for the frozen (FZ) and fine-tuned (FT) conditions. BERT-RI uses a randomly initialized utterance encoder with no pre-training but with fine-tuning.

In spite of that, the representations learned through pre-training are simply not performant without task-specific fine-tuning, suggesting that they are fundamentally lacking in information that is important for the dialogue context. We should note that this is in stark contrast to many other non-dialogical semantic tasks, where frozen BERT performs on par or *better* than the fine-tuned model (Peters et al., 2019).

By performing additional pre-training on a large dialogue-like corpus (OpenSubtitles), we were able to raise the performance of the frozen encoder by a small amount. This deserves further investigation. Bao et al. (2020) find that further pre-training BERT on a large-scale Reddit and Twitter corpus is helpful for response selection, but given the unimpressive results with subtitles, it remains an open question how well the text chat and social media domains transfer to natural dialogue.

There is also abundant room to investigate how speech-related information, such as laughter, prosody, and disfluencies can be incorporated into a DAR model that uses pre-trained features. Stolcke et al. (2000) showed, for example, that dialogue acts can have specific prosodic manifestations that can be used to improve dialogue act classification. Incorporating such information is crucial if models pre-trained on large-scale text corpora are to be adapted for use in dialogue applications.

References

John L. Austin and James O. Urmson. 2009. *How to Do Things with Words: The William James Lectures Delivered at Harvard University in 1955*, 2. ed., [repr.] edition. Harvard Univ. Press, Cambridge, Mass. OCLC: 935786421.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 85–96, Online. Association for Computational Linguistics.

Chandrakant Bothe, Sven Magg, Cornelius Weber, and Stefan Wermter. 2018a. Conversational analysis using utterance-level attention-based bidirectional recurrent neural networks. *Proc. Interspeech 2018*, pages 996–1000.

Chandrakant Bothe, Cornelius Weber, Sven Magg, and Stefan Wermter. 2018b. A Context-based Approach for Dialogue Act Recognition using Simple Recurrent Neural Networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jean Carletta. 2007. Unleashing the killer corpus: Experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation*, 41(2):181–190.

Christophe Cerisara, Pavel Král, and Ladislav Lenc. 2017. On the effects of using word2vec representations in neural networks for dialogue act recognition. *Computer Speech & Language*, 47:175–193.

Saurabh Chakravarty, Raja Venkata Satya Phanindra Chava, and Edward A Fox. 2019. Dialog acts classification for question-answer corpora. In *ASAIL@ ICAIL*.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709, Florence, Italy. Association for Computational Linguistics.

Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue Act Recognition via CRF-Attentive Structured Network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 225–234, New York, NY, USA. Association for Computing Machinery.

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.

Daniel Jurafsky, Liz Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Convolutional Neural Networks for Discourse Compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and Their Compositionality*, pages 119–126.

Zornitsa Kozareva and Sujith Ravi. 2019. ProSeqo: Projection Sequence Networks for On-Device Text Classification. 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 3894–3903, Hong Kong, China. Association for Computational Linguistics.

Pierre Lison and Jorg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, page 7.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining Methods for Dialog Context Representation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845, Florence, Italy. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *NIPS Proceedings*, page 9.

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.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Eugénio Ribeiro, Ricardo Ribeiro, and David Martins de Matos. 2019. Deep dialog act recognition using multiple token, segment, and context information representations. *Journal of Artificial Intelligence Research*, 66:861–899.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, 26(3):339–373.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? In *Chinese Computational Linguistics*, Lecture Notes in Computer Science, pages 194–206, Cham. Springer International Publishing.

Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2017a. A Hierarchical Neural Model for Learning Sequences of Dialogue Acts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 428–437, Valencia, Spain. Association for Computational Linguistics.

Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2017b. Preserving Distributional Information in Dialogue Act Classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2151–2156, Copenhagen, Denmark. Association for Computational Linguistics.

Jesse Vig and Kalai Ramea. 2019. Comparison of Transfer-Learning Approaches for Response Selection in Multi-Turn Conversations. In *Proceedings of the Workshop on Dialog System Technology Challenges*, page 7, Honolulu, Hawaii.

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.

Tianyu Zhao and Tatsuya Kawahara. 2018. A unified neural architecture for joint dialog act segmentation and recognition in spoken dialog system. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–208.

Appendix A

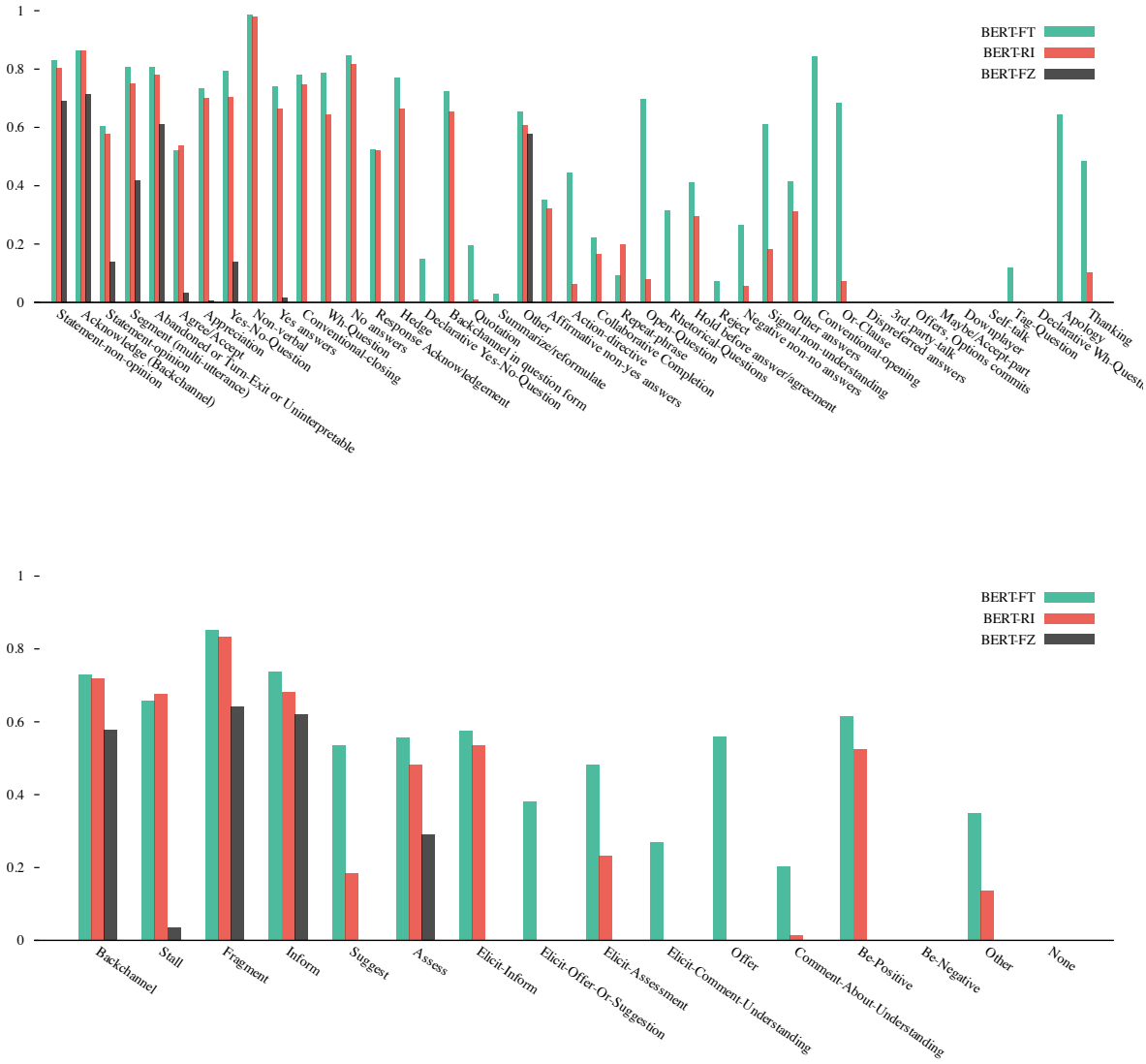


Figure 1: F1 scores by dialogue act for BERT with standard pre-training and DAR fine-tuning (BERT-FT) vs. the same model without pre-training (BERT-RI) and without fine-tuning (BERT-FZ). Dialogue acts are ordered with the most common on the left.

Builder, we have done it: Evaluating & Extending Dialogue-AMR NLU Pipeline for Two Collaborative Domains

Claire Bonial[1], Mitchell Abrams[2], David Traum[3], and Clare R. Voss[1]

[1]U.S. Army Research Laboratory, Adelphi, MD 20783
[2]Institute for Human and Machine Cognition, Pensacola, FL 32502
[3]USC Institute for Creative Technologies, Playa Vista, CA 90094
claire.n.bonial.civ@mail.mil

Abstract

We adopt, evaluate, and improve upon a two-step natural language understanding (NLU) pipeline that incrementally tames the variation of unconstrained natural language input and maps to executable robot behaviors. The pipeline first leverages Abstract Meaning Representation (AMR) parsing to capture the propositional content of the utterance, and second converts this into "Dialogue-AMR," which augments standard AMR with information on tense, aspect, and speech acts. Several alternative approaches and training data sets are evaluated for both steps and corresponding components of the pipeline, some of which outperform the original. We extend the Dialogue-AMR annotation schema to cover a different collaborative instruction domain and evaluate on both domains. With very little training data, we achieve promising performance in the new domain, demonstrating the scalability of this approach.

1 Introduction

We adopt, evaluate, and improve upon the two-step NLU pipeline, described in Bonial et al. (2020), which aims to incrementally tame the variation of incoming natural language that the robot must interpret before responding. For each domain in which it operates, the robot must determine whether or not the commands it receives correspond to one of its executable behaviors, such as MOVEMENT (along a front-back axis) and ROTATION. The NLU pipeline leverages AMR to capture the basic content of the input language, and then a conversion system adds behavior time, completion status and speech act information to the original "Standard-AMR," and updates the main action relation of the input AMR to a relation consistently representing an executable robot behavior (see Fig. 1 for a Standard and Dialogue-AMR example comparison). There are two high-level components of the NLU pipeline: a Standard-AMR parser and a graph-to-graph conversion system to convert the Standard-AMR into Dialogue-AMR. Here, we offer the first evaluation of both the Dialogue-AMR annotation schema itself and the components of the pipeline used to automatically obtain the Dialogue-AMR. We test not only in the human-robot, search-and-navigation dialogue domain for which the schema and pipeline was developed, but also in a somewhat similar, yet challenging domain: human-human communication collaboratively building structures in the virtual gaming environment, "Minecraft." In this way, we address the question of what would happen if we wanted our robot to collaborate on a new and different task. We refer to this challenge as "domain extension," instead of "domain adaptation," as we aim to maintain the coverage of our original domain while extending to a new one.

```
(a)                          (b)
(m / move-01                 (c / command-SA
 :ARG0 (y / you)               :ARG0 (c2 / commander)
 :direction (b / back))        :ARG1 (g / go-02 :completable -
                                       :ARG0 r
                                       :direction (b / back)
                                       :time (a / after
                                                  :op1 (n / now)))
                               :ARG2 (r / robot))
```

Figure 1: *Move back* in (a) Standard-AMR (parser output), (b) Dialogue-AMR (conversion system output).

After providing background on AMR and Dialogue-AMR (§2) and detailing our approach (§3), we report on the human-robot evaluation (§4), followed by the Minecraft evaluation (§5), and domain extension of the conversion system and subsequent evaluation (§6). Our contributions include:
i. Retraining existing **Standard-AMR parsers** (3.1) and evaluating on the human-robot (4.1) and Minecraft domains (5.1);
ii. Evaluating and improving a **conversion system** for automatically obtaining Dialogue-AMR (3.2) in both the robot (4.2) and Minecraft (5.2) domains;
iii. Extending the coverage of the **Dialogue-AMR annotation schema** (2.1) to a new domain (6.1) and evaluation after domain extension (6.3).

Proceedings of the 14th International Conference on Computational Semantics, pages 173–183
June 17–18, 2021.

2 Background

To summarize where this work is situated with respect to the past research on this topic—while Bonial et al. (2020) details the Dialogue-AMR annotation schema and proposes the two-step pipeline as one way of automatically obtaining Dialogue-AMR, the technical details of an implementation of the pipeline itself are not described and no evaluation is given. Subsequent research from Abrams et al. (2020) does provide an initial evaluation of a baseline version of the graph-to-graph conversion component of the proposed two-step pipeline; we adopt and evaluate an updated version of this component (described in greater detail in §3.2), however, our evaluation is not directly comparable to the evaluation given in Abrams et al. (2020), since the earlier version of the component was tested on only a limited subset of the annotation categories of Dialogue-AMR. Thus, the current paper constitutes the first evaluation of the proposed two-step pipeline and its components, as well as an evaluation of the extensibility of those components and the Dialogue-AMR schema itself to a new domain.

2.1 AMR & Dialogue-AMR

The two-step NLU pipeline of Bonial et al. (2020) leverages AMR, as it abstracts away from some idiosyncratic surface variation in favor of a more consistent representation for the same concept. This serves the purposes of a dialogue system well: AMR smooths over the nuances of language that may be unimportant for mapping a particular input to one of the robot's behaviors. Nonetheless, "Standard-AMR" does not represent some aspects of meaning that are critical for the human-robot dialogue domain, where the robot must be cued as to what the current dialogue state is, as well as what the current time and completion status of various instructions are. To capture this information, the NLU pipeline uses the "Dialogue-AMR" formalism (Bonial et al., 2020), which adds action time, completion status (i.e., limited tense, aspect) and speech act information to the Standard-AMR. Additionally, to facilitate the final step of mapping to one of the robot's behaviors, Dialogue-AMR further generalizes from the input language, converting a variety of surface realizations (e.g., *turn, rotate, pivot*) of a particular action relation into a single canonical numbered relation (e.g., `turn-01`) to represent one of the robot's behaviors (e.g., ROTATION). Standard-AMR and Dialogue-AMR are contrasted in Figs. 1 and 2.

In Dialogue-AMR, the content of the Standard-AMR is nested in a structure that adds the speech act information as the root predicate (e.g., `command-SA` in Figs. 1, 2). Additionally, the main action from the Standard-AMR (e.g., `move-01`) is converted to one of the action relations (e.g., `go-02`), termed the "robot-concept relation" that maps to an executable robot behavior. Information about the time of that behavior is added (in Fig. 2, the motion event will happen in the future, after the speaking time of the command; thus, it is represented as `:time after-now`).[1] Finally, the behavior completion status, a type of aspect information, is added—whether or not the instructed behavior is telic or contains a clear end point (in Fig. 2, indicated by `completable +`).[2]

Dialogue-AMR draws upon an inventory of 13 speech acts and 26 robot behaviors or "robot-concept relations." Action time and completion status are integrated into Dialogue-AMR by adopting the annotation schema of Donatelli et al. (2018), which categorizes the robot behavior as *past, present,* or *future,* and categorizes 4 aspectual labels: `:stable +/-`, `:ongoing +/-`, `:complete +/-`, and `:habitual +/-`. Dialogue-AMR uses the added category `:completable +/-` to signal whether or not a hypothetical event has an end-goal achievable for the robot.

2.2 Annotated Corpora

We draw from two datasets with Standard-AMR annotations, collected with the aim of developing an interactive agent for collaboration in grounded scenarios. We leverage the DialAMR corpus (Bonial et al., 2020) as training and evaluation data for the NLU pipeline within the human-robot dialogue domain. DialAMR encompasses 1122 instances of The Situated Corpus of Understanding Trans-

[1] In ongoing work to extend the Dialogue-AMR schema, we plan to refine the `:time` annotations to better capture the possibility that an instructed action could already be underway at speaking time, given that we observed that in highly collaborative dialogue, utterances often overlap with actions.

[2] End-point information is needed by a robot to execute a behavior in a low-bandwidth environment where there is a communications lag, precluding real-time voice teleoperation. What constitutes a fully specified behavior is somewhat task and robot-specific; for example, a robot with a static, front-facing camera can assume, as a default, that a picture taken for a user will be from this perspective unless the user specifies otherwise, but a robot with a movable, 360-degree view camera may need to ask the user to provide information on the desired camera angle.

actions (SCOUT), annotated with both Standard-AMR and Dialogue-AMR. SCOUT is comprised of over 80 hours of dialogues from the robot navigation domain (Marge et al., 2016, 2017), collected via a "Wizard-of-Oz" experimental design (Riek, 2012), in which participants directed what they believed to be an autonomous robot to complete search and navigation tasks. The DialAMR corpus was used in the development of the Dialogue-AMR schema, as well as training and testing of the components of the conversion system of Abrams et al. (2020), which we initially adopt, described in §3.2. The data from SCOUT selected for the DialAMR corpus includes a randomly selected, continuous 20-minute experimental trial, which contains 304 utterances (called the *Continuous-Trial* subset). This is the held-out test set that we use throughout our "in-domain" evaluation, as it is representative of an ongoing human-robot interaction.

In addition to in-domain evaluation, we extend evaluation of the Dialogue-AMR schema and NLU pipeline by annotating and testing on the Minecraft Dialogue Corpus (Narayan-Chen et al., 2019). This corpus consists of 509 conversations and game logs, in which two humans communicate via the Minecraft gaming interface chat window while collaboratively building blocks structures. Standard-AMR annotations for the Minecraft corpus (Bonn et al., 2020) were obtained from the developers via a private data-sharing agreement. Our addition of Dialogue-AMR annotations to this corpus is described in §6.1.

3 Approach: Two-Step NLU Pipeline

We adopt and evaluate the two-step NLU pipeline described in Bonial et al. (2020) and Bonial et al. (2019), including both a Standard-AMR parser and a system for converting this into Dialogue-AMR. We describe our selection of an initial Standard-AMR parser and conversion system, both of which we retrain and improve upon, below.

3.1 Standard-AMR Retrained Parser

Standard-AMR provides an initial interpretation of an utterance to be transferred to the Dialogue-AMR. Therefore, an effective Standard-AMR parser is critical for the overall success of the NLU pipeline. We considered several open-source AMR parsers as candidates, and selected two recent releases, the parsers described in Zhang et al. (2019) and Lindemann et al. (2019), which both make use of BERT embeddings (Devlin et al., 2019) and were evaluated on AMR releases, thus providing us with baselines to compare them to each other and to assess our retrained models against their reported performances.

We were able to retrain both of these state-of-the-art AMR parsers on the AMR 2.0 corpus and the recently released AMR 3.0 corpus (a larger corpus including the 2.0 data), and then also retrain them on each of these individual releases of Standard-AMR together with the Standard-AMR subset of the DialAMR corpus of over 800 Standard-AMRs, to adapt them to our human-robot dialogue domain. We evaluated these particular combinations of training data because we wanted to explore whether or not the larger set of data in the AMR 3.0 corpus improved performance on the human-robot dialogue domain, or if it further washed out the distinctions from our smaller in-domain corpus. This yielded a total of eight parsers (see Table 1) for us to evaluate and select from for the purpose of then including in the full NLU parsing pipeline.

3.2 Conversion System

The next step in the NLU pipeline is a graph-to-graph conversion system that uses the input of the utterance text and the Standard-AMR graph to create a Dialogue-AMR graph. We leverage an existing conversion system, "Abrams+", and experiment with improvements to how it classifies the robot-concept relation in our own updated graph-to-graph conversion system, "G2G".

3.2.1 Abrams+ Conversion

We obtained a version of the conversion system described in Abrams et al. (2020), which had been updated by that author in two ways: i. expanded to handle the additional speech acts and robot-concept relation categories of the full Dialogue-AMR schema outlined in Bonial et al. (2020), not all of which were present during the original development, and ii. shifted from a Naïve Bayes to a SVM model for speech act classification. We refer to this system as "Abrams+". This graph-to-graph conversion system implements both rule-based and classifier-based methods in converting a Standard-AMR graph into a Dialogue-AMR graph, and leverages the original utterance and the structure of the Standard-AMR to produce the final Dialogue-AMR, which includes the speech act, tense and aspect information, and a designation of the robot-concept relation. As we use this system as our

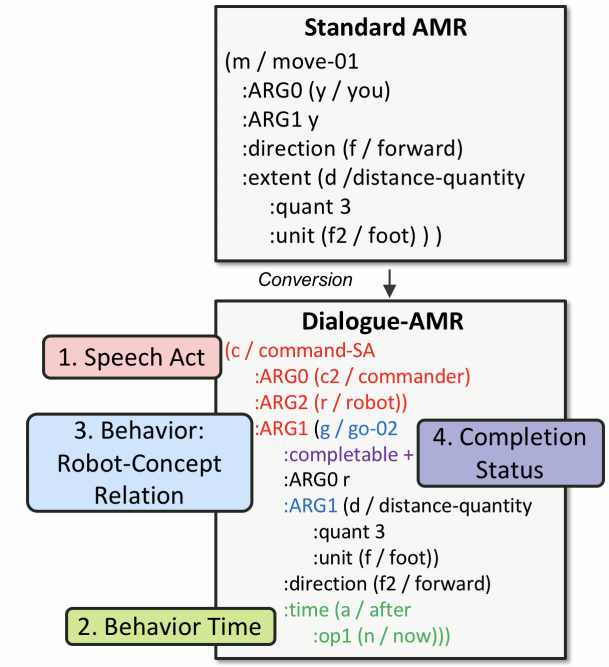


Figure 2: Standard and Dialogue-AMR comparison for Commander instructing robot *Move forward three feet*.

starting point for improvement, we will briefly describe how each of these additions are made in the order just listed, but refer the reader to Abrams et al. (2020) for full details.

Following the numbering of the example in Fig. 2, the first step in the transformation process employs a SVM model with token unigrams features to predict the speech act from the original utterance—critical information for human-robot communication that cannot be gleaned from the Standard-AMR graphs alone.[3] After classification, the speech act label is then stored as a slot to be added to the Dialogue-AMR graph and referenced for decision-making processes downstream. Second, to add behavior time, another classifier—a Naïve Bayes model using token unigrams as features—determines if the event corresponding to the robot behavior pertains to a *past*, *present*, or *future* action. Third, designation of the robot behavior is implemented through a keyword-based approach, which extracts the top root relation (keyword) in the Standard-AMR and checks it against a keyword dictionary of similar actions, and maps it to a single robot-concept relation. Fourth, particular combinations of speech act, tense, and the presence or absence of certain arguments of the robot-concept relation trigger an aspectual label that corresponds to an action's completion status. In the final step of transformation process, the system's rule-based methods use pattern matching techniques to serve multiple functions, including slot filling and slot changing (e.g., transforming mentions of *you* to the fixed role of the `addressee` in Dialogue-AMR).

[3]We acknowledge that the interpretation of speech acts, and indirect speech acts in particular, can be affected by context. Following (Hinkelman and Allen, 1989), we start with only the linguistic signal in the first phase. Since the restricted domain is predictable, it is usually sufficient, but further research aims to leverage situational information and dialogue context where necessary, e.g., to disambiguate an ability question from an indirect instruction.

3.2.2 G2G: Our Updated Conversion System

While we hypothesize speech act, tense, and aspect classification may be fairly robust to language in a new domain, we readily acknowledge that new domains will require the robot to engage in novel behaviors, for example, BUILDING in the Minecraft domain. Thus, although there are many different aspects of the conversion system that we could attempt to improve upon (e.g., classifier types, ordering of components), we saw an opportunity to have the most impact on system performance in multiple domains by focusing on varying the robot-concept relation classification approach. We describe three variants (one keyword-based and two classifier-based) of our updated G2G conversion system below.

G2G *Expanded* Keyword-Based Variant We expanded upon the keyword approach of the Abrams+ system, which was restricted to searching for keyword matches with the top, root relation of the Standard-AMR. We found that this restriction was problematic because the same root relation in the Standard-AMR could correspond to multiple robot-concept relations. *Move* and *go*, generally parsed as `move-01` and `go-02`, are particularly prevalent and could correspond to either front-back MOVEMENT or a ROTATION behavior; both of these were keywords triggering front-back movement in Abrams+, which therefore incorrectly categorized utterances like *Move right 45 degrees* (a ROTATION behavior). In our expansion, the G2G keyword variant searches for matches within all utterance tokens, AMR relations, and arguments. Furthermore, the keyword dictionary was informed by a data-driven analysis in which we created histograms of all utterance tokens and Standard-AMR relations within an instance mapped to a particular robot-concept relation in the manual Dialogue-AMR annotations. In this way, we could see which words and relations occurred with multiple robot-concept relations, like `move-01`, and therefore

remove these from our keyword dictionaries, while adding keywords that are unique to a particular robot-concept relation in the data, such as *degrees*, which consistently cues a ROTATION behavior.

G2G One-Hot and GloVe Variants We also experimented with classifier-based approaches to robot-behavior classification, which we hypothesized may be more efficient to extend to a new domain than a keyword-based approach. The classifiers are Support Vector Machines with different vectorization methods including one-hot encoding and word embeddings from GloVe. Training data for the robot-concept relation classifier comes from examples of each robot-concept category in Bonial et al. (2020), gold-standard labels from the *Continuous-Trial* subset utterances 101-305 (those not used in a held-out test set), and examples pulled from speech act classifier training bins. There are a total of 26 labels for this task, and while many of the movement actions were abundant from these other sources, some of the minority labels (e.g., `equip-01`, `wait-01`, `clarify-10`) required up-sampling to balance training proportions.

4 In-Domain Evaluation

4.1 In-Domain Standard-AMR Parsing

We evaluated the retrained parsers on the SCOUT *Continuous-trial* dataset. We note substantial improvement in Standard-AMR parsing Smatch scores on this set when training with DialAMR in addition to the base training sets (AMR 2.0 and 3.0).[4] Results for the AMR parsing models are presented in Table 1. The noticeably high scores on the parsers retrained on the AMR 3.0 + DialAMR is due in large part to the nature of the speakers' language in the SCOUT corpus and the high levels of similarity in participants' instructions to the robot. This underscores how critical evaluation in another dialogue domain is. We note that, at the segment level as well as can be seen in the Table 1, the Lindemann et al. (2019) parser retrained with DialAMR data evaluated across-the-board to higher scores than the comparably retrained Zhang et al. (2019) parser. Of those two Lindemann et al. (2019) parsers whose Smatch scores did not differ significantly, we selected the one trained with the larger 3.0 dataset with its larger language model as the first component in the full parsing pipeline.

[4]Smatch is an evaluation algorithm for scoring AMR graphs (Cai and Knight, 2013).

Parser	Training	P	R	F
Zhang et al.	AMR 2.0	.47	.77	.58
	2.0 + DialAMR	.73	.77	.75
	AMR 3.0	.52	.80	.63
	3.0 + DialAMR	.88	.89	.89
Lindemann	AMR 2.0	.53	.77	.63
	2.0 + DialAMR	**.92**	.94	**.93**
	AMR 3.0	.55	.81	.65
	3.0 + DialAMR	.91	**.95**	**.93**

Table 1: Retrained AMR parser Smatch results on SCOUT *Continuous-trial* test set.

4.2 In-Domain Conversion to Dialogue-AMR

To pinpoint the performance of the conversion system alone (without error introduced by the automatic Standard-AMR parsing), we report results with gold-standard, manually assigned input Standard-AMR parses. Results are summarized in Evaluation Domain A of Table 2. Focusing initially on the overall Smatch Precision, Recall, and F-scores of the conversion system, our updated system, G2G, leveraging the classifier with one-hot vectorization achieves the highest precision (.85) and F-score (.83) in our domain. All approaches perform comparably overall, especially given that Smatch scores can vary slightly (Opitz et al., 2020) because Smatch is a non-deterministic, greedy hill-climbing algorithm with a preset, default number of random restarts (Cai and Knight, 2013).

Drilling down into the accuracy of the individual component classification tasks, we find accuracy scores of 1.00 for speech acts, .93 for tense, and .93 for aspect across all system variants, as these components are unchanged, and we only alter the robot-concept classification. Again, we note that these accuracy scores are extremely high, given the repetitive nature of the language and prevalence of certain types of commands and feedback assertions. For robot-concept classification, the G2G *expanded* keyword approach (.97 accuracy) does outperform the Abrams+ baseline keyword method (.94 accuracy). Both keyword approaches outperform the G2G classifier-based approaches: one-hot vectorization achieves an accuracy of .90 and GloVe an accuracy of .84. Notably, higher accuracy on the robot-concept classification task does not necessarily translate to higher Smatch F-scores overall. High component accuracy but lower overall F-Score generally indicates that while the system is correctly determining all of the information being added to the Dialogue-AMR, it is not always putting these pieces together correctly. In

Conversion Variant	Evaluation Domain A: SCOUT test data				Evaluation Domain B: Minecraft test data			
	Smatch			Robot Concept	Smatch			Robot Concept
	P	**R**	**F**	**Accuracy**	**P**	**R**	**F**	**Accuracy**
Abrams+	.81	**.82**	.82	.94	.71	.63	.67	.30
G2G-Keyword	.82	**.82**	.82	**.97**	.72	**.64**	**.68**	**.32**
G2G-One-Hot	**.85**	**.82**	**.83**	.90	.73	.62	.67	.20
G2G-GloVe	.84	.81	.82	.84	**.74**	.62	.67	.24
Extended G2G-Keyword	.82	.81	.82	**.94**	.73	**.67**	.70	.41
Extended G2G-One-Hot	**.85**	**.82**	**.83**	.93	**.77**	.65	**.71**	**.54**
Extended G2G-GloVe	.84	.81	.82	.89	.76	.65	.70	.45

Table 2: Summary of Smatch scores & Robot-Concept Relation classification accuracy for each variant conversion system, including our G2G system before and after Minecraft domain extension, tested on SCOUT and Minecraft.

other words, the final step in the conversion system, where slots are captured and changed from the original Standard-AMR structure to the structure of the Dialogue-AMR, is where some of the error reflected in Smatch scores stems from.

5 Minecraft Domain Evaluation

In this section, we report on the Minecraft domain performance of the NLU pipeline with the retrained Standard-AMR parser, the Abrams+ conversion system, and our updated G2G system variants prior to any domain adaptation in order to determine how vital domain extension really is in somewhat similar instruction-giving domains. Given that theoretically speech acts, tense and aspect are somewhat consistent in language regardless of the domain, we hypothesize that these features of our annotation schema and the components of the conversion system capturing them will perform reasonably well on the new Minecraft dialogue domain. However, the main actions or behaviors involved in the collaboration of interlocutors in the original search and navigation domain are quite different from those of building virtual structures from blocks in the new Minecraft domain. We therefore expect that the conversion system will fail to correctly map many of the main action predicates in the Minecraft dialogues to an executable robot behavior. However, we accept this as an interesting question of domain extension for moving our robot to a new task: Is it more efficient to expand a rule-based approach for capturing these new behaviors, or to use a classifier-based approach?

5.1 Minecraft Standard-AMR Parsing

We test the parser selected as the first pipeline component (described in §4.1) on Minecraft data, scoring the parser output on 100 sequential instances of Minecraft dialogue against manually assigned Standard-AMR annotations.[5] The overall Smatch F-score is .57, with a Precision of .63 and Recall of .52. Thus, despite the potential similarity in the two instruction-giving dialogue domains, it is clear that the automatic parsing performance is significantly worse for the Minecraft data than our original domain (where the best Smatch F-score was .93). Error analysis reveals some extremely complicated language phenomena, including dimensions and frequency expressions capturing, for example, the repetition of a placement action: *For the four squares that come out from the middle blocks, add two blue blocks on.* Although this indicates that the parser would benefit from retraining with Minecraft data,[6] in our immediate research we focus on domain extension of the conversion system in order to explore how robust the conversion system might be to noise in the parser input.

5.2 Minecraft Conversion to Dialogue-AMR

This evaluation compares the conversion system output against manually assigned Dialogue-AMRs for the same 100-instance, sequential subset of utterances from the Minecraft corpus used as the test set for the Standard-AMR parser (see §6.1 for Dialogue-AMR annotation details); again, we use gold-standard, manually assigned Standard-AMR parses as input to the conversion system. Results are summarized in Evaluation Domain B of Ta-

[5]The Minecraft AMR corpus includes AMRs for the locations of blocks (expressed as Cartesian coordinates) as each movement takes place; because our focus is natural language dialogue, we removed these instances from our test set.

[6]Bonn et al. (2020) report an F-score of .66 on a Minecraft test set after retraining the Zhang et al. (2019) parser on Minecraft data.

ble 2. Focusing first on overall Smatch scores, our updated system variant leveraging the *expanded* keyword approach performs slightly better (.68 F-score) than both the baseline Abrams+ (.67 F-score) and the classifier-based approaches (.67 F-scores). Although the scores have dropped about 15 points from the original domain, they remain comparable across variants.

When drilling down into the accuracy of the individual components of the conversion system, we find that robot concept classification yields the lowest accuracy scores, with a range of .20-.32. Among the variant approaches to robot-concept classification explored, the *expanded* keyword approach achieves the highest accuracy. The speech act and tense have the same accuracy scores across all versions, .44 and .56, respectively, since these classifiers are stable within the system variants. In this evaluation, aspect varies slightly across approaches as it depends on combinations of speech act and robot-concept relation slot values—its accuracy ranges from .25-.49, with the Abrams+ variant obtaining the highest result. Thus, we see that our hypothesis that speech act, tense, and aspect classification may be fairly robust to a new domain is partially confirmed: robot-concept classification is certainly the most challenging with the lowest accuracy, but the performance of all components is significantly worse than the original domain, suggesting more widespread differences in the language of the two domains.

6 Domain Extension

Here, we describe the small amount of domain extension done to tailor our G2G conversion system to the Minecraft domain, beginning with extensions of the annotation schema itself.

6.1 Extending Dialogue-AMR Schema

One expert Standard-AMR and Dialogue-AMR annotator provided manual Dialogue-AMR annotations to a continuous 100-instance subset of the Minecraft corpus to serve as a test set. This was done by manually augmenting the Standard-AMR release of the Minecraft corpus, maintaining all of the Standard-AMR annotation choices. Additionally, a separate, continuous 200-instance subset of the data was annotated with speech acts and the corresponding robot-concept relations of Dialogue-AMR to serve as training data for the speech act classifier and robot-concept relation classification.[7]

In providing the manual Dialogue-AMR annotation of the Minecraft data, we noted several changes and additions that needed to be made to the annotation schema to account for novel concepts arising in the collaborative building domain, as well as novel dialogue phenomena. First, as expected, we added agent behaviors that would be needed for this domain: BUILDING, represented with the relation `build-01` (e.g., *What are we **building** this time?*), and PLACING, represented with the relation `move-01` (e.g., *Please **place** two red blocks on top of each side...*).

Second, we noted novel dialogue phenomena that we had not observed in the SCOUT data. Speech acts were often nested in this data, such that the content of one speech act was not a typical agent behavior (e.g., a speech act of commanding a ROTATION behavior), but instead another speech act. For example, there were frequent requests for evaluation, often after each building step was completed: *How's this?* and *Is this good?*[8] As a result, we had to shift our annotation schema and conversion system in order to allow for speech act relations to sit where we would normally expect the robot-concept relation.

Finally, we noted frequent use of the verb *need* as an indicator of a less direct command in the Minecraft data: *This will **need** to be placed as far right as you can....* This was interpreted by the interlocutor as a command, i.e., *Place this as far right as you can*. Thus, the *need* relation that roots the Standard-AMR ultimately mapped to the `command-SA` relation of the Dialogue AMR. This phenomenon has significant ramifications for the conversion system, as it was generally assumed, for the SCOUT data, that the utterance and Standard-AMR provides propositional content cuing the robot-concept relation, but we did not expect AMR relations corresponding to the speech act in our

[7] Contact the first author for Minecraft Dialogue-AMR annotations used for train/test.

[8] Following Bunt et al. (2012), Dialogue-AMR speech acts are distinguished between Information Transfer Functions and Action Discussion Functions. Thus, while syntactically questions, cases such as *How's this?* are not annotated using the Dialogue-AMR `Question` speech act, which is reserved for questions that obligate the addressee to introduce new information content into the conversation and demonstrate a commitment to the answer assertion (Traum, 2003). In contrast, these cases obligate the addressee to evaluate the current state of play while simultaneously providing feedback that common conversational ground has been achieved with respect to the desired structure. Indeed, common responses such as *Excellent, Builder* do not fit with a question interpretation.

domain, although plausible (e.g., ***I command*** *you to move forward*).

6.2 Extending Robot-Concept Classification

We added to our *expanded* keyword dictionary to test the effectiveness of a rule-based approach in domain extension. Only two additional concepts were required, `build-01` and `move-01`, but these robot concepts are extremely prevalent in the data. Additionally, in order to test how well a classifier-based approach would capture new behaviors and extend the conversion system to a new domain, we retrained the robot-concept classifier on 166 new manually-annotated training instances of robot concepts from the Minecraft domain. Domain extension also included retraining the speech act classifier on 224 speech acts found in 200 instances of manually annotated Minecraft data.

6.3 Domain-Extended G2G Evaluation

After domain extension, the G2G variant leveraging the one-hot classifier (.71 F-score) very slightly outperforms the keyword (.70 F-score) and GloVe variants (.70 F-score) (again, comparing system output against manually assigned Dialogue-AMRs for the continuous, 100-instance Minecraft test set). Results are summarized in the bottom three rows of Evaluation Domain B of Table 2. The scores remain comparable across all three variants, but we do see improvement overall when comparing against system variants prior to domain extension.

Turning to analysis of the accuracy of individual components of the conversion system, the additional training instances improve speech act classification (from .44 prior to retraining to .57 after) and robot-concept classification for the Minecraft domain. Prior to domain extension, the *expanded* keyword variant achieved the highest accuracy for robot-concept classification (.32), but classifier-based methods with more training data outperform even a domain-extended, data-driven keyword approach, which achieves an accuracy of .41, while one-hot vectorization achieves an accuracy of .54 and GloVe .45. Error analysis reveals that the keyword-based approach struggles to classify robot concepts in this domain, in part, because of language that contains vocatives (e.g. *Excellent, builder*)–which triggers a top `say-01` relation in the Standard-AMR graph–and various uses of *need*, which trigger a `need-01` relation. As noted in the discussion of domain extension of the annotation schema (§6.1), both of these root relations do not cue any domain robot concept, but rather provide information about speech acts and speaker/listener roles, which were consistently implicit in our original domain. Thus, we are currently updating the system to allow for certain relations in the Standard-AMR (e.g., `need-01`) to cue for or map to particular speech acts (e.g., `command-SA`).

This demonstrates a weakness of the keyword-based approach in general: unforeseen linguistic phenomena such as vocatives can strongly affect the accuracy of this approach, while the classifier approach is more robust to these differences since it considers all tokens in the utterance for robot-concept relation prediction, thereby avoiding mis-classification due to this kind of "noise" in the data. When considering our earlier hypothesis that the classifier-based approach to robot-concept classification would be more efficient to extend to a new domain than the keyword-based approach, the results and error analysis here provide modest support for this hypothesis. Both approaches are similarly time-efficient as far as the initial extension efforts are concerned: the keyword approach requires manual observation of the data and subsequent selection and addition of keywords to the dictionaries associated with certain robot-concept relations, while the classifier approach requires some additional manual annotation in the new domain. However, empirically the classifier-based approach slightly outperforms the keyword-based approach in the Minecraft domain, and extending the keyword-based approach requires additional changes in traversal of the graph in order to find the appropriate concept to serve as the keyword for matching, so the effort necessarily goes beyond merely selecting and adding keywords.

Turning back to our original SCOUT test set after Minecraft domain extension (results summarized in the bottom three rows of Evaluation Domain A in Table 2), we find that tailoring the conversion system to Minecraft and expanding the coverage of language that the system can handle has little negative effect on performance in our original domain. We see comparable results for the classifier-based model using one-hot vectorization, maintaining an F-score of .83, which was also the best-performing model for the original domain.

6.4 Full Automatic Pipeline Evaluation

In order to scale up to real-time use, the two-step NLU pipeline will leverage the retrained automatic Standard-AMR parser described in §3.1; however,

up to this point we have reported conversion system results using manually obtained, gold-standard Standard-AMR parses in order to explore the validity of our conversion system approaches without the noise from parsing. Table 3 summarizes the performance of the overall best-performing (across both Smatch scores and component accuracy) *expanded* keyword and one-hot vectorization classifier G2G variants, after domain extension, given Standard-AMR input from the parser. The *expanded* keyword variant is the best-performing model with automatic input, but the scores are close. Although the Smatch F-score has dropped from .71 (with gold-standard input) to .59, we still find this to be very encouraging performance, given the challenges of semantic parsing in a new domain.

Conversion Variant	**SCOUT**			**Minecraft**		
	P	**R**	**F**	**P**	**R**	**F**
Ext. G2G Keyword	.75	.76	.75	**.67**	**.53**	**.59**
Ext. G2G One-Hot	**.83**	**.80**	**.81**	.62	.52	.57

Table 3: Smatch scores for best-performing domain-extended (ext.) G2G variants **using automatically obtained Standard-AMR input** from retrained parser.

7 Related Work

This research is part of a growing body of work in representing various levels of interpretation in existing meaning representation frameworks, and in AMR in particular. We briefly note especially relevant work here. Bastianelli et al. (2014) present their Human Robot Interaction Corpus (HuRIC) following the same Penman Notation (Penman Natural Language Group, 1989) syntax of AMR, but significantly altering AMR to use the sense distinctions and semantic role labels of FrameNet (Fillmore et al., 2012), thereby rendering the use of automatic parsers trained on AMR data challenging. Shen (2018) presents a small corpus (266 instances) of manually annotated AMRs for spoken language to explore the validity of using AMR for spoken language understanding, with promising results but noting that additional data is needed. There is also a neural AMR graph converter for abstractive summarization (producing summary graphs from source graphs) (Liu et al., 2015); however, neural approaches require substantial training data in the form of annotated input and output graphs. The current motivation for the multi-step approach explored here is to handle a low resource problem, as we lack sufficient data to experiment with employing a neural network.

8 Conclusions & Future Work

This paper evaluates and improves upon a two-step NLU pipeline that gradually tames the variation of language so that it can be understood and acted upon by a robot with a limited repertoire of domain concepts and behaviors. After enumerating the extensions needed for the annotation schema itself and contributing a dataset of Dialogue-AMR for the new Minecraft collaborative dialogue domain, we achieve promising results with roughly 200 instances of training data.

We have integrated our updated pipeline into a software stack for a physical robot and are now performing a series of experiments where we use the same dialogue-management system, but vary the NLU component in order to compare task success with the two-step NLU pipeline against a baseline NLU system with a simple syntactic parser. We hypothesize that the NLU pipeline described here, and the deeper semantics of Dialogue-AMR specifically, will be especially advantageous for tracking and grounding user utterances involving coreference (e.g., *Go to **the sign** and send a picture of **it**.*), light verb constructions, which AMR represents identically to parallel synthetic verbs (e.g., *make a left turn; turn left*), negation (e.g., *no, not the door on the right, the left!*), and complex, nested prepositions (e.g., *move through the doorway in front of you on the left*)—all utterances where a simple syntactic parse has been found to lack information needed for interpretation of the intent and grounding. The extrinsic evaluation will also provide an opportunity to explore whether or not the conversion system variant with the best overall Smatch scores corresponds to the best real-world performance, or if we should consider other metrics, such as S^2match (Opitz et al., 2020) and SemBleu (Song and Gildea, 2019). As our results did not demonstrate a clear "best" rule-based, keyword or classifier approach to domain extension, we will continue to experiment with all three variants and consider which is the most time-efficient to extend, either by adding to the keyword dictionary or adding annotations. Overall, we are optimistic that the semantic representation of Dialogue-AMR, which provides a deeper understanding of both what a person said and what they really meant in the conversational context, will enhance human-robot collaboration.

References

Mitchell Abrams, Claire Bonial, and Lucia Donatelli. 2020. Graph-to-graph meaning representation transformations for human-robot dialogue. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 250–253, New York, New York. Association for Computational Linguistics.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Luca Iocchi, Roberto Basili, and Daniele Nardi. 2014. HuRIC: a human robot interaction corpus. In *LREC*, pages 4519–4526.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Claire Bonial, Lucia Donatelli, Stephanie M. Lukin, Stephen Tratz, Ron Artstein, David Traum, and Clare Voss. 2019. Augmenting Abstract Meaning Representation for human-robot dialogue. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 199–210, Florence, Italy. Association for Computational Linguistics.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. 2012. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 430–437.

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.

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.

Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 96–108.

Charles J Fillmore, Russell Lee-Goldman, and Russell Rhodes. 2012. The FrameNet Constructicon. *Sign-based construction grammar*, pages 309–372.

Elizabeth Hinkelman and James Allen. 1989. Two constraints on speech act ambiguity.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4576–4585, Florence, Italy. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Matthew Marge, Claire Bonial, Brendan Byrne, Taylor Cassidy, A. William Evans, Susan G. Hill, and Clare Voss. 2016. Applying the Wizard-of-Oz technique to multimodal human-robot dialogue. In *RO-MAN 2016: IEEE International Symposium on Robot and Human Interactive Communication*.

Matthew Marge, Claire Bonial, Ashley Foots, Cory Hayes, Cassidy Henry, Kimberly Pollard, Ron Artstein, Clare Voss, and David Traum. 2017. Exploring variation of natural human commands to a robot in a collaborative navigation task. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 58–66.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. 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.

Penman Natural Language Group. 1989. The Penman user guide. *Technical report, Information Sciences Institute*.

Laurel Riek. 2012. Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *Journal of Human-Robot Interaction*, 1(1).

Hongyuan Shen. 2018. *Semantic Parsing in Spoken Language Understanding using Abstract Meaning Representation*. Ph.D. thesis, Brandeis University.

Linfeng Song and Daniel Gildea. 2019. Sembleu: A robust metric for amr parsing evaluation. *arXiv preprint arXiv:1905.10726*.

David Traum. 2003. Semantics and pragmatics of questions and answers for dialogue agents. In *proceedings of the International Workshop on Computational Semantics*, pages 380–394.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR Parsing as Sequence-to-Graph Transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy. Association for Computational Linguistics.

A Transition-based Parser for Unscoped Episodic Logical Forms

Gene Louis Kim[♡], **Viet Duong**[♢], **Xin Lu**[♠], **and Lenhart Schubert**[♣]
University of Rochester
Department of Computer Science
{gkim21[♡],schubert[♣]}@cs.rochester.edu
{vduong[♢],xlu32[♠]}@u.rochester.edu

Abstract

"Episodic Logic: Unscoped Logical Form" (EL-ULF) is a semantic representation capturing predicate-argument structure as well as more challenging aspects of language within the Episodic Logic formalism. We present the first learned approach for parsing sentences into ULFs, using a growing set of annotated examples. The results provide a strong baseline for future improvement. Our method learns a sequence-to-sequence model for predicting the transition action sequence within a modified cache transition system. We evaluate the efficacy of type grammar-based constraints, a word-to-symbol lexicon, and transition system state features in this task. Our system is available at https://github.com/genelkim/ulf-transition-parser. We also present the first official annotated ULF dataset at https://www.cs.rochester.edu/u/gkim21/ulf/resources/.

1 Introduction

EL-ULF was recently introduced as a semantic representation that accurately captures linguistic semantic structure within an expressive logical formalism, while staying close to the surface language, facilitating annotation of a dataset that can be used to train a parser (Kim and Schubert, 2019). The goal is to overcome the limitations of fragile rule-based systems, such as the Episodic Logic (EL) parser used for gloss axiomatization (Kim and Schubert, 2016) and domain-specific ULF parsers used for schema generation and dialogue systems (Lawley et al., 2019; Platonov et al., 2020). EL's rich model-theoretic semantics enables deductive inference, uncertain inference, and natural logic-like inference (Morbini and Schubert, 2009; Schubert and Hwang, 2000; Schubert, 2014); and the unscoped version, EL-ULF, supports Natural Logic-like monotonic inferences (Kim et al., 2020) and inferences based on some classes of entailments, presuppositions, and implicatures which are common in discourse (Kim et al., 2019). The lack of robust parsers have prevented large scale experiments using these powerful representations. We will refer to EL-ULF as simply ULF in the rest of this paper.

```
(i.pro ((pres want.v)
        (to (dance.v
             (adv-a (in.p (my.d ((mod-n new.a)
                                 (plur shoe.n)))))))))
```

Figure 1: An example ULF for the sentence, *"I want to dance in my new shoes"*.

In this paper we present the first system that learns to parse ULFs of English sentences from an annotated dataset, and provide the first official release of the annotated ULF corpus, whereon our system is trained. We evaluate the parser using SEMBLEU (Song and Gildea, 2019) and a modified version of SMATCH (Cai and Knight, 2013), establishing a baseline for future work.

An initial effort in learning a parser producing a representation as rich as ULF is bound to face a data sparsity issue.[1] Thus a major goal in our choice of a transition-system-based parser has been to reduce the search space of the model. We investigate three additional methods of tackling this issue: (1) constraining actions in the decoding phase based on faithfulness to the ULF type system, (2) using a lexicon to limit the possible word-aligned symbols that the parser can generate, and (3) defining learnable features of the transition system state.

2 Unscoped Logical Form

Episodic Logic is an extension of first-order logic (FOL) that closely matches the form and ex-

[1]The training set in our initial release is only 1,378 sentences.

Proceedings of the 14th International Conference on Computational Semantics, pages 184–201
June 17–18, 2021.

pressivity of natural language, using reifying operators to enrich the domain of basic individuals and situations with propositions and kinds, keeping the logic first-order. It also uses other type-shifters, e.g., for mapping predicates to modifiers, and allows for generalized quantifiers (Schubert, 2000). ULF fully specifies the semantic type structure of EL by marking the types of the atoms and all of the predicate-argument relationships while leaving operator scope, anaphora, and word sense unresolved (Kim and Schubert, 2019). ULF is the critical first step to parsing full-fledged EL formulas. Types are marked on ULF atoms with a suffixed tag resembling the part-of-speech (e.g., .v, .n, .pro, .d for verbs, nouns, pronouns, and determiners, respectively). Names are instead marked with pipes (e.g. |John|) and a closed set of logical and macro operators have unique types and are left without a type marking. Each suffix denotes a set of possible semantic denotations, e.g. .pro always denotes an *entity* and .v denotes an *n-ary predicate* where *n* can vary. The symbol without the suffix or pipes is called the *stem*.

Type shifters in ULF maintain coherence of the semantic type compositions. For example, the type shifter adv-a maps a predicate into a verbal predicate modifier as in the prepositional phrase *"in my new shoes"* in Figure 1, as opposed to its predicative use *"A spider is in my new shoes"*.

The syntactic structure is closely reflected in ULF even under syntactic movement through the use of rewriting *macros* which explicitly mark these occurrences and upon expansion make the exact semantic argument structure available. Also, further resembling syntactic structure, ULFs are trees. The operators in operator-argument relations of ULF can be in first or second position, disambiguated by the types of the participating expressions. This further reduces the amount of word reordering between English and ULFs. The EL type system only allows function application for combining types, $\langle A, B\rangle, A \rightarrow B$, much like Montagovian semantics (Montague, 1970), but without type-raising.

3 Background

Currently, there is semantic parsing research occurring on multiple representational fronts, which is showcased by the cross-framework meaning representation parsing task (Oepen et al., 2019). The key differentiating factor of ULF from other meaning representations is the model-theoretic expressive capacity. To highlight this, here are a few limitations of notable representations: AMR (Banarescu et al., 2013a) neglects issues such as articles, tense, and nonintersective modification in favor of a canonicalized form that abstracts away from the surface structure; Minimal Recursion Semantics (Copestake et al., 2005) captures meta-level semantics for which inference systems cannot be built directly based on model-theoretic notions of truth and entailment; and extant semantic parsers for DRSs generate FOL-equivalent LFs, thus precludes proper treatment of phenomena such as generalized quantifiers, modification, and reification. Due to space limitations, we refer to Kim and Schubert (2019) for an in-depth description and motivation of ULF, including comparisons to other representations. We also refer to Schubert (2015) which places EL—the antecedent of ULF—in a broad context.

Our ULF parser development draws inspiration from the body of semantic parsing research on graph-based formalism of natural language, in particular, the recent advances in AMR parsing (Peng et al., 2018; Zhang et al., 2019a). The core organization of our parser is based on Peng et al. (2018), which uses a sequence-to-sequence model to predict the transition action sequence for a cache transition system with transition system features and hard attention alignment.

There are many transition-based parsers that were developed for parsing meaning representations (Zhang et al., 2016; Buys and Blunsom, 2017; Damonte et al., 2017; Hershcovich et al., 2017). These are mainly based on what's called an arc-eager parsing method, termed by Abney and Johnson (1991). Arc-eager parsing greedily adds edges between nodes before full constituents are formed, which keeps the partial graph as connected as possible during the parsing process (Nivre, 2004). They modify arc-eager parsing in various ways to generalize to the graph structures. Our transition system can be considered a modification of bottom-up arc-standard parsing due to restrictions on arc formation. While this leads to a longer action sequence for parsing, the parser's access to complete constituents allows promotion-based symbol generation for unary operators such as type shifters and standard bottom-up type analysis for constrained parsing.

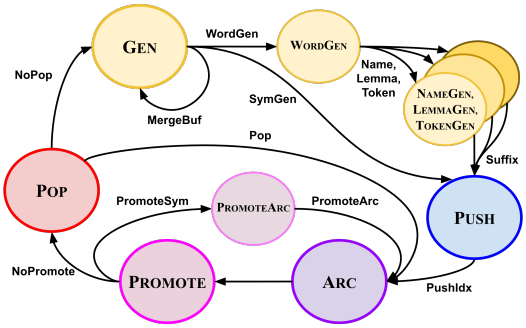


Figure 2: State transition diagram of the node generative transition system. Nodes in the figure are phases and edges are actions. An unlabeled edge means that this state transition occurs no matter what action is taken in that phase. The transition system starts in the GEN phase.

4 Our Transition System

Our transition system is a modification of the cache transition system (Gildea et al., 2018) which has been shown to be effective in AMR parsing (Peng et al., 2018). The distinctive aspect of our version is that the transition system generates nodes that are derived, but distinct, from the input sequence. We call it a node generative transition system. This eliminates the two-stage parsing framework of Peng et al. (2018). Our transition system also restricts the parses to be bottom-up to enable node generation and decoding constraints by the available constituents since ULF has an bottom-up compositional type system. The transition parser configuration is

$$C = (\sigma, \eta, \beta, G_p) \quad (1)$$

where σ is the stack, η is the cache, β is the buffer, and G_p is the partial graph. The parser is initialized with ([],[$, . . . , $],[$w_1, \ldots, w_n$],$\emptyset$), that is an empty stack, the cache with null values ($), the buffer with the input sequence of words, where each word is a token, lemma, POS tuple, $w_i = (t_i, l_i, p_i)$, and an empty partial graph, $G_p = (V_p, E_p)$, where V_p is ordered. A vertex, $v_i = (s_i, a_i) \in V$, is a pair of a ULF symbol s_i, and its alignment a_i—the index of the word from which s_i was produced. We will refer to the leftmost element in β as w_{next}.

While the size of the cache is a hyperparameter that can be set for the cache transition parser, we restrict the cache size to 2 in order to keep the oracle simple despite the newly added actions. This means that only tree structures can be parsed. In describing the transition system, we differentiate between *phases* and *actions*. Phases are classes of states in the transition system and the actions move between states. Figure 2 shows the full state transition diagram and shows how the phases dictate which actions can be taken and how actions move between phases. Actions may take variables to specify how to move into the next phase. Phases also determine which features go into the determining the next action. We will write phases in small caps (e.g. GEN) and actions in bold (e.g. **TokenGen**) for clarity.

The GEN and PROMOTE phases are novel to our transition system. The GEN phase generates graph vertices that are transformations of the buffer values. This allows us to put words of the input sentence in β instead of a pre-computed ULF atom sequence. The PROMOTE phase enables context-sensitive symbol generation. It generates unaligned symbols in the context of an existing constituent in the partial graph. (Use of logical operators without word alignments only makes sense with respect to something for the operators to act on.) We now describe each of the actions in the transition system. The following are parser actions that were almost directly inherited from the vanilla cache transition parser.

- **PushIndex**(i) pushes (i, v) onto σ, where v is the vertex currently at index i of η. Then it moves the vertex generated by the prior GEN phase to index i in η.
- **Arc**(i, d, l) forms an arc with label l in direction d (i.e. left or right) between the vertex at index i of the cache and the rightmost vertex in the cache. The **NoArc** action is used if no arc is made.
- **Pop** pops (i, v) from σ where i is the index of η which v came from. v is placed at index i of η and shifts the appropriate elements to the right.

We introduce two sets, S_p and S_s, which define the vocabulary of the two unaligned symbol generation actions: **PromoteSym** and **SymGen**, respectively. S_p consists of logical and macro operators that do not align with English words. S_s consists of symbols that could not be aligned in the training set and are not members of S_p.

4.1 Promotion-based Symbol Generation

PROMOTE includes a subordinate PROMOTEARC phase for modularizing the parsing decision. The following parsing actions are in this phase.

- **PromoteSym**(s_p) generates a promotion symbol,

Stack	Cache	Buffer	Edges	Actions taken
[]	[$, $]	[Do, you, want, to, see, me, ?]	$\emptyset$	—
[^{0}]	[$, do.aux-s]	[you, want, to, see, me, ?]	$\emptyset$	**Lemma**(aux-s); **Push**(0)
[^{0}]	[$, pres]	[you, want, to, see, me, ?]	E_1	**NoArc**; **PSym**(pres); **PArc**(arg0)
[^{0}]	[$, χ_0]	[you, want, to, see, me, ?]	E_2	**NoArc**; **PSym**(χ_0); **PArc**(ι)
[^{0}, ^{0}]	[χ_0, you.pro]	[want, to, see, me, ?]	E_2	**NoArc**; **NoP**; **Lemma**(pro); **Push**(0)
[^{0}]	[$, χ_0]	[want, to, see, me, ?]	E_3	**Arc**(0, R, arg0); **NoP**; **Pop**
[^{0}, ^{0}]	[χ_0, want.v]	[to, see, me, ?]	E_3	**NoArc**; **NoP**; **Lemma**(v); **Push**(0)
[^{0}, ^{0}, χ_0^0]	[want.v, to]	[see, me, ?]	E_3	**NoArc**; **NoP**; **Lemma**($\varnothing$); **Push**(0)
[^{0}, ^{0}, χ_0^0, want.v^0]	[to, see.v]	[me, ?]	E_3	**NoArc**; **NoP**; **Lemma**(v); **Push**(0)
[^{0}, ^{0}, χ_0^0, want.v^0, to^0]	[see.v, me.pro]	[?]	E_3	**NoArc**; **NoP**; **Token**(pro); **Push**(0)
[^{0}, ^{0}, χ_0^0, want.v^0]	[to, see.v]	[?]	E_4	**Arc**(0, R, arg0); **NoP**; **Pop**
[^{0}, ^{0}, χ_0^0]	[want.v, to]	[?]	E_5	**Arc**(0, R, arg0); **NoP**; **Pop**
[^{0}, ^{0}]	[χ_0, want.v]	[?]	E_6	**Arc**(0, R, arg0); **NoP**; **Pop**
[^{0}]	[$, χ_0]	[?]	E_7	**Arc**(0, R, arg1); **NoP**; **Pop**
[^{0}]	[$, χ_1]	[?]	E_8	**NoArc**; **PSym**(χ_1); **PArc**(ι)
[^{0}, ^{0}]	[χ_1, ?]	[]	E_8	**NoArc**; **NoP**; **Lemma**($\varnothing$); **Push**(0)
[^{0}]	[$, χ_1]	[]	E_9	**Arc**(0, R, arg0); **NoP**; **Pop**
[]	[$, $]	[]	E_9	**NoArc**; **NoP**; **Pop**

Figure 3: Example run of the transition system running on the sentence *"Do you want to see me?"* from our parser. The left four columns show the parser configuration after taking the actions shown in the rightmost column. We make the following modifications for brevity. When a **WordGen** action takes place, it is always followed by one of **Name**, **Lemma**, or **Token** and then a **Suffix**(e) action. Thus we omit the **WordGen** and **Suffix** actions and transfer the argument of **Suffix** to the **Name**, **Lemma**, or **Token** action. "Promote" is abbreviated as "P" (e.g., **PromoteSym** as **PSym**) and **PushIdx** as **Push**. Stack item indices (i, v) are written as v^i instead. χ and ι stand for COMPLEX and INSTANCE which are the special node and edge labels, respectively, for constructing non-atomic ULF operators in penman format. Edge labels arg0 and arg1 simply indicate the argument position in ULF. $E_n = \{e_i \mid 0 \leq i < n\}$ where $e_0 = (\text{do.aux-s} \xleftarrow{arg0} \text{pres})$, $e_1 = (\text{pres} \xleftarrow{\iota} \chi_0)$, $e_2 = (\chi_0 \xrightarrow{arg0} \text{you})$, $e_3 = (\text{see.v} \xrightarrow{arg0} \text{me.pro})$, $e_4 = (\text{to} \xrightarrow{arg0} \text{see.v})$, $e_5 = (\text{want.v} \xrightarrow{arg0} \text{to})$, $e_6 = (\chi_0 \xrightarrow{arg0} \text{want.v})$, $e_7 = (\chi_0 \xleftarrow{\iota} \chi_1)$, $e_8 = (\chi_1 \xrightarrow{arg0} ?)$.

$s_p \in S_p$, appends the vertex (s_p, NONE) to V_p, and proceeds to the PROMOTEARC phase.

- **NoPromote** skips the PROMOTE phase and proceeds to the POP phase.
- **PromoteArc**(l) makes an arc from the last added vertex, v_p, to the vertex at the rightmost position of the cache, v_{η_r}, by adding (v_p, v_{η_r}, l) to E_p. v_p then takes the place of v_{η_r} in the cache and v_{η_r} is no longer accessible by the transition system. The system proceeds to the ARC phase.

4.2 Sequential Symbol Generation

We replace the **Shift** action with the GEN phase to generate ULF atoms based on the tokenized text input. This phase allows the parser to generate a symbol using w_{next} as a foundation, or generate an arbitrary symbol that is not aligned to any word in β. GEN includes subordinate phases WORDGEN, LEMMAGEN, TOKENGEN, and NAMEGEN for modularizing the decision process.

- **WordGen** proceeds to WORDGEN phase, in which the following actions are available.
  1. **Name** proceeds to the NAMEGEN phase.
  2. **Lemma** proceeds to the LEMMAGEN phase.
  3. **Token** proceeds to the TOKENGEN phase.
- **Suffix**(e) is the only action available in the NAMEGEN, LEMMAGEN, and TOKENGEN phases. It generates a symbol s consisting of a stem and suffix extension e from w_{next}. In the NAMEGEN phase, the stem is t_{next} with surrounding pipes; in the TOKENGEN phase, the stem is t_{next}; and in the LEMMAGEN phase, the stem is l_{next}. (s, i) where i is the index of w_{next} is added to V_p and we move forward one word in β. The system proceeds to the PUSH phase.
- **SymGen**(s) adds an unaligned symbol (s, NONE) to V_p and proceeds to the PUSH phase.
- **SkipWord** skips word in β and returns to the GEN phase.
- **MergeBuf** takes w_{next} and merges it with the word after it $w_{\text{next}+1}$. This is stored at the front of the buffer as a pair $(v_\beta, v_{\beta+1})$. This forms a single stem with a space delimiter in the NAMEGEN phase and an underscore delimiter in the LEMMAGEN and TOKENGEN phases. The system returns to the GEN phase. This is used to handle multi-word expressions (e.g. *"had better"*).

The transition system begins in the GEN phase.

4.3 Oracle Extraction Algorithm

In order to train a model of the parser actions, we need to extract the desired action sequences from gold graphs. We modify the oracle extraction algorithm for the vanilla cache transition parser, described by Gildea et al. (2018). The oracle starts with a gold graph $G_g = (V_g, E_g)$ and maintains the partial graph $G_p = (V_p, E_p)$ of the parsing process, where V_g is sequenced by the preorder traversal of G_g. The oracle maintains s_{next}, the symbol in the foremost vertex of V_g that has not yet been added to G_p. The oracle begins with a transition system configuration, C, initialized with the input sequence, $w_1, ..., w_n$.

The oracle is also provided with an approximate alignment, $A = \{(w_i, v_j) \mid 1 \leq i \leq n, 1 \leq j \leq m\}$, between the input sequence, $w_{i:n}$, to the nodes in the gold graph, $V_g, |V_g| = m$, which is generated with a greedy matching algorithm. The matching algorithm uses a manually-tuned similarity heuristic built on the superficial similarity of English words, POS, and word order to the stems, suffixes, and preorder positions of the corresponding ULF atoms. A complete description of the alignment algorithm is in appendix B. This alignment is not necessary to maintain correctness of the oracle, but it is used to cut the losses when the input words become out of sync with the gold graph vertex order.[2] Steps 5-7 of the GEN phase uses the alignments to identify whether the buffer or the vertex order is ahead of the other and appropriately sync them back together.

The oracle uses the following procedure, broken down by parsing phase, to extract the action sequence to build the $G_p = G_g$ with C and A.

- GEN phase: Let $b = \text{Stem}(s_{\text{next}})$, $e = \text{Suffix}(s_{\text{next}})$, $n = \text{IsName}(s_{\text{next}})$.[3]
  1. If n and $t_{\text{next}} = b$, **NameGen**(e)
  2. If not n and $t_{\text{next}} =_i b$, **TokenGen**(e)
  3. If not n and $l_{\text{next}} =_i b$, **LemmaGen**(e)
  4. **MergeBuf** if
     n and $\text{Pre}(\text{Concat}(t_{\text{next}}, \text{“ ”}, t_{\text{next+1}}), b)$ or
     not n and $\text{Pre}_i(\text{Concat}(l_{\text{next}}, \text{“\_”}, l_{\text{next+1}}), b)$ or
     not n and $\text{Pre}_i(\text{Concat}(t_{\text{next}}, \text{“\_”}, t_{\text{next+1}}), b)$
  5. If $(w_i, v_{\text{next}}) \in A$ for w_i before w_{next} or $v_{\text{next}} \in S_s$, then **SymGen**(v_{next})
  6. If $(w_{\text{next}}, v_j) \in A$ for v_j which comes after v_{next} or $v_j \in V_p$, then **SkipWord**.
  7. Otherwise, **SymGen**(v_{next})

  Step 5-7 allow the oracle to handle the generation of symbols that are not in word order, by skipping any words that come earlier than the symbol order; and generating symbols that cannot be aligned with **SymGen** for any reason.
- PUSH phase: The push phase of the vanilla cache transition parser's oracle—viz., choosing the cache position whose closest edge into β is farthest away—is extended to account not only for direct edges, but also for paths that include only unaligned-symbols.[4]
- ARC phase: The vanilla cache transition system's rule of generating the ARC action for any edge, $e \in E_g \wedge e \notin E_p$ between the rightmost cache position and the other positions, is extended to also require the child vertex to be fully formed. That is, for the vertex v_{child}, $|\text{descendants}(v_{\text{child}}, G_g)| = |\text{descendants}(v_{\text{child}}, G_p)|$. This enforces bottom-up parsing, which is necessary for both the promotion-based symbol generation and type composition constraint.
- PROMOTE phase: If the vertex in the rightmost cache position, v_{η_r}, is fully formed ($|\text{descendants}(v_{\eta_r}, G_g)| = |\text{descendants}(v_{\eta_r}, G_p)|$) and has a parent node in the PROMOTE lexicon ($\text{label}(\text{parent}(v_{\eta_r}, G_g)) \in S_p$), then the parser generates the action sequence **PromoteSym**($\text{parent}(v_r, G_g)$), **PromoteArc**(l_p) where l_p is the label for the edge from the parent of v_{η_r} to v_{η_r} in G_g ($\text{EdgeLabel}(\text{parent}(v_{\eta_r}, G_g), v_{\eta_r}, G_g)$).

5 Model

Our model has three basic components: (1) a word sequence encoder, (2) a ULF atom sequence encoder, and (3) an action decoder, all of which are

[2] When the words become out-of-sync with the gold graph the oracle must rely on **SymGen** to generate the graph nodes. Since **SymGen** requires selecting the correct value out the entire vocabulary of ULF atoms, it is much more difficult to predict correctly than **NameGen**, *TokenGen*, and *LemmaGen* which require only selecting the correct type tag.

[3] $=$ is string match, $=_i$ is case-insensitive string match, Pre determines whether its first argument is a prefix of the second and Pre_i is the case-insensitive counterpart.

[4] The motivation for this is that if only unaligned symbols exist in the path, the full path can be made without changing the relative status of any other node in the transition system. Let v_1 and v_2 be the end points of the path. With v_1 in the cache and the word aligned to v_2, $w_{v_2} = w_{\text{next}}$, **SymGen** and PROMOTE can generate all nodes in the path without interacting with the rest of the transition system.

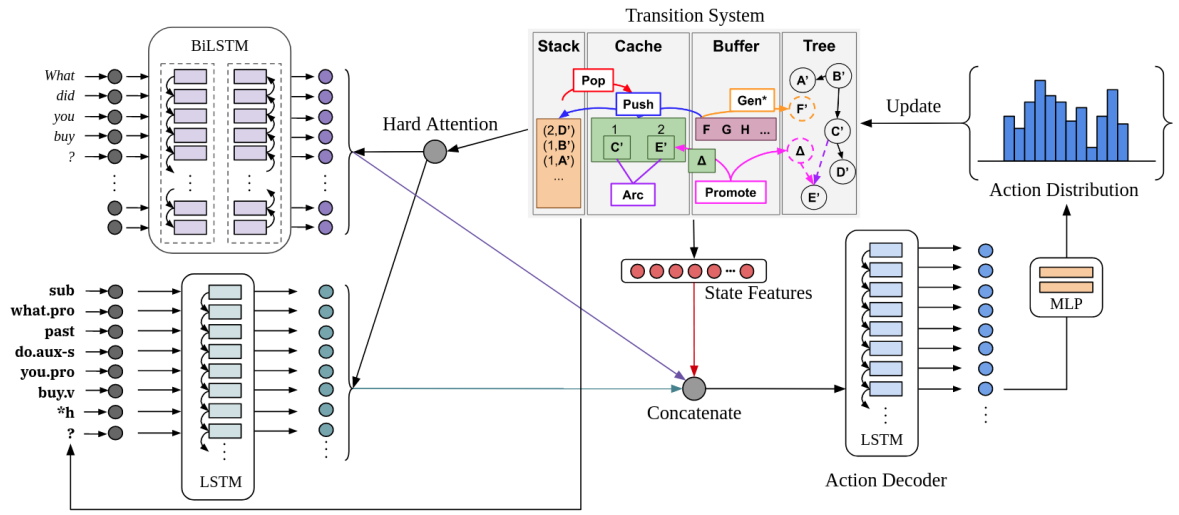


Figure 4: The model consists of a sentence-encoding BiLSTM, a symbol-encoding LSTM, and an action-decoding LSTM. New symbols generated in the GEN and PROMOTE phases of the transition system are appended to the symbol sequence. The transition system supplies hard attention pointers that select the relevant word and symbol embeddings. These are concatenated with the transition state feature vector and supplied as input to the action decoder, which predicts the next action that updates the transition system.

LSTMs. During decoding, the transition system configuration, C, is updated with decoded actions and used to organize the action decoder inputs using the sequence encoders. The system models the following probability

$$P(a_{1:q}|w_{1:n}) = \prod_{t=1}^{q} P(a_t|a_{1:t-1}, w_{1:n}; \theta) \quad (2)$$

where $a_{1:q}$ is the action sequence, $w_{1:n}$ is the input sequence, and θ is the set of model parameters. Figure 4 is a diagram of the full model structure.

5.1 Word and Symbol Sequence Encoders

The input word embedding sequence $w_{1:n}$ is encoded by a stacked bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with L_w layers. Each word embedding sequence is a concatenation of embeddings of GloVe (Pennington et al., 2014), lemmas, part-of-speech (POS) and named entity (NER) tags, RoBERTa (Liu et al., 2019), and features learned by a character-level convolutional neural network (CharCNN, Kim et al., 2016). As ULF symbols are generated during the parsing process, the symbol embedding sequence $s_{1:m}$, which is the concatenation of a symbol-level learned embedding and the CharCNN feature vector over the symbol string, is encoded by a stacked *uni*directional LSTM of L_s layers.

5.2 Hard Attention

Peng et al. (2018) found that for AMR parsing with cache transition systems, a hard attention mechanism, tracking the next buffer node position and its aligned word, works better than a soft attention mechanism for selecting the embedding used during decoding. We take this idea and modify the tracking mechanism to find the most relevant word, w_i, and symbol, s_j, for each phase.

- ARC and PROMOTE*: The symbol s_j in the rightmost cache position and aligned word w_i.
- PUSH: The symbol s_j generated in the previous action and aligned word w_i.
- Otherwise: The last generated symbol s_j and the word w_i in the leftmost β position.

This selects the output sequences $\boldsymbol{h}_{w_i}^{L_w}$ and $\boldsymbol{h}_{s_j}^{L_s}$ from the encoders for the action decoder.

5.3 Transition State Features

Similar to Peng et al. (2018), we extract features from the current transition state configuration, C, to feed into the decoder as additional input in the form of learned embeddings

$$\boldsymbol{e}_f(C) = [\boldsymbol{e}_{f_1}(C); \boldsymbol{e}_{f_2}(C); ...; \boldsymbol{e}_{f_l}(C)] \quad (3)$$

where $\boldsymbol{e}_{f_k}(C)$ $(k = 1, ..., l)$ is the k-th feature embedding, with l total features. Our features, which are heavily inspired by Peng et al. (2018), are as follows.

- Phase: An indicator of the phase in the transition system.
- POP, GEN features: *Token features*[5] of the rightmost cache position and the leftmost buffer position; the number of rightward dependency edges from the cache position word and the first three of their labels; and the number of outgoing ULF arcs from the cache position and their first three labels.
- ARC, PROMOTE features: For the two cache positions, their token features and the word, symbol[6], and dependency distance between them; furthermore, their first three outgoing and single incoming dependency arc labels and their first two outgoing and single incoming ULF arc labels.
- PROMOTEARC features: Same as the PROMOTE features but for the rightmost cache position use the node/symbol generated in the preceding **PromoteSym** action.
- PUSH features: Token features for the leftmost buffer position and all cache positions.

5.4 Action Encoder/Decoder

The action sequence is encoded by a stacked unidirectional LSTM with L_a layers where the action input embeddings, $\boldsymbol{h}_{a_{1:q}}$ are concatenations of the word and symbol encodings.

$$\boldsymbol{h}_{a_k} = [\boldsymbol{h}^{L_w}_{w_i}; \boldsymbol{h}^{L_s}_{s_j}; \boldsymbol{e}_f(C)] \tag{4}$$

The state features $\boldsymbol{h}^{L_a}_{a_k}$ are then decoded into prediction weights with a linear transformation and ReLU non-linearity.

6 Parsing

The model is trained on the cross-entropy loss of the model probability (2) using the oracle action sequence. Both training and decoding are limited to a maximum action length of 800. For the training set the oracle has an average action length of 134 actions and a maximum action length of 1477.

6.1 Constrained Decoding

We investigate two methods of constraining the decoding process with prior knowledge of ULF to overcome the challenge of using a small dataset. These automatic methods filter out clearly incorrect choices at the cost of some decoding speed and further tailor the parser to ULFs.

ULF Lexicon To improve symbol generation, we introduce a lexicon with possible ULF atoms for each word. Nouns, verbs, adjectives, adverbs, and preposition entries are automatically converted from the Alvey lexicon (Carroll and Grover, 1989) with some manual editing. Pronouns, determiners, and conjunctions entries are extracted from Wiktionary[7] category lists. Auxiliary verbs entries are manually built from our ULF annotation guidelines. When generating a word-aligned symbol the stem is searched in the lexicon. If the string is present in the lexicon, only corresponding symbols in the lexicon are allowed to be generated. Since the lexicon is not completely comprehensive, this constraint may lead to some additional errors.

Type Composition The type system constraint adds a list of types, T_v, to accompany $|V_p|$ (the vertices of the partial graph), which stores the ULF type of each vertex. When a vertex, v, is added to G_p, its ULF type, t_v is added to T_v. This ULF type system is generalized with placeholders for macros and each stage in processing them. When the parser predicts an arc action during decoding, the types source, t_s, and target, t_t nodes are run through a type composition function. If the types can compose, $t_c = (t_s.t_t), t_c \neq \emptyset$, the type of the source node is replaced with t_c. Otherwise, the resulting C is not added to the search beam.

7 Experiments

We ran our experiments on a hand-annotated dataset of ULFs totaling 1,738 sentences (1,378 train, 180 dev, 180 test). The dataset is a mixture of sentences from crowd-sourced translations, news text, a question dataset, and novels. The distribution of sentences leans towards more questions, requests, clause-taking verbs, and counterfactuals because a portion of the dataset comes from the dataset used by Kim et al. (2019) for generating inferences from ULFs of those constructions.

The data is split by segmenting the dataset into 10 sentence segments and distributing them in a round-robin fashion, with the training set receiving eight chunks in each round. This splitting method is designed to allow document-level topics to distribute into splits while limiting any performance inflation of the dev and test results that can result

[5]The token features are the ULF symbol and the word, lemma, POS, and NER tags of the aligned index of the input.

[6]Symbol distance is based on the order in which the symbols are generated by the parser.

[7]https://en.wiktionary.org/

when localized word-choice and grammatical patterns are distributed into all splits.

Kim and Schubert (2019) found that interannotator agreement (IA) on ULFs using the EL-SMATCH metric (Kim and Schubert, 2016) is 0.79.[8] We add a second pass to further reduce variability in our annotations.[9] Further details about the dataset are available in appendix A and the complete annotation guidelines are available as part of the dataset.

ULF-AMR In order to use parsing and evaluation methods developed for AMR parsing (Banarescu et al., 2013a), we rewrite ULFs in penman format (Kasper, 1989) by introducing a node for each ULF atom and generating left-to-right arcs in the order that they appear (:ARG0, :ARG1, etc.), assuming the leftmost constituent is the parent. In order to handle non-atomic operators in penman format which only allows atomic nodes, we introduce a COMPLEX node with an :INSTANCE edge to mark the identity of the non-atomic operator.

Setup We evaluate the model with SEMBLEU (Song and Gildea, 2019), a metric for parsing accuracy of AMRs (Banarescu et al., 2013b). This metric extends BLEU (Papineni et al., 2002) to node- and edge-labeled graphs. We also measure EL-SMATCH, a generalization of SMATCH to graphs with non-atomic nodes, for analysis of the model since it has F1, precision, and recall components.

The tokens, lemmas, POS tags, NER tags, and dependencies are all extracted using the Stanford CoreNLP toolkit (Manning et al., 2014). In all experiments the model was trained for 25 epochs. Starting at the 12th epoch we measured the SEMBLEU performance on the dev split with beam size 3. Hyperparameters were tuned manually on the dev split performance of a smaller, preliminary version of the annotation corpus. We use RoBERTa-Base embeddings with frozen parameters, 300 dimensional GloVe embeddings, and 100 dimensional t_i, l_i, p_i, action, and symbol embeddings. The word encoder is 3 layers. The symbol encoder and action decoder are 2 layers. Experiments were run on a single NVIDIA Tesla K80 or GeForce RTX 2070 GPU. Training the full model takes about 6 hours. The full tables of results and default parameters are available in appendix D.

[8] cf. AMR is reported to have about 0.8 IA using the SMATCH metric (Tsialos, 2015)

[9] We did not measure IAA on our dataset and take the prior report as an lower-end estimate given the similarity of our annotations methods and our additional review phase. Our annotation process was collaborative and result in a single annotation per sentence so IAA cannot be measured.

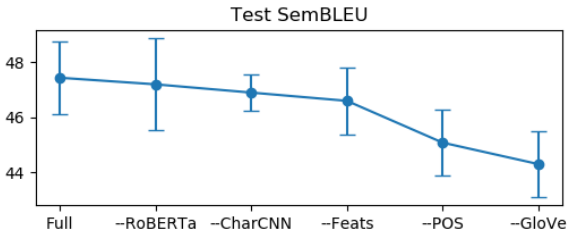


Figure 5: Ablation tests with standard deviation error bars of 5 runs of different random seeds.

7.1 Results

Ablations In our ablation tests, the model from the training epoch with the highest dev set SEMBLEU score is evaluated on the test split with beam size 3.[10] The results are shown in Figure 5.

CharCNN and RoBERTa are the least important components—to the point that we cannot conclude that they are of any benefit to the model due to the large overlap in the performance of models with and without them. The GloVe, POS, and feature embeddings are more important. The importance of POS is not surprising given the tight correspondence between POS tags and ULF type tags.

Model	SEMBLEU	EL-SMATCH
(Zhang et al., 2019a)	12.3	34.3
(Cai and Lam, 2020)	34.2	52.6
Our best model	**47.4**	**59.8**

Table 1: Comparison to AMR parsers.

Comparison to Baselines We compare our parser performance against two AMR parsers with minimal AMR-specific assumptions. The major recent efforts by the research community in AMR parsing make these parsers strong baselines. Specifically, we compare against the sequence-to-graph (STOG) parser (Zhang et al., 2019a) and Cai and Lam's (2020) graph-sequence iterative inference (GS) parser.[11] The ULF dataset is preprocessed for these parsers by stripping pipes from names to support the use of a copy mechanism and splitting node labels with spaces into multiple nodes to make the labels compatible with their data

[10] Our initial experiments re-evaluated the top-5 choices with a beam size of 10, but we found that the performance consistently degraded and abandoned this step.

[11] We do not compare our model against the existing rule-based ULF parsers since they are domain specific and cannot handle the range of sentences that appear in our dataset.

pipelines. Table 1 shows the results.[12] The STOG parser fares poorly on both metrics. A review of the results revealed that the parser struggles with node prediction in particular. This is likely the result of the dataset size not properly supporting the parser's latent alignment mechanism.[13] The GS parser performs better than the STOG parser by a large margin, but is still far from our parser's performance. The GS parser also struggles with node prediction, but is more successful in maintaining the correct edges in spite of incorrect node labels.

Investigating the dev set results reveals that our parser is quite successful in node generation, since by design the node generation process reflects the design of ULF atoms. Despite the theoretical capacity to generate node labels without a corresponding uttered word or phrase, our parser only does this for common logical operators such as reifiers and modifier constructors. The GS parser on the other hand, is relatively successful on node labels without uttered correspondences, correctly generating the elided "you" in imperatives and the logical operators ! and multi-sent which indicate imperatives and multi-sentence annotations, respectively. Our parser also manages to correctly generate a variety of verb phrase constructions, but fails to recognize reified infinitives as arguments of less frequent clausal verbs such as "neglect", "attach", etc. (as opposed to "have", "tell") and instead interprets "to" as either an argument-marking prepositions or reification of an already reified verb. Examples of parses and a discussion of specific errors are omitted here due to space constraints and provided in appendix E.

Constrained Decoding When evaluating decoding constraints, we select the model by re-running the five best performing epochs with constraints. When using the type composition constraint, we additionally increase the beam size to 10 so that the parser has backup options when its top choices are filtered out. Table 2 presents these results. We see a increase in precision for +Lex, but a greater loss in recall. +Type reduces performance on all metrics. Due to the bottom-up parsing procedure, a filtering of choices can cascade into fragmented parses. The outputs for an arbitrarily selected run of the model has on average 2.9 fragments per sentence when decoding with the type constraint and 1.4 without. This and the relative performance on the precision metric suggest that constraints improve individual parsing choices, but are too strict, leading to fragmented parses.

	SEMBLEU	EL-SMATCH		
		F1	Precision	Recall
Full	**47.4**	**59.8**	60.7	**59.0**
+Lex	46.2	57.5	**61.5**	54.1
+Type	40.0	55.8	59.1	52.8

Table 2: Statistics of model performances with constraints added—the average of 5 runs.

Dependence on Length To investigate the performance dependence on the problem size, we partition the test set into quartiles by oracle action length. The 0 seed of our full model has SEMBLEU scores of 52, 47, 48, and 31 on the quartiles of increasing length. As expected, the parser performs better on shorter tasks. The parser performance is relatively stable until the last quartile. This is likely due to a long-tail of sentence lengths in our dataset. This last quartile includes sentences with oracle action length ranging from 148 to 1474.

8 Conclusion

We presented the first annotated ULF dataset and the first parser trained on such a dataset. We showed that our parser is a strong baseline, outperforming existing semantic parsers from a similar task. Surprisingly, our experiments showed that even in this low-resource setting, constrained decoding with a lexicon or a type system does more harm than good. However, the symbol generation method and features designed for ULFs result in a performance lead over using an AMR parser with minimal representational assumptions.

We hope that releasing this dataset will spur other efforts into improving ULF parsing. This of course includes expanding the dataset, using our comprehensive annotation guidelines and tools; but we see many additional avenues of improvement. The type grammar opens up many promising possibilities: sampling of silver data (in conjunction with ULF to English generation (Kim et al., 2019)), use as a weighted constraint, or direct incorporation into a model to avoid the pitfalls we observed in our simple approach to semantic type enforcement.

[12] Our parser gets the exact ULF for 6 out of the 180 sentences (3.3%). They were all yes-no questions which tend to be a bit shorter than informative declarative sentences (e.g. *"Can't you do something?"*).

[13] The STOG parser is improved by (Zhang et al., 2019b) with about 1 point of improvement on SMATCH. Unfortunately, the code for this parser is not released to the public.

9 Acknowledgments

This work was supported by NSF EAGER grant NSF IIS-1908595, DARPA CwC subcontract W911NF-15-1-0542, and a Sproull Graduate Fellowship from the University of Rochester. We are grateful to the anonymous reviewers for their helpful feedback.

References

Steven P Abney and Mark Johnson. 1991. Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20(3):233–250.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013a. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013b. 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.

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.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, 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.

J. Carroll and C. Grover. 1989. The derivation of a large computational lexicon of english from LDOCE. In Boguraev B. and Briscoe E., editors, *Computational Lexicography for Natural Language Processing*, pages 117–134. Longman, Harlow, UK.

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.

Daniel Gildea, Giorgio Satta, and Xiaochang Peng. 2018. Cache transition systems for graph parsing. *Computational Linguistics*, 44(1):85–118.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Vancouver, Canada. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

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, Mandar Juvekar, and Lenhart Schubert. 2020. Monotonic inference for underspecified episodic logic. In *Proceedings of the 1st Workshop on Natural Logic Meets Machine Learning (NALOMA)*. Association for Computational Linguistics.

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.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

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.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on 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*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

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.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the workshop on incremental parsing: Bringing engineering and cognition together*, pages 50–57.

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. 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.

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.

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 Schubert. 2015. Semantic representation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4132–4138. AAAI Press.

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.

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.

Aristeidis Tsialos. 2015. Abstract meaning representation for sembanking. Available at `www.inf.ed.ac.uk/teaching/courses/tnlp/2014/Aristeidis.pdf`, accessed December 8, 2018.

Florian Wolf. 2005. *Coherence in natural language : data structures and applications*. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Brain and Cognitive Sciences.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 421–431, Berlin, Germany. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. 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 3786–3798, Hong Kong, China. Association for Computational Linguistics.

A Dataset Details

We chose a variety of text sources for constructing this dataset to reduce genre-effects and provide good coverage of all the phenomena we are investigating. Some of these datasets include annotations, which we use only to identify sentence and token boundaries. The dataset includes 1,738 sentences, with a mean, median, min, and max sentence lengths of 10.275, 8, 2, and 128 words, respectively.

A.1 Data Sources

- **Tatoeba**

  The Tatoeba dataset[14] consists of crowd-sourced translations from a community-based educational platform. People can request the translation of a sentence from one language to another on the website and other members will provide the translation. Due to this pedagogical structure, the sentences are fluent, simple, and highly-varied. The English portion downloaded on May 18, 2017 contains 687,274 sentences.

- **Discourse Graphbank**

  The Discourse Graphbank (Wolf, 2005) is a discourse annotation corpus created from 135 newswire and WSJ texts. We use the discourse annotations to perform sentence delimiting. This dataset is on the order of several thousand sentences.

- **Project Gutenberg**

  Project Gutenberg[15] is an online repository of texts with expired copyright. We downloaded the top 100 most popular books from the 30 days prior to February 26, 2018. We then ignored books that have non-standard writing styles: poems, plays, archaic texts, instructional books, textbooks, and dictionaries. This collection totals to 578,650 sentences.

- **UIUC Question Classification**

  The UIUC Question Classification dataset (Li and Roth, 2002) consists of questions from the TREC question answering competition. It covers a wide range of question structures on a wide variety of topics, but focuses on factoid questions. This dataset consists of 15,452 questions.

[14] https://tatoeba.org/eng/
[15] https://www.gutenberg.org

Most of the dataset is annotated by random selection of a single or some contiguous sequence of sentences by annotators. However, part of the annotated dataset comes from inference experiments run by Kim et al. (2019) regarding questions, requests, counterfactuals, and clause-taking verbs. Therefore, the dataset has a bias towards having these phenomena at a higher frequency than expected from a random selection of English text.

A key issue regarding the dataset is its difficulty. We primarily quantify this with the AMR parser baseline, the sequence-to-graph (STOG) parser (Zhang et al., 2019a), in the main text, which performs quite poorly on this dataset. Its performance indicates that the patterns in this dataset are too varied for a modern parsing model to learn without built in ULF-specific biases. Although, part of this is due to the size of the dataset, if the dataset consisted only of short and highly-similar sentences, we would expect a modern neural model, such as the AMR baseline, to be able to learn successful parsing strategy for it.

This reflects the design of the dataset construction. Although the dataset indeed includes many short sentences, especially from the Tatoeba and UIUC Question Classification datasets, the sentences cover a wide range of styles and topics. The Tatoeba dataset is built from a crowd-sourced translation community, so the sentences are not limited in genre and style and has a bias toward sentences that give people trouble when learning a second language. We consider this to be valuable for a parsing dataset since, while the sentences from Tatoeba are usually short, they vary widely in topic and tend to focus on tricky phenomena that give language-learners—and likely parsers—trouble. Sentences from the Discourse Graphbank (news text) and Project Gutenberg (novels) further widen the scope of genres and styles in the dataset. This should make it difficult for a parsing model to overfit to dataset distribution. The dataset also has a considerable representation of longer sentences ($\sim$10% of the dataset is $>$20 words) including dozens of sentences exceeding 40 words, reaching up to 128 words.

A.2 Annotation Interface & Interannotator Agreement

We use the same annotation interface as Kim and Schubert (2019), which includes (1) syntax and bracket highlighting, (2) a sanity checker based on the underlying type grammar, and (3) uncertainty marking to trigger a review by a second annotator. The complete English-to-ULF annotation guideline is attached as a supplementary document.

Kim and Schubert (2019) reports interannotator agreement (IA) of ULF annotations using this annotation procedure. In summary, they found that agreement among sentences that are marked as *certain* are 0.79 on average and can be up-to 0.88 when we filter for well-trained annotators. For comparison, AMR annotations have been reported to have annotator vs consensus IA of 0.83 for newswire text and 0.79 for webtext using the *smatch* metric (Tsialos, 2015).

In order to mitigate the issue of low agreement of some annotators in the IA study, each annotation in our dataset was reviewed by a second annotator and corrected if necessary. There was an open discussion among annotators to clear up uncertainty and handle tricky cases during both the original annotation and the reviewing process so the actual dataset annotations are more consistent than the test of IA agreement (which had completely independent annotations) would suggest.

A.3 Dataset Splits

The data split is done by segmenting the dataset into 10 sentence segments and distributing them in a round-robin fashion, with the training set receiving eight chunks in each round. This splitting method is designed to allow document-level topics to distribute into splits while limiting any performance inflation of the dev and test results that can result when localized word-choice and grammatical patterns are distributed into all splits.

The Tatoeba dataset further exacerbates the issue of localized word-choice and grammatical patterns since multiple sentences using the same phrase or grammatical construction often appear back-to-back. We suspect that this is because the Tatoeba dataset is ordered chronologically and users often submit multiple similar sentences in order to help understand a particular phrase or grammatical pattern in a language that they are learning.

B Full ULF Alignment Details

The ULF-English alignment system takes into account the similarity of the English word to the ULF atom without the type extension, the similarity of the type extension with the POS tag, and the relative distance of the word and symbol in question.

Given a sentence $s = w_{1:n}$, which is tokenized, $t_{1:n}$, lemmatized, $l_{1:n}$, and POS tagged, $p_{1:n}$, a set of symbols that are never aligned S_u, and a list of ULF atoms $a_{1:m}$, which can be broken up into the base stems, $b_{1:m}$, and suffix extensions, $e_{1:m}$, in order of appearance in the formula (i.e. DFS preorder traversal), the word/atom similarity is defined using the following formulas.

$$\text{Sim}(w, a) = \max(\text{Olap}(t, b), \text{Olap}(l, b)) + 0.5 * (\text{Olap}(p, e) + (1 - |\text{RL}(w, n) - \text{RL}(a, m)|))$$

where token overlap, Olap, is defined as

$$\text{Olap}(x, y) = \frac{2 * |\text{MaxSharedSubstr}(x, y)|}{|x| + |y|}$$

and relative location RL is defined as

$$\text{RL}(x, n) = \frac{\text{IndexOf}(x)}{n}$$

Next, in order of $\text{Sim}(w, a)$, we consider each word-atom pair, (w_i, a_i), $1 \leq i \leq n$ until $\text{Sim}(w, a) <$ MinSim, where MinSim is set to 1.0, based on cursory results. We further disregard any alignments that include an atom which shouldn't be aligned (a_i s.t. $a_i \in S_u$). We assume that spans of words align to connected subgraphs, so we cannot accept all word-atom pairs. An word-atom pair, (w_i, a_i), is accepted into the set of token alignments, A_t, if and only if the following conditions are met:

1. w_i has no alignments or a_i is connected to an atom, a', that is already aligned to w_i.
2. a_i is not in any other alignment or w_i is adjacent to another, w' which is already aligned to a_i.

The token-level (word-atom) alignment, A_t, is then converted to connected (span-subgraph) alignment, A. This is done with the following algorithm:

1. For every atom a_i in one of the aligned pairs of A_t, merge all of the words aligned to a_i into a single span, s_i. During the initial alignment, we ensured that these words would form a span.
2. Merge all overlapping spans into single spans and collect the set of atoms that are aligned to each of these spans into a subgraph.[16] These collected subgraphs will be connected because we ensured that for any word the nodes that it is aligned to forms a connected subgraph.

[16]This can be done in $\mathcal{O}(n \log n)$ time by sorting the spans, then doing a single pass of merging overlapping elements.

C RoBERTa Handling Details

Except for RoBERTa, all other embeddings are fetched from their corresponding learned embedding lookup tables. RoBERTa uses OpenAI GPT-2 tokenizer for the input sequence and segments words into subwords prior to generating embeddings, which means one input word may correspond to multiple hidden states of RoBERTa. In order to accurately use these hidden states to represent each word, we apply an average pooling function to the outputs of RoBERTa according to the alignments between the original and GPT-2 tokenized sequences.

D Full Tables

Tables of the full set of raw results and parameters are presented in this section. Table 3 shows the ablations on the model without decoding constraints. This is the basis of Figure 5 in the main text. Table 4 shows the performance change with the lexicon constraint and Table 5 shows the performance change with the composition constraint. These tables are the basis of Table 2 in the main text. Our experiments with the lexicon constraint were more extensive since the type constraint takes considerably longer to run due to requiring a larger beam size and more computational overhead. Table 7 presents all of the model parameters in our experiments.

E Parse Examples

Figure 6 shows six parse examples of our parser and the GS parser in reference to the gold standard. Generally, we find that our parser does much better on node generation for nodes that correspond to an input word. For example, the GS parser on example 1 uses *(plur *s)* for the word "speech" and *iron.n* for the words "silver" and "silence". This isn't to say that our parser doesn't make mistakes. But the mistakes are not as open-ended. For example, our parser mistakenly annotates "silver" as a noun in example 1 when in fact it should be an adjective (compared against "golden"). The GS parser seems to pick the closest word in its vocabulary, which is generated from the training set and closed. This leads to strange annotations like *iron.n* for the word "silence". If there is nothing close available, then it can derail the entire parse. In example 4, the GS parser is unable to find a node label for the word "device" which derails the parse to generate *(mod-n*

Ablation	SEMBLEU		EL-SMATCH					
			F1		Precision		Recall	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Full	46.4 ± 1.4	$\mathbf{47.4} \pm 1.3$	58.4 ± 0.7	$\mathbf{59.8} \pm 1.0$	59.1 ± 1.1	$\mathbf{60.7} \pm 1.5$	57.8 ± 0.5	$\mathbf{59.0} \pm 0.7$
-RoBERTa	45.5 ± 2.4	47.2 ± 1.7	58.3 ± 1.4	59.3 ± 1.0	59.1 ± 1.6	60.5 ± 1.1	57.5 ± 1.2	58.3 ± 0.9
-CharCNN	46.4 ± 1.0	46.9 ± 0.7	58.8 ± 0.8	59.3 ± 0.4	59.4 ± 1.3	60.1 ± 0.5	58.1 ± 0.6	58.5 ± 0.5
-$e_f(C)$ Feats	47.0 ± 1.2	46.6 ± 1.2	58.6 ± 0.5	58.8 ± 1.1	60.4 ± 1.2	60.2 ± 1.1	56.9 ± 0.4	57.4 ± 1.2
-POS	43.8 ± 1.7	45.1 ± 1.2	56.9 ± 1.1	58.3 ± 1.1	56.8 ± 1.0	58.7 ± 1.1	56.9 ± 1.2	57.9 ± 1.2
-GloVe	43.2 ± 1.8	44.3 ± 1.2	56.6 ± 1.0	57.1 ± 0.9	56.9 ± 2.7	58.3 ± 2.2	56.4 ± 1.7	56.1 ± 2.2

Table 3: Ablation results without decoding constraints, mean and standard deviation of 5 runs.

Ablation	SEMBLEU		EL-SMATCH					
			F1		Precision		Recall	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Full	47.3 ± 0.6	46.2 ± 0.3	56.3 ± 0.7	57.5 ± 0.8	60.2 ± 0.5	61.5 ± 1.2	52.9 ± 0.9	54.1 ± 1.5
$\Delta\bar{x}$		-1.2		-2.3		+0.8		-4.9
-RoBERTa	46.6 ± 1.3	46.9 ± 0.6	56.1 ± 0.6	57.8 ± 0.4	60.0 ± 0.7	60.5 ± 0.9	52.6 ± 0.6	55.3 ± 0.5
-CharCNN	45.8 ± 2.3	45.5 ± 2.5	56.1 ± 1.4	56.9 ± 1.1	59.3 ± 2.4	59.6 ± 1.8	53.3 ± 1.1	54.5 ± 1.5
-$e_f(C)$ Feats	45.9 ± 1.5	45.6 ± 0.9	56.5 ± 0.6	57.0 ± 0.5	62.0 ± 0.8	61.4 ± 0.6	52.0 ± 1.1	53.3 ± 0.5
-POS	44.1 ± 2.0	44.5 ± 0.9	55.3 ± 0.2	56.6 ± 0.7	58.5 ± 2.2	60.4 ± 0.8	52.6 ± 2.3	53.2 ± 1.4
-GloVe	46.1 ± 1.1	45.4 ± 1.4	55.9 ± 0.9	57.0 ± 0.6	59.5 ± 1.5	60.3 ± 0.8	52.7 ± 1.0	54.0 ± 0.7

Table 4: Ablation results with the lexicon constraint, mean and standard deviation of 5 runs. $\Delta\bar{x}$ is the difference in the mean score between the test set results of the model with the lexicon constraint and without, i.e. Table 3. We only list this for the full model, but the pattern of higher precision but lower scores on other metrics generally holds for the other variants as well.

Ablation	SEMBLEU		EL-SMATCH					
			F1		Precision		Recall	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Full	38.3 ± 2.3	40.0 ± 1.4	54.2 ± 1.2	55.8 ± 1.2	57.6 ± 1.0	59.1 ± 1.2	51.1 ± 1.5	52.8 ± 1.4
$\Delta\bar{x}$		-7.4		-4.0		-1.6		-6.2

Table 5: Ablation results with the type composition constraint, mean and standard deviation of 5 runs. $\Delta\bar{x}$ is the difference in the mean score between the test set results of the model with the type constraint and without, i.e. Table 3. We only ran the full model for this test because this constraint takes much longer to run.

Model	Fragments/Sentence	
	α	τ
Full	1.4	2.9
-CharCNN	1.1	3.5
-$e_f(C)$ Feats	1.4	3.9
-POS	1.5	3.2
$\bar{x}$	1.4	3.4

Table 6: Fragments per sentence on the test set decoding results for a subset of the ablated lexicon-constrained models (Table 4). α is the original model and τ is with the type composition constraint.

(mod-n man.n) (mod-n man.n iron.n) mod-n mod-n) for the text span "device is attached firmly to the ceiling".

This isn't to say that the GS parser always performs worse than our parser. When it comes to words that are elided ({*you*}.*pro* in example 4), nodes generated from multiple words (*had_better.aux-s* in example 3), or logical symbols unassociated with a particular word (*multi-sent* in example 6), the GS parser consistently performs better than our parser. Our parser has no special mechanism for these handling these cases and prefers to avoid generating node labels without an anchoring word.

A common mistake by our parser seems to be nested reifiers, which is not possible in the EL type system (e.g. *(to (ka come.v))* in example 5 and *(to (ka (show.v ..)))* in example 6). Other common mistakes that could be fixed by type coherence enforcement is mistakenly shifting a term into a modifier (e.g. *(adv-a (to ...))* in example 6). In the EL type system only predicates can be shifted into modifiers.

GloVe.840B.300d embeddings	
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	0.001
max_grad_norm	5.0
dropout	0.33
num_epochs	25
Beam size	
without type composition filtering	3
with type composition filtering	10
Vocabulary	
word_encoder_vocab_ size	9200
symbol_encoder_vocab_ size	7300
Batch size	32

Table 7: Default model parameters.

1. *"Speech is silver but silence is golden."*

   Gold: (((k speech.n) ((pres be.v) silver.a)) but.cc ((k silence.n) ((pres be.v) golden.a)))

   Ours: (((k speech.n) ((pres be.v) silver.n)) (k silence.n) ((pres be.v) golden.a))

   GS: (((k (plur *s)) ((pres be.v) (= (k iron.n)))) but.cc ((k iron.n) ((pres be.v) =)))

2. *"You neglected to tell me to buy bread."*

   Gold: (you.pro ((past neglect.v) (to (tell.v me.pro (to (buy.v (k bread.n)))))))

   Ours: (you.pro ((past neglect.v) (adv-e (to (tell.v me.pro (to (buy.v (k bread.n)))))))

   GS: (you.pro ((past fail.v) (to (tell.v me.pro {ref}.pro))))

3. *"You'd better knuckle down to work."*

   Gold: (you.pro ((pres had_better.aux-s) (knuckle.v down.adv-a (to work.v))))

   Ours: (you.pro ((pres would.aux-s) (knuckle.v down.a (adv-a (to.p work.v)))))

   GS: (you.pro ((pres had_better.aux-s) (go.v (to.p-arg (k work.n)) (adv-a (to.p (ka work.v))))))

4. *"Make sure that the device is attached firmly to the ceiling."*

   Gold: ({you}.pro ((pres make.v) sure.a
   (that ((the.d device.n)
   ((pres (pasv attach.v)) firmly.adv-a (to.p-arg (the.d ceiling.n)))))))

   Ours: (((pres make.v) sure.a that.pro (tht
   ((the.d device.n) ((pres be.v) (k (n+preds attach.v (to.p-arg ceiling.n))))))))

   GS: (({you}.pro ((pres make.v) (sure.a
   (that (the.d (mod-n (mod-n man.n) (mod-n man.n iron.n) mod-n mod-n)))))) !)

5. *"Can't I persuade you to come?"*

   Gold: (((pres can.aux-v) not i.pro (persuade.v you.pro (to come.v)) ?)

   Ours: (sub ((pres can.aux-v) not i.pro (persuade.v you.pro (to (ka come.v)) ?)))

   GS: (((pres can.aux-v) not i.pro (come.v (to come.v) you.pro)) ?)

6. *"Look carefully. I'm going to show you how it's done."*

   Gold: (multi-sent (({you}.pro ((pres look.v) carefully.adv-a)) !)
   (i.pro ((pres be-going-to.aux-v)
   (show.v you.pro (ans-to (sub how.pq (it.pro ((pres (pasv do.v)) *h))))))))

   Ours: (((pres look.v) carefully.adv-a)
   (tht (i.pro ((pres be.v) (go.v
   (adv-a (to (ka (show.v you.pro (sub how.pq (it.pro ((pres be.v) do.n))))))))))))

   GS: (multi-sent (({you}.pro ((pres be.v) you.pro fine.a)) !)
   (i.pro ((pres be.aux-v) (go.v (to (do.v you.pro *h))))))

Figure 6: Several parse examples comparing behavior of our parser with the stronger baseline, the GS parser. For each example, the top is the gold parse, the center is our parser, and the bottom is the GS (Cai and Lam, 2020) parser. Errors are written in red. If something from the gold parse is omitted, a red highlighted block marks the location.

"Politeness, you simpleton!" retorted [MASK]: Masked prediction of literary characters

Eric Holgate
The University of Texas at Austin
Department of Linguistics
holgate@utexas.edu

Katrin Erk
The University of Texas at Austin
Department of Linguistics
katrin.erk@utexas.edu

Abstract

What is the best way to learn embeddings for entities, and what can be learned from them? We consider this question for the case of literary characters. We address the highly challenging task of guessing, from a sentence in the novel, which character is being talked about, and we probe the embeddings to see what information they encode about their literary characters. We find that when continuously trained, entity embeddings do well at the masked entity prediction task, and that they encode considerable information about the traits and characteristics of the entities.

1 Introduction

Neural language models have led to huge improvements across many tasks in the last few years (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019).[1] They compute embeddings for words and word pieces. But when we describe the semantics of a sentence, we talk about entities and events and their relations, not words. And it is to be expected that more complex reasoning tasks would eventually require representations at the semantic level rather than the word level. Entities differ from words in that they are persistent, localized, and variable (within a given range). So, would it be beneficial to compute embeddings of entities in addition to embeddings of words for downstream inference? And how should entity embeddings be computed?

There has been a steady rise in work on entity representations and how they can be combined with language models, for example Li et al. (2016); Bosselut et al. (2018); Rashkin et al. (2018); Louis and Sutton (2018).In this paper, we add to the growing literature on neural representations of entities by considering a particularly challenging case: the representations of entities in very long texts, in particular in novels. Intriguingly, Bruera (2019) recently tested whether literary characters, when represented through distributional vectors trained on the first half of a novel, can be recognized in the second half, and found the task to be near impossible. We take up that same task, but train character embeddings in a masked character prediction task. We ask the following questions. (a) Is it possible to use literary character embeddings to do masked character prediction, that is, to guess from a sentence in a novel which character it mentions? (b) If this task is doable, is it doable only locally, or can we train on the first third of a novel and then guess characters towards the end of the novel? (c) What do the resulting embeddings tell us about the literary characters when we probe them? (d) Can the embeddings identify a literary character from a short description of their personality?

We find that when continuously trained, entity embeddings do well at the masked entity prediction task, and that they encode considerable information about the traits and characteristics of the entities. Modeling semantics for natural language understanding is about modeling entities and events, not words. So we view this work as an initial step in the direction of entity modeling over time.

2 Related Work

Entities have been increasingly common subjects within NLP research. There has been recent work aimed at inducing both characteristics of entities, such as personalities, physical and mental states, and character traits, as well as distributed entity representations, similar to lexical embeddings.

[1] The [MASK] in the title is actually La Carconte, from the *Count of Monte Cristo* by Alexandre Dumas.

Proceedings of the 14th International Conference on Computational Semantics, pages 202–211
June 17–18, 2021.

2.1 Modeling Personalities and Characteristics

Psychologists have studied the relationship between *personality traits* and *human behavior*. Within NLP, there have been recent attempts to model this link computationally.

Bamman et al. (2013) explored entity modeling by using Bayesian methods to induce moderately fine-grained character archetypes/stereotypes from film plot summaries. The authors utilized dependency relations to identify, for each entity, the verbs for which they were the agent, the verbs for which they were the patient, and any modifiers attributed to them. Bamman et al. successfully induced clusters that could be manually aligned with tropes like *the jerk jock*, *the nerdy klutz*, *the villain*, etc.

Plank and Hovy (2015) recently appealed to psychological personality dimensions in relation to linguistic behavior. They constructed a dataset by crawling twitter for mentions of any of the 16 Myers-Briggs Type Indicators comprising four personality dimensions (MBTI; Myers and Myers 2010), labeling tweets with author gender identity. Plank and Hovy then train logistic regression models to predict each of the four dimensions from user tweet data using tweet context features and other features that are traditional for Twitter data (e.g., counts of tweets, followers, favorites, etc.). In all four dimensions, logistic regression classifiers outperform majority baselines, supporting the notion that linguistic behavior correlates with MBTI designations.

Flekova and Gurevych (2015) similarly explored personality traits, though they utilized the Five-Factor Model of personality instead of MBTIs (John et al., 1999). Here, authors collected extraversion/intraversion ratings for approximately 300 literary characters, and explore three sources of signal to predict the extraversion scores. The first system aligns most closely with Plank and Hovy's work as it considers only character speech (both style and content). Flekova and Gurevych go slightly farther, however, as they also show that character actions and behaviors as well as the descriptions of characters given in narration carry useful signal for extraversion prediction.

Rashkin et al. (2018) modeled the mental state of characters in short stories, including motivations for behaviors and emotional reactions to events. The authors noted a substantial increase in performance in mental state classification when entity-specific contextual information was presented to the classifier, suggesting that entity-specific context may be useful to a wide array of downstream tasks.

Louis and Sutton (2018) further explored the relation between character properties and actions taken in online role-playing game data. In *Dungeons and Dragons*, a giant is more likely than a fairy to wield a giant axe, but a fairy is more likely to be agile or cast spells. Louis and Sutton show that computational models can capture this interaction by using character description information in conjunction with action descriptions to train action and character language models. When a formal representation of a given character is included, performance improves.

Bosselut et al. (2018) demonstrated dynamically tracked cooking ingredients, identifying which ingredient entity was selected in any given recipe step, and recognizing what changes in state they underwent as a result of the action described in the step. For example, these dynamic entity representations enabled the model to determine that an ingredient was clean after having been washed.

2.2 Entity Representations and Entity Libraries

Recently, major work in NLP has begun to explicitly model entities for use in downstream tasks. While still new (and limited in scope), much of this work has relied upon the notion of an Entity Library, a vocabulary of individuals which utilizes consecutive mentions to construct distributed vector representations, though methods of learning these representations have varied.

Entity representations have been shown to improve the quality of generated text. In Ji et al. (2017), researchers build a generative language model (an RNN) which has access to an entity library which contains continuous, dynamic representations of each entity mentioned in the text. The result is that the library explicitly groups coreferential mentions, and each generated mention affects the subsequently generated text.

Tracking entity information has also been shown to be useful for increasing the consistency of responses in dialogue agents (Li et al., 2016). Researchers introduce a conversation model which maintains a persona, defined as the character that the artificial agent performs during conversational interactions. The persona maintains elements of identity such as background facts, linguistic behav-

ior (dialect), and interaction style (personality) in continuously updated distributed representations. The model maintains the capability for the persona to be adaptive, as the agent may need to present different characteristics to different interlocutors as interactions take place, but reduces the likelihood of the model providing contradictory information (i.e., maintaining these distributed representations prevents the model from claiming to live in both Los Angeles, Madrid, and England in consecutive queries). Crucially, this desirable change is achieved without the need for a structured ontology of properties, but instead through persona embeddings that are learned jointly with word representations.

Fevry et al. (2020) demonstrates that entity representations trained only from text can capture more declarative knowledge about those entities than a similarly sized BERT. Researchers showed that these representations are useful for a variety of downstream tasks, including open domain question answering, relation extraction, entity typing, and generalized knowledge tasks.

Yamada et al. (2020) explore another entity masking task in the context of transformer pretraining. They train a large transformer on both masked words and masked entities in Wikipedia text. Here, however, each entity-in-context exists as its own token, rather than a representation that is aggregated over a sequence of mentions. Yamada et al. test on entity typing, relation classification, and named entity recognition.

Finally, Bruera (2019) introduces the data that we will use to build our model (described in detail below), and compares the ability to construct computational embeddings for proper names with that of common nouns. Researchers trained a distributional semantics model to create and store two different representations for literary characters in novels, each from a separate section of text from the novel. The model is then asked to match the characters' representations from one portion of text to the representations computed from the other portion of text, which the authors term the Doppelgänger Task. Importantly, their results showed that the ability to match these representations is much reduced in the case of proper names when compared to common nouns. This insight serves as a major motivation for the current work, where we follow the hypothesis that entities can be represented in a distributional fashion after all, though not with the same training as with common nouns.[2] We assume that entity representations must be persistent, continuously available, and dynamic.

Dataset	Min	Max	Mean
OriginalNovels	6,770	568,531	118,184.7
WikiNovels	484	13,261	5,104.6

Table 1: Document length statistics for each Novel Aficionados dataset.

3 Data

In the current paper, we present a model that is able to construct entity representations for characters in classic literary novels. Novels are a compelling environment for this exploration as they feature a relatively small number of entities that appear frequently over a long document. To this end, we turn to the Novel Aficionados dataset introduced by Bruera (2019).

The dataset comprises 62 pieces of classic literature, represented as both their original texts (deemed the *OriginalNovels dataset*; these texts are distributed by Project Gutenberg, which maintains a repository of free eBooks of works no longer protected by copyright), and their English Wikipedia summaries (the *WikiNovels dataset*). In order to have sufficient description of as many characters as possible, we only utilize the corpus of original novels in training our representations, as this corpus yields significantly more mentions per character. We utilize the Wikipedia summaries as a test set to determine how well our entity representations work outside the domain of the novels themselves.

The novels are distributed within the dataset in both their original form and having been preprocessed with BookNLP (Bamman et al., 2014). BookNLP is a natural language processing pipeline that extends from Stanford CoreNLP (Manning et al., 2014) and is specifically aimed at scaling to books and other long documents. BookNLP includes part of speech tagging, dependency parsing, NER, and supersense tagging. Most critical to our application, BookNLP provides quotation speaker identification, pronominal coreference resolution,[3]

[2] While our work is inspired by Bruera (2019) and conducted on the same data, we introduce a different task that is not directly comparable to the Doppelgänger Task.

[3] Unfortunately, the texts are not distributed with more general coreference resolution (outside of character aliases and pronominal resolution). This means we are unable to include nominal expressions as character mentions to be considered

and character name clustering. This means that, in addition to standard anaphoric coreference resolution, BookNLP can identify different proper names as character aliases (i.e., Jane Fairfax, a character from Jane Austen's *Emma*, is referenced throughout the text not only by her full name, but also *Jane*, *Miss Jane Fairfax*, and *Miss Fairfax*; BookNLP is able to recognize this and map all of these aliases to a single, unique character ID). Concerning the quality of the coreference resolution, Bamman et al. report average accuracy of 82.7% in a 10-fold cross-validation experiment on predicting the nearest antecedent for a pronominal anaphor. While the accuracy of character clustering was not evaluated, manual inspection of the data revealed it to be very reliable.

4 Modeling

Our hypothesis is that it is possible to represent characters in a novel through an embedding in such a way that it is possible for a model to recognize who is who, or, as we call the task here, *Is this Me?*. Bruera (2019) found that with an approach that treated characters like common nouns, the related *Doppelgänger Task* was not feasible.[4] We hypothesize that if embeddings are learned to best facilitate *Is this Me?* prediction, the task will be feasible. We further hypothesize that the resulting embeddings can be found to contain information about the characters. In a way, our approach is similar to recent contextualized language models like BERT (Devlin et al., 2019) in that we, too, train on a masked prediction task, and we, too, hope to find the resulting embeddings to be useful beyond the prediction task itself.

4.1 A model for the "Is this Me?" task

Our model keeps track of characters as they appear in a novel, and trains an embedding for each character through the *Is this Me?* task, a masked prediction task: Given a sentence of the novel with a masked character mention, and given the current embedding for character c, a classifier decides whether this is a mention of c or not. This is a binary task. The embedding for each character is updated incrementally as the novel is read, and as such, the entity embeddings are learned directly from the data. The classifier weights are updated alongside the character embeddings.

Because the classifier weights are learned as the model reads the novels, we read all novels in parallel. The classifier is trained on a binary masked prediction task, where negative examples are drawn from the same novel. (That is, a negative example for Emma in the novel *Emma* might be Harriet, but it would never be Heathcliff.) A sketch of the model is shown in Figure 1.

Entity Library. The entity library, shown in blue in Figure 1 is a collection of embeddings of literary characters, each represented by a 300 dimensional embedding learned incrementally throughout the novel. Entity embeddings are randomly initialized and passed through a projection layer (green in the figure) before being received by the classifier.

Contextual Sentence and Target Mention Representations. We utilize the base, uncased distribution of BERT to compute *contextualized sentence representations* of each target sentence, shown in orange in Figure 1. Contextualized sentence representations are truncated to a maximum of 150 subword tokens.[5] We do not fine tune BERT on our data. All character mentions in a sentence are masked. The input to the classifier is a *target representation* of one of the masked entity mentions, using *mix* representations introduced in (Tenney et al., 2019). The *target mention representation* is computed directly from the contextualized sentence representations obtained from BERT and is a scalar mix of the layer activations using learned scalars.

***Is this Me?* Binary Classifier.** The classifier for the binary *Is this Me?* task takes as input an entity embedding, transformed through the projection layer, along with a target mention embedding from BERT as described above. The classifier consists of a single, ReLU activation layer. We keep the classifier this simple intentionally, as, to be successful, we want the entity representations to do the heavy lifting.

4.2 Model details

Training Data. We restrict our modeling to characters that appear at least 10 times to ensure that

by the model.

[4] Although related, the *Is this Me?* and *Doppelgänger* tasks are truly different in nature. As such, we cannot compare results on the *Is this Me?* task to results on the *Doppelgänger Task* directly.

[5] This limit was determined by inspecting the length of each sentence in the corpus in subword tokens and permits nearly all sentences to remain untruncated.

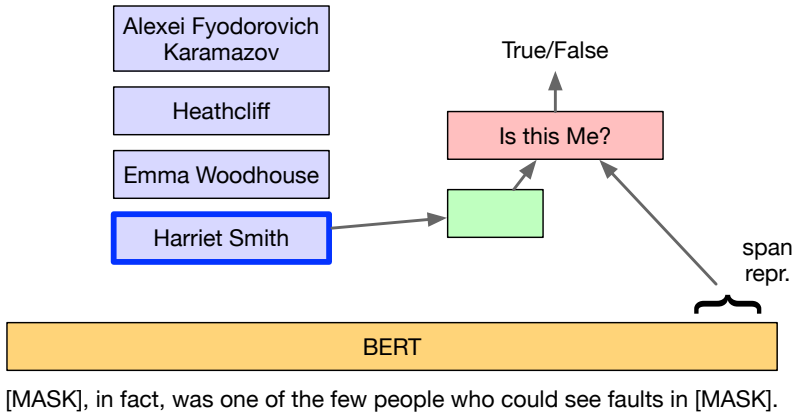


Figure 1: Sketch of the model. The sentence is from Jane Austen's "Emma". All characters are masked. In this case, the first character is Knightley, the second – the target – is Emma. Blue: entity library. Red: *Is this Me?* classifier.

there is enough information to train a representation.

As our intent is to induce entity representations for each character, we must mask each character mention. For each mention of any character predicted by BookNLP in each sentence in a novel, we replace the mention with a single [MASK] token in order to obscure the character's identity from the model. Multiword mentions are reduced to a single [MASK] token in order to prevent the model from being able to detect signal from mention length. Masking is applied to any mention, even for characters that appear fewer than 10 times.

For each sentence in a novel that contains at least one character mention, we produce at least two examples for the model: one positive example from the gold data, and one hallucinated example by randomly selecting a confound character from the same novel. If a character is mentioned more than one time in the same sentence, one mention is randomly selected to be the target mention for that character in that sentence. If a sentence talks about more than one character, a single positive example is generated for each character. Consider this sentence from Jane Austen's *Emma*:

> Whenever $[\text{MASK}]_{\text{James}}$ goes over to see $[\text{MASK}]_{\text{James'}}$ daughter, you know, $[\text{MASK}]_{\text{Miss Taylor}}$ will be hearing of us.

We first have to decide whether to first generate examples for James or for Miss Taylor. We pick one of the two at random, let us assume it is James. We next randomly select one of the two mentions of James to be the target mention. Let us say we pick the first. The input to the model for the *positive* example is then the Tenney et al. (2019) mix representation of the target mention concatenated with the current entity representation of James. We then construct a *negative* example by randomly selecting a character other than James to serve as a confound, following standard practice. If, for example, we were to sample Isabella (Emma's sister), the input to the model for the negative example from this mention would be the exact same mix representation of the target mention concatenated with the current entity embedding of the confound character, Isabella. With positive and negative examples constructed for James's mention, we then turn to the remaining character, Miss Taylor, and construct a positive and negative example for her mention.

Note that restricting the possible set of confounds for a given character to characters in the same novel, we have created a more difficult negative example than if we were to sample across all novels. For example, telling the difference between Elizabeth Bennet and Jane Bennet (both from Austen's *Pride and Prejudice*) is significantly more difficult than telling the difference between Elizabeth Bennet and the Cowardly Lion (from Baum's *The Wonderful Wizard of Oz*).

Training. All learned weights (the entity embeddings themselves, those in the projection layer, the scalars guiding the target mention representation, and those in the classifier) are updated with respect to cross entropy loss, optimized with Adam (Kingma and Ba, 2015) at a learning rate of 2e-05.

5 Experiments

5.1 Is this Me?

We first address our questions (a) and (b) from above: Is it possible to predict a masked mention of a literary character from an entity embedding, either within the same novel or in a summary of the same novel? And does performance degrade if we "skip ahead", using a character embedding trained on the beginning of a novel to predict a mention near the end?

5.1.1 Continuous Training

We begin by allowing the model to train entity representations (and all other learned weights) continuously throughout each novel. This means that we treat each example as a test example, and only allow the model to update based on its performance on a given example after its prediction has been made, as in a standard learning curve. As such, although the model is updated after every example, our performance statistics are computed over its prediction made *before* the update operation (meaning there is no performance computed over already-seen examples). As Table 2 shows, the model does well at this task, with overall accuracy across all characters and all novels of 74.37%. Accuracy was consistent across positive and negative examples. Most learning happened quickly within the first 50,000 examples, though accuracy did continue to increase through the entire run (Figure 2).

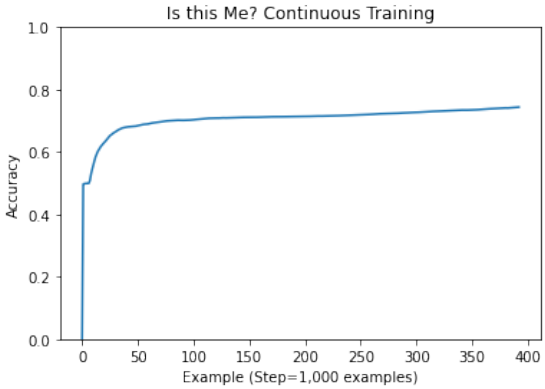


Figure 2: Is this Me? Continuous Training learning curve.

As should be expected, overall accuracy at the book level in this task is subject to frequency effects. Book-level accuracy exhibits strong positive correlation with the total number of examples per novel ($r = 0.584$; $p \ll 0.01$; Figure 3, left). Interestingly, however, book-level accuracy also increases with the number of characters modeled per book ($r = 0.500$; $p \ll 0.01$; Figure 3, right). To see whether the model is affected by language differences between older and more recent books, we used linear regression to predict book-level accuracy from novel publication date, finding very low correlation ($R^2 = 0.008$; $p = 0.663$).

	Examples	Correct	Accuracy
Positive	196,154	149,505	76.22%
Negative	196,154	142,258	72.52%
Total	392,308	291,763	74.37%

Table 2: *Is this Me?* accuracy for continuously trained entity representations.

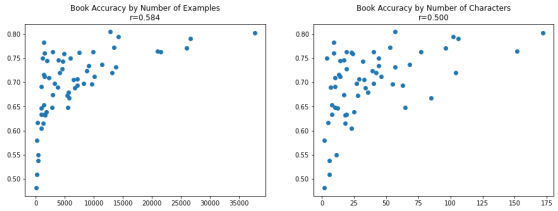


Figure 3: Is this Me? Continuous Training - Book-Level Accuracy. Accuracy within book (y-axis) is plotted against the number of examples for that book (x-axis).

At the character level, frequency effects were not nearly as strong, except in cases where characters were mentioned very frequently (defined here as characters with over 300 mentions). Across all characters, testing showed moderate positive correlation with mention frequency ($r = 0.174$; $p \ll 0.01$; Figure 4, left). Within frequently appearing characters, correlation with mention frequency was much higher ($r = 0.633$; $p \ll 0.01$; Figure 4, right).

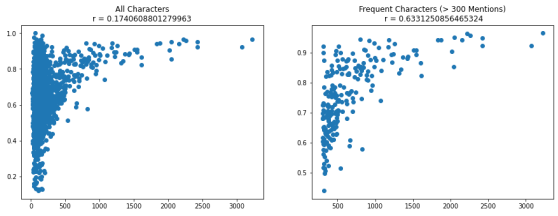


Figure 4: Is this Me? Continuous Training - Character-Level Accuracy. Accuracy within character (y-axis) is plotted against the number of examples for that character (x-axis).

5.1.2 Applicability to Novel Summaries

We also explored the extent to which the entity representations after having been trained on the full novel, could identify the same entities in short summaries of the same novel. To that end, we used the the WikiNovel summaries distributed with the Novel Aficionados dataset. The summaries show a strong domain shift compared to the novels. While they frequently do contain proper, if succinct, descriptions of the novel's plot and the involvement of major characters, they also exhibit significantly different patterns of language. Wikipedia entries do not just summarize the novels, they also frequently include metadiscursive language, as in this sentence from the WikiNovels summary of Austen's *Emma*:

> This point of view appears both as something perceived by *[emma_woodhouse]* an external perspective on events and characters that the reader encounters as and when *[emma_woodhouse]* recognises it and as an independent discourse appearing in the text alongside the discourse of the narrator and characters.

Because of this shift in domain, we see vastly reduced performance in character prediction and a heavy bias towards claiming the target mention is not a given character when using the model trained on the sentences from the novel. This is shown in Table 3. We evaluated the model in two settings. In the Pos. Only setting, all data points were positives, such that the model would have perfect accuracy by always saying yes. In the Pos. & Neg. setting, we use the same negative example generation technique as used in the model's training. While the model performs slightly better than chance when negative examples are included, it remains clear that future work should explore ways to generalize the entity representations such that they may be more informative across domain boundaries.

Example Types	Num. Examples	Accuracy
Pos. Only	7,736	36.12%
Pos. & Neg.	15,573	56.13%

Table 3: *Is this Me?* accuracy for continuously trained entity representations on WikiNovel summaries.

5.1.3 Non-Continuous Training

In §2 we noted that identifying masked character mentions is a not trivial due to the nature of narratives themselves. Literary plots are often constructed to force profound change in the behaviors, beliefs, and characteristics of central characters. This may be among the reasons that Bruera (2019) reported such difficulty with the Doppelgänger task. To see if distance in the novel affects our representations, we experimented with "skipping ahead" in the novel in order to determine the impact on performance when entities are not continuously updated.

Inspired by traditional character arcs, we split each novel into three sections of equal length (determined by number of sentences). The underlying assumption is that, due to the structure of the narrative, each character (especially main characters) will undergo some form of growth or change in between each novel section, suggesting that the learned entity representations should never be static in order to encode the results of that growth. We allowed a new *Is this Me?* classifier to learn representations for all literary entities using only the first third of the novels as training data, then froze the entity embeddings, and evaluated classifier performance against the middle and final thirds independently. We hypothesized that the model would exhibit a gradual decrease in performance as it moved further away from the point in time at which the entity representations were fixed, with the performance on the middle third better than performance toward the ends of the novels. Instead, we found a fairly rapid decline in performance (Table 4). Performance stays above chance, however, suggesting there is a kernel of each representation that is informative regardless. While this experiment does not explicitly demonstrate character development/change, the sharp decrease in performance when entity representations are fixed implicitly supports the claim that such change is present. Capturing that development directly, however, is another very difficult task and well-worthy of being the subject of future work.

Trained On	Beginning	Middle	End
Beginning	68.50%	55.70%	57.15%
Beg. & Mid.	68.50%	63.24%	57.45%

Table 4: *Is this Me?* accuracy on novels split into thirds.

5.2 Probing Entity Embeddings

We have found that, at least when entity embeddings are continuously trained, they can be used to predict a masked mention in a novel with reasonable accuracy. But are the resulting embeddings useful beyond the masked entity prediction task? To find this out, we turn to our questions (c) and (d) from above, and see if we can predict character gender from entity representation, and whether the identity of a character can be predicted from a description.

5.2.1 Predicting Gender from Literary Character Representation

We used a simple logistic regression model to probe the extent to which gender is encoded in the entity representations obtained from the continuous training in §5.1.1. As we have no gold annotation of literary character gender, we utilize the BookNLP preprocessing to look for gendered pronouns (*she/he*) for each character as a form of distant supervision. Manual inspection shows this heuristic to be very reliable after omitting characters for which no pronominal coreference link is available and characters who exhibit coreference chains featuring both gendered pronouns. This left a total of 2,195 characters (1,533 male, 662 female) to be considered for this experiment.

We learn a single weight for each embedding dimension for a total of 300 weights. In each case, we trained the classifiers on 80% of the characters across all novels (1,756 characters), leaving a test set of 439 characters. Each model was run four times, and we present the mean performance statistics in Table 5. Results were favorable across all runs, suggesting the learned character representations do encapsulate some knowledge of literary character gender.

μ **Acc**	μ **MSE**	μ **F1**
60.15%	0.3984	0.7208

Table 5: Model performance on predicting character gender from entity embeddings: Accuracy, mean squared error, and F1.

5.2.2 Character Descriptions

While the WikiNovels corpus is noisy and cluttered with metadiscursive literary commentary, as noted in §5.1.2, certain Wikipedia novel summaries do contain detailed descriptions of major characters. To better evaluate the ability of our learned entity representations to generalize outside of the domain of the novels on which they where trained, we manually extracted a subset of sentences which more readily pertained to our research question.

We isolated five novels which featured clean character descriptions within their summaries: Jane Austen's *Emma*, Charles Dickens's *A Tale of Two Cities* and *Great Expectations*, Fyodor Dostoevsky's *The Brothers Karamazov*, and Charlotte Brontë's *Jane Eyre*. From the character descriptions within these summaries we generated a total of 605 *Is this Me?*-style examples (positive and negative).[6] The pre-trained classifier exhibited performance above chance (61.63% accuracy), and a surprising ability to handle challenging out of domain sentences. While the model successfully predicted a high level description of Emma Woodhouse (Table 6; Row 1), it struggled with a similar description of Estella Havisham (Row 2). The model was also able to identify a character based on the description of a pivotal plot point (Row 3), but unsurprisingly struggled with more critical descriptions (Row 4).

6 Conclusion

In the ideal case, an entity embedding would constitute a compact representation of a person, their character traits and life story, and would allow for inferences about that person, including story arcs in which that person is likely to occur. What is the best way to learn embeddings for entities, and what can be learned from them? We have considered this question for the case of literary characters. We have trained entity embeddings through a masked prediction task, reading a collection of novels from begininng to end. We found that when trained continuously, the entity embeddings did well at the *Is this Me?* task: Given a target entity embedding, and given a sentence of the novel with a masked entity mention, is this a mention of the target entity? The *Is this Me?* task becomes much harder when we "skip ahead", training only on the first third of a novel and then evaluating on the middle and end. The task also becomes much harder when applied to Wikipedia summaries of novels, which show a marked domain difference from the novels themselves. Probing the entity embeddings that result from the masked prediction task, we find that they encode a good amount of information about the

[6] This set of examples may be found at http://www.katrinerk.com/home/software-and-data.

Novel	Target	Candidate	Result	Sentence
Emma	Emma	Emma	+	[MASK] the protagonist of the story is beautiful high spirited intelligent and lightly spoiled young woman from the landed gentry.
Great Expectations	Estella	Estella	-	She hates all men and plots to wreak twisted revenge by teaching [MASK] to torment and spurn men, including Pip who loves her.
A Tale of Two Cities	Miss Pross	Miss Pross	+	[MASK] permanently loses her hearing when the fatal pistol shot goes off during her climactic fight with Madame Defarge.
A Tale of Two Cities	Lucy Manette	Lucy Manette	-	She is the golden thread after whom book the second is named so called because [MASK] holds her father and her family lives together and because of her blond hair like her mother.

Table 6: Examples of the *Is this Me?* continuously trained classifier's performance on out-of-domain masked mentions found within the WikiNovels corpus. Non-target mentions have been de-masked for better readability.

entities. The gender of the literary character can in many cases be recovered from the embedding, and it is even often possible to identify a person from a Wikipedia description of their characteristic traits.

Looking ahead, the training regime and trained embeddings allow for many further analyses. We would like to probe further into the "skipping ahead" to better understand why it is so difficult. Intuitively, characters that undergo more development across the length of a novel should be more difficult to predict. It is not clear to what extent this is the case with the current model; this needs further study. In addition, we would like to model the change and development of characters more explicitly, for example by representing them as a trajectory over time rather than a single embedding. It would also be beneficial to further explore the ways in which character traits are implicitly present within entity representations learned from the *Is this Me?* task. While we attempted to probe this superficially via the evaluation on out-of-domain Wikipedia data, this data does not offer the annotation that would be necessary to perform a more in-depth analysis

We would also like to extend the model by including additional relevant input. At the moment, we essentially ask the model to bootstrap entity representations from scratch, using only the contextualized sentence representations produced by BERT and the current entity representations as input. Other useful information such as semantic relations (retrievable via dependency parse) may be useful. We may also consider the kind of events and modifiers that a given entity participates in to be able to exploit patterns across character archetypes (similar to Bamman et al. (2014)). We are also looking to extend the model to directly model relations between characters as relations between entity embeddings, to see whether this would help performance and to see to what extent the interpersonal relations of characters would be encoded in their embeddings.

Overall, we find the results presented in the current paper to be promising as a first step towards natural language understanding systems that use neural models of entities over time. As we have outlined here, however, there is still much work to be done.

7 Acknowledgements

This research was supported by the DARPA AIDA program under AFRL grant FA8750-18-2-0017. We acknowledge the Texas Advanced Computing Center for providing grid resources that contributed to these results, and results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation. We would like to thank the anonymous reviewers for their valuable feedback, as well as Jessy Li and Pengxiang Cheng.

References

David Bamman, Brendan OConnor, and Noah A Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 352–361.

David Bamman, Ted Underwood, and Noah Smith. 2014. A Bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *Proceedings of the 6th International Conference for Learning Representations (ICLR)*.

Andrea Bruera. 2019. Modelling the semantic memory of proper names with distributional semantics. Master's thesis, Universita degli Studi di Trento.

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.

Thibault Fevry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951. Association for Computational Linguistics.

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1805–1816.

Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith. 2017. Dynamic entity representations in neural language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1830–1839, Copenhagen, Denmark. Association for Computational Linguistics.

Oliver P John, Sanjay Srivastava, et al. 1999. The Big Five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research*, 2(1999):102–138.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model.

Annie Louis and Charles Sutton. 2018. Deep Dungeons and Dragons: Learning character-action interactions from role-playing game transcripts. 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 708–713.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Isabel Briggs Myers and Peter B Myers. 2010. *Gifts Differing: Understanding Personality Type*. Nicholas Brealey.

Matthew 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.

Barbara Plank and Dirk Hovy. 2015. Personality traits on Twitter -or- how to get 1,500 personality tests in a week. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–98.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299, Melbourne, Australia. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Tuning Deep Active Learning for Semantic Role Labeling

Skatje Myers
University of Colorado at Boulder
skatje.myers@colorado.edu

Martha Palmer
University of Colorado at Boulder
mpalmer@colorado.edu

Abstract

Active learning has been shown to reduce annotation requirements for numerous natural language processing tasks, including semantic role labeling (SRL). SRL involves labeling argument spans for potentially multiple predicates in a sentence, which makes it challenging to aggregate the numerous decisions into a single score for determining new instances to annotate. In this paper, we apply two ways of aggregating scores across multiple predicates in order to choose query sentences with two methods of estimating model certainty: using the neural network's outputs and using dropout-based Bayesian Active Learning by Disagreement. We compare these methods with three passive baselines — random sentence selection, random whole-document selection, and selecting sentences with the most predicates — and analyse the effect these strategies have on the learning curve with respect to reducing the number of annotated sentences and predicates to achieve high performance.

1 Introduction

The ability to identify the semantic elements of a sentence (*who* did *what* to *whom*, *where* and *when*) is crucial for machine understanding of natural language and downstream tasks such as information extraction (MacAvaney et al., 2017) and question-answering systems (Yih et al., 2016). The process of automatically identifying and classifying the predicates in a sentence and the arguments that relate to them is called semantic role labeling (SRL). The current state-of-the-art semantic role labeling systems are based on supervised machine learning and rely on large corpora in order to achieve good performance. Large corpora have been created for languages such as English (Weischedel et al., 2013), but such resources are lacking in most other languages. Additionally, those corpora created may still not translate well to other in-language domains, due to sentence structure or domain-specific vocabulary. Creation of additional annotated corpora requires a significant amount of time and often the hiring of domain experts, causing a bottleneck for developing advanced NLP tools for other languages and domains.

Active learning (AL) focuses on choosing only the most informative and least repetitive instances to have annotated, thereby reducing the total needed annotation to train a supervised model, without sacrificing performance. This is done by iteratively re-training the model and assessing its confidence in its predictions in order to choose additional data for annotation that would have maximal impact on the learning rate.

Traditionally, practitioners use the model's probability distributions for the annotation candidates to quantify how informative a new training instance would be for the model. However, state-of-the-art SRL systems rely on deep learning, whose predictive probabilities are not a reliable metric of uncertainty. In lieu of this, Gal and Ghahramani (2016) found that we can estimate model confidence by calculating the rate of disagreement of multiple Monte Carlo draws from a stochastic model, accomplished by utilising dropout during forward passes. Previous work (Siddhant and Lipton, 2018)(Shen et al., 2017) has combined this finding with Bayesian Active Learning by Disagreement (Houlsby et al., 2011) as a way of selecting informative instances for active learning for SRL and other NLP tasks; hereafter referred to as DO-BALD.

Semantic role labeling for a single sentence is a complicated structural prediction, involving multiple predicates and varying spans. This complexity makes identifying the training examples with maximal impact more challenging. In this work, we compare two ways of aggregating confidence

Proceedings of the 14th International Conference on Computational Semantics, pages 212–221

scores for individual predicates into a unified score to assess the usefulness of selecting a sentence for active learning. We test these strategies with two active learning approaches to calculating certainty for a predicate instance: the model's output probabilities and a granular DO-BALD selection method. Additionally, we compare the benefits of these AL approaches with three baselines: random sentence selection, random document selection, and selecting sentences with the most predicates.

We will discuss the practical workflow of SRL annotation and the way this must be considered in utilising active learning effectively to create new datasets. Although the current standard data selection methodology for SRL corpora, which typically involves selecting entire documents, leaves much room for improvement by even passive strategies, we will show that active learning can provide significant reductions in annotation of both number of sentences and number of predicates. We aim to provide this comparison within the broader context and understanding of SRL annotation in practice.

2 Background

Active learning begins with the selection of a classifier, a small pool of labeled training data (also referred to as a seed set) for the classifier to initially be trained on, and a large amount of unlabeled data. AL is an iterative process where the classifier is trained on the labeled data and then through some query selection strategy, an instance or instances are chosen from the unlabeled data for a human annotator to provide a label for. Typically, they're chosen after the classifier attempts to predict labels for the unlabeled data and provides feedback about what instances may be the most informative. The newly annotated data is then added to the pool of labeled data that will be used to train the classifier on the next iteration. This iteration continues until some stopping criteria are met, such as the classifier's confidences about the remaining unlabeled data exceeding a certain threshold, or simply until funds or time are exhausted.

Proposition Bank (PropBank) (Palmer et al., 2005) is verb-oriented semantic representation. Predicates in text are assigned a *roleset ID* based on the sense of the word, such as play.01 (*to play a game*) or play.02 (*to play a role*). The roleset determines the permissible semantic roles, or arguments, for that predicate. The core arguments are given generalised numbered labels, ARG0 through ARG5. Typically an ARG0 is the agent or experiencer, while ARG1 is typically the patient or theme of the predicate. Additionally, there are modifier arguments to incorporate other semantically relevant information such as location (ARGM-LOC) and direction (ARGM-DIR). The following is an example of the arguments related to the predicate "give" according to the roleset in Table 1:

Roleset id: give.01	
transfer	
Arg0	giver
Arg1	thing given
Arg2	entity given to

Table 1: PropBank roleset for *give.01*.

[$_{\text{ARG0}}$ She] had [$_{\text{Pred}}$ given] [$_{\text{ARG1}}$ the answers] [$_{\text{ARG2}}$ to two low-ability geography classes].

Sentences may contain several predicates and each predicate has its own arguments. Predicates commonly consist of verbs, but also include nominalisations and predicative adjectives.

Many large corpora have been annotated in English, such as Ontonotes (Weischedel et al., 2013). Although Ontonotes has since been retrofitted to unify different parts of speech into the same rolesets based on sense and given expanded nominalisations, light verb constructions, and other multiword expressions (O'Gorman et al., 2018), an earlier version of it was released as the dataset for the CoNLL-2012 shared task. This dataset is still frequently used as an evaluation corpus for experimental SRL techniques. Additionally, there are many domain-specific SRL corpora, such as clinical records (Albright et al., 2013) and the geosciences (Duerr et al., 2016). These domain-specific annotations are necessary because the vocabulary and sentence structure may differ too much for models trained on more general text to perform well.

Much of the text annotated with PropBank annotations was annotated using Jubilee (Choi et al., 2010). The text is set up to be presented to annotators in the order of the predicate's lemma, enabling annotators to concentrate on the differences between rolesets of particular lemmas and providing efficiency through minimising context-switching. With this methodology, annotation time can pri-

marily be reduced by minimising the number of predicates being annotated.

While this setup is typical of large-scale annotation projects, it's less feasible in the context of active learning. If each iteration results in querying annotators for only 100 sentences, there is little benefit to splitting annotation tasks based on lemmas. The more practical approach is to annotate on a sentence-by-sentence basis. In this case, reducing predicates is still beneficial, but since the cognitive burden of reading and understand the sentence must be done anyway, reducing the number of sentences is of high importance.

When new datasets are annotated, typically entire documents are chosen. Annotation projects frequently do several layers of annotation on the same text, which may include NER, syntactic parsing, SRL, coreference resolution, and event coreference. In the case of SRL, this results in numerous sentences with the same topic and vocabulary being used. The random selection of sentences used as a baseline in active learning studies may be an improvement over the selection criteria used in practice since the distribution of it will result in a more diverse dataset. For this reason, it's important when discussing how much annotation reduction an AL technique provides by selecting individual sentences to compare to the learning curve of random selection, rather than the full dataset. Our experiments include a whole-document selection method to provide comparison.

3 Related Work

Active learning has been utilised with success in numerous NLP tasks, such as named entity recognition (Shen et al., 2017), word sense disambiguation (Zhu and Hovy, 2007), and sentiment classification (Li et al., 2013). In recent years, active learning has been applied to SRL. Since probabilities from off-the-shelf NN models may sometimes be inaccessible, Wang et al. (2017) proposed working around this by designing an additional neural model to learn a strategy of selecting queries. Given an SRL model's predictions, this query model classifies instances as requiring human annotation or not. Their approach was a hybrid of active learning and self-training. The self-training is enacted by accepting the SRL model's predicted labels into the training pool for future iterations when the sentence was determined not to require human annotation. This approach requires 31.5% less annotated data to achieve comparable performance as training on the entirety of the CoNLL-2009 dataset.

Koshorek et al. (2019) compared data selection policies while simulating active learning for question-answer driven SRL (QA-SRL). QA-SRL is a form of representing the meaning of a sentence using question-answer pairs. Rather than annotating spans of text with argument names, such as PropBank's ARG0, annotators enumerate a list of questions relating to the actions in a sentence, such as *who* is performing an action and *when* is it happening, along with the corresponding answers from the original text. This representation provides similar coverage to PropBank, but can also represent implicit arguments that aren't directly represented by the syntax.

The process of identifying spans that are arguments of a predicate and the generation of questions based on the arguments were treated as independent tasks. To provide an approximate upper bound on the learning curve, they simulated active learning on the dataset, splitting the unlabeled candidates into K subsets, and selecting the subset that improved the model the most on the evaluation data. Against this oracle policy, they compared the following selection strategies, sampling K random subsets to choose from: selecting a random subset, selecting the subset with the highest average token count among sentences, and selecting the subset that has the maximal average entropy over the model's predictions.

The uncertainty strategy performed worse than random selection for argument span detection, and was not tested for question generation. Selecting the sentences with high token counts tended to improve the F-score for argument span detection by 1-3% given an equal number of training instances (and attaining 60% on the full dataset), while being largely comparable to random selection for question generation.

Active learning for SRL has also been applied in combination with multi-task learning (Ikhwantri et al., 2018), using a subset of PropBank roles along with a new "greet" role. The authors compared single- and multi-task SRL, both with and without active learning. Under multi-task learning the model jointly learns to identify semantic roles as well as to classify tokens as entities such as "Person" or "Location". They introduced a set of semantic roles that accommodate conversational language and annotated a small corpus of Indonesian

chatbot data to provide training and testing data. By selecting sentences using model uncertainty in the single-task context, F-score was improved by less than 1% compared to randomly selecting the data.

Modern SRL systems utilise deep learning, which poses a challenge to assessing the model's certainty in its predictions. The predictive probabilities in the output layer cannot be reliably interpreted as a measure of model certainty. Gal and Ghahramani (2016) proposed using dropout as a Bayesian approximation for model certainty, estimating it using the variation in multiple forward passes.

This dropout principle was tested on numerous NLP tasks by Siddhant and Lipton (2018), including SRL. For their SRL experiments, they used a neural SRL model based on the He et al. (2017) model, with modifications to the decoding method (instead using a CRF decoder) and increasing the dropout rate from 0.2 to 0.25.

In comparison to the baseline of random selection, they tested the classic uncertainty measure of using the output probabilities of the model, normalised for sentence length, with two Bayesian Active Learning by Disagreement methods for selecting additional instances: Monte Carlo Dropout Disagreement (DO-BALD) and Bayes-by-Backprop (BB-BALD). The DO-BALD method applies dropout during multiple predictions of instances in the unlabeled pool and selects instances based on how many of those predictions disagree on the most common label of the entire sequence. This selection strategy is similar to the selection method we propose in this paper, but with several differences. The most significant difference is that the authors treat agreement between predictions as all-or-nothing, rather than allowing partial agreement based on arguments. They also are using a higher number of predictions (100 per sentence as opposed to 5 per predicate) to calculate disagreement between, which may be necessary in this all-or-nothing approach. In contrast, we consider each predicate-argument label sequence independently.

They tested their methods on both the CoNLL-2005 and CoNLL-2012 datasets, which use PropBank annotation. While the Bayesian methods were similar to the standard uncertainty selection method in the case of SRL, these methods resulted in approximately 2-3% increase for F-score compared to random selection when training on the same number of tokens. These results were much more modest than results for other tasks such as NER.

4 Data

We used two independent datasets for our experiments: The English section of Ontonotes (version 5.0) (Weischedel et al., 2013) with the latest frame updates (O'Gorman et al., 2018) and the colon cancer portion of THYME (Albright et al., 2013).

Ontonotes 5.0 consists of 1.5 million words across multiple genres. The majority of this data is sourced from news, but it also includes telephone conversations, text from The Bible, and web data. THYME is comprised of clinical notes and pathology reports of colon and brain cancer patients. For our experiments, we used only the colon cancer portion. The data is split into training, validation, and test subsets.

We simulated active learning on the training subset of each corpus, dividing it into an initial seed set and a set of sentences to select from. The initial seed sets for sentence-based experiments were 200 randomly chosen sentences. For the whole-document baseline, the seed set is comprised either of documents from multiple genres, totalling 200 sentences, in the case of Ontonotes; or a single patient (consisting of two clinical notes and one pathology report, totalling 195 sentences) in the case of the THYME corpus.

In both cases, we utilised validation data to determine early stopping. Due to the excessive computational time required to predict the standard validation sets for these corpora for every epoch for every iteration, as well as the fact that a real-world scenario would be unlikely to have such a disproportionally large validation set to perform active learning, we selected a subset of the validation data for use. In the experiments involving selecting individual sentences, we used the same randomly chosen 250 sentences. In the case of the baselines of choosing random documents, we used validation datasets approximating 250 sentences, comprised of whole documents.

Evaluation was performed on the standard test subset for each respective corpus.

5 Model

We used AllenNLP's (Gardner et al., 2018) implementation of a state-of-the-art BERT-based model (Shi and Lin, 2019). Our training procedure for

this model used 25 epochs or stopped early with a patience of 5. Trained under the same experimental configuration on the full training subsets, this model achieves an F-score of 83.82 and 83.48 on the Ontonotes and THYME datasets respectively.

After training on the initial seed dataset, each iteration of active learning selected batches of 100 sentences re-trained from scratch. In the case of the whole-document baseline, for the creation of each batch, we selected random documents until the number of sentences selected met or exceeded 100.

6 Selection Methods

6.1 Model Output

We used the classic approach of selecting query sentences based on the probability distribution over labels from the model's output. For each predicate in a sentence, we summed the highest probability for each token and then normalised by sentence length. This results in a single confidence score for the label sequence.

6.2 DO-BALD

The model output of neural networks are a poor estimate of confidence, due to their nonlinearity and tendency to overfit and be overconfident in their predictions (Gal and Ghahramani, 2016)(Dong et al., 2018).

Using Monte Carlo dropout as a Bayesian approximation of uncertainty, as proposed by Gal and Ghahramani (2016), we applied a dropout rate of 10% during the prediction stage. We employ the Bayesian Active Learning by Disagreement approach by predicting each candidate sentence multiple times to select sentences based on how often those predictions agree with each other.

The number of predictions used correspondingly increases the time required to select data upon each iteration. Gal and Ghahramani (2016) used between 1000 and 10 forward passes in their experiments and Siddhant and Lipton (2018) used 100 per sentence when applying DO-BALD to SRL. An ideal solution would minimise this variable for efficiency with as little loss as possible in the benefit gained by sampling the distribution. In our experiments, we chose to perform 5 predictions per predicate. Due to sentences containing multiple predicates, this typically results in 10-15 predictions per sentence.

Prediction 1	[$_{\text{ARG0}}$ John Smith] [$_{\text{Pred}}$ bought] [$_{\text{ARG1}}$ apples].
Prediction 2	[$_{\text{ARG0}}$ John] Smith [$_{\text{Pred}}$ bought] [$_{\text{ARG1}}$ apples].
Prediction 3	[$_{\text{ARG0}}$ John Smith] [$_{\text{Pred}}$ bought] [$_{\text{ARG1}}$ apples].
Prediction 4	[$_{\text{ARG0}}$ John Smith] [$_{\text{Pred}}$ bought] [$_{\text{ARG1}}$ apples].
Prediction 5	[$_{\text{ARG0}}$ John] Smith [$_{\text{Pred}}$ bought] [$_{\text{ARG1}}$ apples].

Table 2: An example of varying argument predictions for a predicate, *bought*, by multiple forward-passes with dropout.

From these predictions, agreement was calculated based on entire argument spans. For each predicate in the sentence, we considered the percent of predictions for each argument type that agreed with the most frequent span choice for that type. Referring to the example in Table 2, the most frequently chosen span for ARG0 was "John Smith", although two of the predictions chose only the partial match of "John". In this case, since two out of the five disagree with the most common prediction, the argument ARG0 has a disagreement rate of 0.4. The rate of disagreement was calculated for each argument type present in the set of predictions and then averaged to summarise the consensus for the entire predicate-argument structure.

By examining the forward-pass predictions predicate-by-predicate and argument-by-argument to determine agreement, our approach is more granular than Siddhant and Lipton (2018)'s method of determining disagreement from the mode of the entirety of the sentence's labels. Our strategy allows for partial credit when the predictions are in agreement about particular arguments.

6.3 Combining Predicate Scores

Since sentences often contain multiple predicates, we must aggregate the scores into a single measure in order to rank sentences by their potential informativeness. We propose two such ways of combining the predicate scores, which we applied to both the Output and DO-BALD methods of calculating certainty of a single predicate-argument structure:

- **Average of Predicates (AP)**: The score for all predicate-argument structures in a sentence is averaged. This provides a balance between the predicates in the sentence, but high confidence for one predicate may diminish the value of a more uncertain predicate.
- **Lowest Scoring Predicate (LSP)**: The score for a sentence is the lowest score of all the predicate-argument structures present in the

sentence. This strategy prioritises sentences that contain a predicate that is most likely to have a high impact on learning, although this may allow selecting for sentences that require annotating additional predicates that have already been learned well by the model.

In the case of DO-BALD, a sentence with two predicates will have ten total forward-passes, five for each predicate. In the following example, a sentence contains one predicate that's very common and may likely already occur in the dataset, come.01 (*motion*), and a second predicate that's less common, make_it.14 (*achieve or arrive at*).

[ARG0 The governor] [ARGM-OutputD could] [ARGM-NEG n't] [Pred make it], so the lieutenant governor came instead.

The governor could n't make it, so [ARG1 the lieutenant governor] [Pred came] instead.

A plausible scenario is that the predictions of the arguments for the rarer predicate "make it" will be in higher disagreement compared to the predictions of the arguments for "came". In this case, the LSP method will be more likely to select the sentence than AP, since it will rank this sentence's likely informativeness based only on the disagreement rate of "make it", whereas AP will average between the two disagreement rates.

6.4 Baselines

We include three passive baseline measurements:

- **Random Sentences (RandSent)**: Choose random batches of sentences on each iteration of active learning.
- **Random Documents (RandDoc)**: Choose random batches of entire documents, until the chosen sentence batch size is reached.
- **Most Predicates (MostPred)** Choose batches of sentences, selecting for those with the highest number of predicates present. Identification of predicates was done automatically using AllenNLP.

7 Results

Out results are reported as a learning curve across number of sentences (Figures 1, 3) and predicates (Figures 2, 4) present in the training pool after each iteration. Selected F-scores for the methods are reported according to number of sentences (Table 3) and approximate number of predicates (Table 4) in the training pool at various points.

# sentences	300	600	900	1200	1500
Ontonotes					
RandSent	55.48	64.32	71.00	72.02	74.95
RandDoc	61.26	64.27	70.20	72.31	73.59
MostPred	59.39	**74.60**	**76.13**	**77.55**	77.52
DO-BALD LSP	60.25	73.48	74.80	76.23	**78.13**
DO-BALD AP	**62.26**	63.92	66.28	69.83	67.29
Output LSP	61.91	70.29	71.08	73.27	74.87
Output AP	62.12	58.52	64.52	62.28	68.39
THYME					
RandSent	64.53	72.07	74.23	75.67	76.88
RandDoc	49.32	64.23	67.11	73.62	75.21
MostPred	**66.66**	74.61	**76.37**	**77.49**	78.66
DO-BALD LSP	58.01	**74.66**	75.81	76.91	**79.03**
Output LSP	64.80	72.87	76.24	77.03	78.69

Table 3: F-score for number of sentences for each query selection method: random sentences, random documents, most predicates, DO-BALD (Lowest Scoring Predicate and Average of Predicates), model output (Lowest Scoring Predicate and Average of Predicates). Sentence count is approximate for whole-document selection.

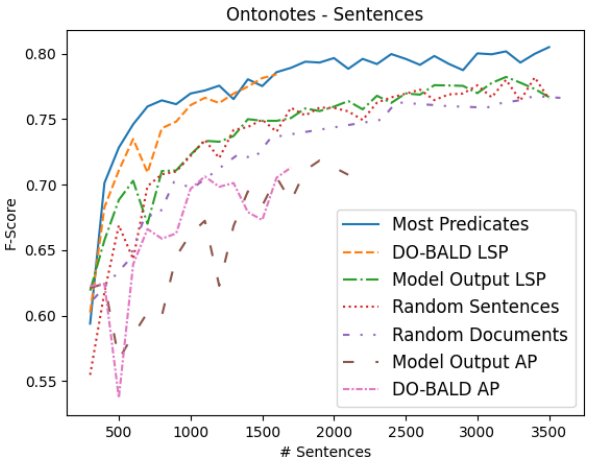


Figure 1: Learning curve of F-score by number of sentences in Ontonotes training data.

7.1 Ontonotes

We can estimate the annotation savings gained by the tested methods by examining the statistics required for each curve to reach a particular F-score. For this purpose, we will choose 78% as a benchmark for a viable SRL model that can produce sufficiently accurate results to feed into downstream NLP applications.

Approx. # predicates	1000	1500	2000	2500	3000
Ontonotes					
RandSent	55.48	66.89	64.32	70.79	72.18
RandDoc	61.26	64.27	67.72	70.20	69.73
MostPred	-	-	59.39	-	-
DO-BALD LSP	60.25	68.27	68.26	**71.08**	**73.47**
DO-BALD AP	**62.43**	66.61	69.67	70.12	70.53
Output LSP	61.91	**68.83**	**70.29**	71.03	72.28
Output AP	56.68	56.00	62.28	68.39	71.09
THYME					
RandSent	66.47	72.06	72.25	76.28	75.67
RandDoc	64.23	67.11	73.32	75.35	**76.23**
MostPred	-	-	70.69	72.57	74.60
DO-BALD LSP	58.01	71.63	**74.66**	**75.82**	75.81
Output LSP	**67.30**	**72.87**	71.57	76.24	76.03

Table 4: F-score for approximate number of predicates for each query selection method: random sentences, random documents, most predicates, DO-BALD (Lowest Scoring Predicate and Average of Predicates), model output (Lowest Scoring Predicate and Average of Predicates). MostPred takes too large of selections to always be within range of these numbers.

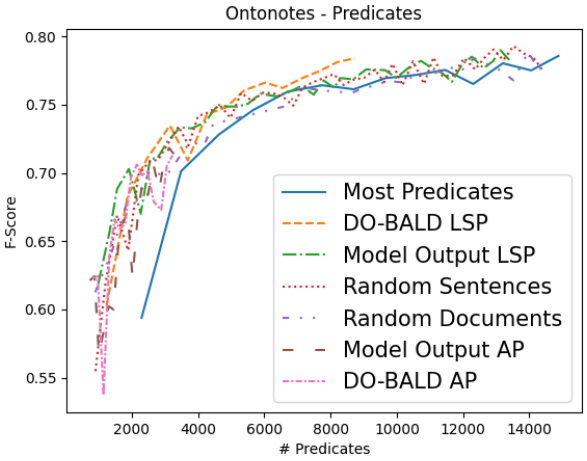


Figure 2: Learning curve of F-score by number of predicates in Ontonotes training data.

The passive selection of random sentences attains this score after 3,000 sentences. The DO-BALD LSP method and MostPred methods achieve this score after 1,400 and 1,200 respectively, providing a **53%-60% reduction in data**. Using the model's output with LSP provided a more slight, but still significant, reduction of 10%. When selecting whole documents, this performance was not achieved until 4,126 sentences were in the training pool. Both of the AP methods, which averaged the predicates in the sentences, performed significantly worse than the baseline.

On the other hand, the reduction in predicate annotation offered by active learning was more modest. The passive strategies of selecting random sentences and documents required 9,333 and 11,598 predicates, respectively. DO-BALD LSP required 7,673 predicates (18% fewer). The MostPred strategy, which offered the best performance on reducing sentences, didn't achieve this until 11,460 predicates, almost comparable to random whole-document selection. Output LSP provided a negligible reduction, with 9,073 predicates.

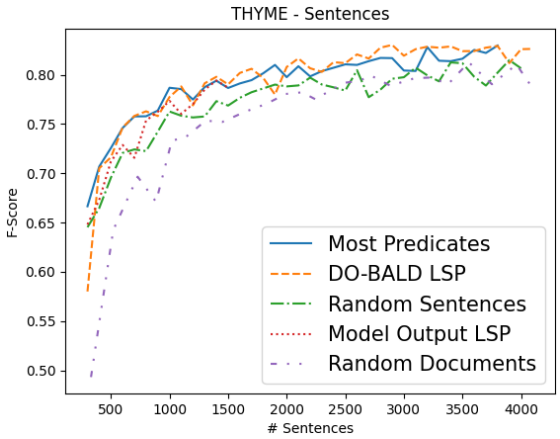


Figure 3: Learning curve of F-score by number of sentences in THYME training data.

The two selection methods that averaged the predicates performed worse than the baselines by sentences. One reason for this may be that the presence of frequent, but easily learned, predicates such as copulas inflating the average confidence of the sentence.

In terms of assessing the impact of whole-document selection, which is necessary for other NLP tasks such as coreference, compared to sampling individual sentences, the difference between sentences (4,126 vs 3,000, respectively) and predicates (11,598 vs. 9,333) required to reach our benchmark was significant. Sampling individual sentences reduces sentence annotation by 27% and predicate annotation by 20% to reach our benchmark.

7.2 THYME

Due to the weak performance of the AP aggregation method on the Ontonotes dataset, we did not perform those experiments on the THYME dataset.

As with our evaluation on the Ontonotes dataset, we can consider the annotation requirements to reach an F-score of 78.

The baseline sentence selection method obtains this benchmark after 1,600 sentences. Consistent with the results on the Ontonotes dataset, the DO-BALD LSP and MostPred methods are the most

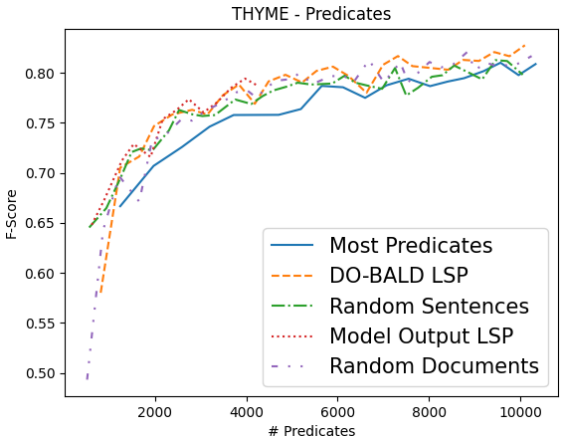


Figure 4: Learning curve of F-score by number of predicates in THYME training data.

efficient ways of selecting sentences, with both requiring **60% fewer sentences to train a model with a test F-score of 78**. The Output LSP method requires 18% fewer sentences.

With respect to predicates, once again we see the baseline RandSent performance (4355 predicates) significantly improved by DO-BALD LSP (20% - 4355 predicates) and Output LSP (16% - 3666 predicates), but MostPred is a detriment (30% *more* annotation - 5651 predicates).

8 Conclusions

Between the two proposed methods of aggregating predicate-argument structure scores into a single value to represent a sentence, averaging across them (AP) or only considering the weakest predicate (LSP), our results show the latter to be substantially better.

Both selecting sentences for the most predicates and selecting sentences with the predicate with the lowest DO-BALD agreement offer a significant 53%-60% decrease in the number of sentences required to train the model to a viable performance level. These findings are consistent for both the broad, general Ontonotes corpus and the niche colon cancer clinical note domain of the THYME corpus.

We assessed the performance of these selection strategies in terms of reducing both number of sentences and number of predicates annotated. Typically, the SRL annotation process of a large annotation project benefits most from a reduction of predicates, due to presenting annotators with batches of a specific predicate to annotate, thereby reducing the cognitive load of switching between different predicate frames. But in the case of projects attempting to develop new corpora with significant budget constraints that would most benefit from an active learning approach, the piecemeal nature of each annotation iteration makes this approach less viable and likely necessitates presenting annotators with the data sentence-by-sentence. In this case, reducing the number of sentences will have a more substantial impact than reducing the number of predicates.

While both DO-BALD LSP and the simpler strategy of selecting sentences with high predicate density provide significant reduction in sentence annotation, only DO-BALD LSP simultaneously reduced predicate annotation as well.

9 Future Work

Smaller batch sizes per iteration allow more efficient selection of data since the model is updated more frequently and we can reduce redundant information content within the batch that would waste annotation time. Using very small batches is not tractable in tasks that require long model training times. Koshorek et al. (2019) tested selection strategies on randomly sampled batches of data, rather than determining priority of individual instances, but that waters down the benefits of using the selection heuristic. In the future, we plan to investigate ways to balance syntactico-semantic redundancy with the model-based selection techniques in order to improve the learning rate for SRL, while reducing training time for each iteration.

We chose to use a random 200 sentences as our seed set, but the ideal amount and method of selection for active learning for SRL remains an open question. If too few sentences are chosen, or they're not sufficiently diverse, we may encounter the missed class effect (Tomanek et al., 2009), where the model becomes overconfident about instances that greatly differ from what's present in its current training pool, and fails to select them for annotation. On the other hand, selecting too large of a seed set negates the benefits of active learning. In future work we plan to explore unsupervised methods of selecting a semantically diverse seed set. Prior work (Dligach and Palmer, 2011) (Peterson et al., 2014) shows that language models may offer an unsupervised way of selecting rare verb instances and thus beneficial SRL instances.

Acknowledgments

We gratefully acknowledge the support of DARPA AIDA FA8750-18-2-0016 (RAMFIS), NIH: 5R01LM010090-09 THYME, Temporal Relation Discovery for Clinical Text, and NSF ACI 1443085: DIBBS Porting Practical NLP and ML Semantics from Biomedicine to the Earth, Ice and Life Sciences. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any government agency. Finally, we thank the anonymous IWCS reviewers for their insightful comments and suggestions.

References

Daniel Albright, Arrick Lanfranchi, Anwen Fredriksen, IV Styler, William F, Colin Warner, Jena D Hwang, Jinho D Choi, Dmitriy Dligach, Rodney D Nielsen, James Martin, Wayne Ward, Martha Palmer, and Guergana K Savova. 2013. Towards comprehensive syntactic and semantic annotations of the clinical narrative. *Journal of the American Medical Informatics Association*, 20(5):922–930.

Jinho Choi, Claire Bonial, and Martha Palmer. 2010. Multilingual Propbank annotation tools: Cornerstone and jubilee. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 13–16, Los Angeles, California. Association for Computational Linguistics.

Dmitriy Dligach and Martha Palmer. 2011. Good seed makes a good crop: Accelerating active learning using language modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 6–10, Portland, Oregon, USA. Association for Computational Linguistics.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753, Melbourne, Australia. Association for Computational Linguistics.

R. Duerr, A. Thessen, C. J. Jenkins, M. Palmer, S. Myers, and S. Ramdeen. 2016. The ClearEarth Project: Preliminary Findings from Experiments in Applying the CLEARTK NLP Pipeline and Annotation Tools Developed for Biomedicine to the Earth Sciences. In *AGU Fall Meeting Abstracts*, volume 2016, pages IN11B–1625.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Fariz Ikhwantri, Samuel Louvan, Kemal Kurniawan, Bagas Abisena, Valdi Rachman, Alfan Farizki Wicaksono, and Rahmad Mahendra. 2018. Multi-task active learning for neural semantic role labeling on low resource conversational corpus. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 43–50, Melbourne. Association for Computational Linguistics.

Omri Koshorek, Gabriel Stanovsky, Yichu Zhou, Vivek Srikumar, and Jonathan Berant. 2019. On the limits of learning to actively learn semantic representations. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 452–462, Hong Kong, China. Association for Computational Linguistics.

Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. 2013. Active learning for cross-domain sentiment classification. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 2127–2133. AAAI Press.

Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2017. GUIR at SemEval-2017 task 12: A framework for cross-domain clinical temporal information extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1024–1029, Vancouver, Canada. Association for Computational Linguistics.

Tim O'Gorman, Sameer Pradhan, Martha Palmer, Julia Bonn, Katie Conger, and James Gung. 2018. The new Propbank: Aligning Propbank with AMR through POS unification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated cor-

pus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Daniel Peterson, Martha Palmer, and Shumin Wu. 2014. Focusing annotation for semantic role labeling. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. Association for Computational Linguistics.

Katrin Tomanek, Florian Laws, Udo Hahn, and Hinrich Schütze. 2009. On proper unit selection in active learning: Co-selection effects for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 9–17, Boulder, Colorado. Association for Computational Linguistics.

Chenguang Wang, Laura Chiticariu, and Yunyao Li. 2017. Active learning for black-box semantic role labeling with neural factors. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2908–2914.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Jingbo Zhu and Eduard Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 783–790, Prague, Czech Republic. Association for Computational Linguistics.

SemLink 2: Chasing Lexical Resources

Kevin Stowe[1], **Jenette Preciado**[2], **Kathryn Conger**[3], **Susan Brown**[3],
Ghazaleh Kazeminejad[3], **James Gung**[3], **Martha Palmer**[3]
[1]Ubiquitous Knowledge Processing Lab (UKP Lab), Department of Computer Science
Technical University of Darmstadt
[2]SoundHound, Boulder, Colorado
[3]University of Colorado, Boulder
[1]stowe@ukp.informatik.tu-darmstadt.de
[2]jenette.preciado@gmail.com
[3]{firstname.lastname}@colorado.edu

Abstract

The SemLink resource provides mappings between a variety of lexical semantic ontologies, each with their strengths and weaknesses. To take advantage of these differences, the ability to move between resources is essential. This work describes advances made to improve the usability of the SemLink resource: the automatic addition of new instances and mappings, manual corrections, sense-based vectors and collocation information, and architecture built to automatically update the resource when versions of the underlying resources change. These updates improve coverage, provide new tools to leverage the capabilities of these resources, and facilitate seamless updates, ensuring the consistency and applicability of these mappings in the future.[1]

1 Introduction

Hand-crafted lexical resources remain an important factor in natural language processing research, as they can offer linguistic insights that are currently not captured even by modern deep learning techniques. SemLink is a connecting point between a number of different lexical semantic resources, providing mappings between different word senses and semantic roles, as well as a corpus of annotation (Palmer, 2009). SemLink has a variety of applications, from performing linguistic analysis of its component parts and their relations (Reisinger et al., 2015), extracting thematic role hierarchies (Kuznetsov and Gurevych, 2018), probing of linguistic formalisms (Kuznetsov and Gurevych, 2020), and computational methods for automatic extraction, improvement, and classification of computational lexical resources (Kawahara et al., 2014; Peterson et al., 2016, 2020).

SemLink incorporates four different lexical resources: PropBank (Palmer and Kingsbury, 2005), VerbNet (Kipper-Schuler, 2005), FrameNet (Baker and Lowe, 1998), and WordNet via the OntoNotes sense groupings (Weischedel et al., 2011).[2] Each resource has different goals and benefits: WordNet has the greatest coverage, with very fine-grained word senses grouped into small "synonym sets". These are linked to each other with semantic relations like hyponymy and troponymy. PropBank defines the argument roles for its verb and eventive noun senses, information not available in WN. FrameNet groups verbs, eventive nouns and some adjectives into semantic frames, with fine-grained argument roles defined for each frame. These frames are linked by various relations, such as "inherited by" and "used by". VerbNet groups verbs into more or less semantically coherent classes based on shared syntactic alternations. This resource uses fairly coarse-grained argument roles and provides a list of typical syntactic patterns that the verbs of a class prefer. In addition, VN provides a semantic representation for each syntactic frame, using the class's argument roles in a first-order-logic representation that incorporates Generative Lexicon subevent structure.

Semlink provides a bridge between these resources, allowing users to take advantage of their different features and strengths. For example, the mappings between the semantic role labels allow users to accurately convert annotations done with PB roles to VN roles and combine their respective data sets into a much larger corpus of training and test data.

The goal of SemLink is to link senses between resources, maximizing the effectiveness of each. It is composed of two primary assets: mappings

[1]https://github.com/cu-clear/semlink

[2]For the remainder of this work, we will refer to each by its acronym: PB, VN, FN, and ON, respectively.

Proceedings of the 14th International Conference on Computational Semantics, pages 222–227
June 17–18, 2021.

between resources, and a corpus of annotated instances. These are verbs in context that receive a PB roleset annotation, and VN class tag, a FN frame tag, and a sense tag based on the ON groupings.

The problem we address here is the constantly changing nature of these resources. They are evolving: new versions incorporate new semantics, new senses, better lexical coverage, and more consistent formatting. This makes it difficult to provide static links between them. SemLink has seen previous updates (Bonial et al., 2013) that improve consistency, but since that time many of the resources it links have undergone significant overhauls. Our work updates SemLink via four distinct contributions:

1. Automatic and manual updates to SemLink mappings based on new resource versions
2. Automatic addition of SemLink annotation instances, nearly doubling its size
3. Addition of sense embeddings and subject/object information
4. Release of software for automatic updates

2 Resources

A brief description of each resource in SemLink follows, along with the changes in each that have been implemented since the previous update.

2.1 PropBank

The previous version of SemLink incorporated PB annotation in the form of roleset mappings to VN classes and FN frames. It also contains gold annotation over sections of the Wall Street Journal corpus, with verbs annotated with their PB roleset. Each verb's arguments are annotated with their correct PB argument relations. These PB rolesets, mappings, and annotations remain core elements of SemLink, and we have expanded and updated each component for SemLink 2.0.

2.2 VerbNet

SemLink incorporates VN as an intermediary between the coarse-grained PB and fine-grained FN. Mapping files are provided that link PB rolesets to VN senses, which are then in turn linked to FN frames. The previous version of SemLink was built upon VN 3.2: this resource has since been updated to a new version (3.3), with substantial changes in class membership, thematic roles (Bonial et al., 2011), and semantics (Brown et al., 2018, 2019). We have incorporated these changes into SemLink 2.0 automatically where possible and manually where necessary.

2.3 FrameNet

The previous version of SemLink was built upon FN version 1.5; since then FN has released a new version (1.7), and this led to many consistency errors across resources. SemLink 2.0 provides manual updates to match the newest version of FN, as well as other consistency improvements.

2.4 OntoNotes Sense Groupings

The SemLink resource focuses less on these groupings than on PB, VN, and FN: it only includes ON as annotations on the provided instances. The ON resource has remained consistent since the release of the previous SemLink version, and thus the instance annotations remain valid.

3 Improvements and Additions

SemLink incorporates these resources via mapping files (for PB, VN, and FN) and predicate instance annotations (including all four resources). We will now overview each of these artifacts, highlighting the updates in our new release and the tools and practices used to generate these updates.

3.1 PB to VN mappings

The previous version of SemLink contains two files comprising the mappings from PB to VN: a mapping file that links PB rolesets to VN senses, and a mapping file linking PB arguments (ARG0, ARG1, etc) to VN thematic roles (Agent, Patient, etc). These files contain a growing number of inaccuracies as the resources have been updated, particularly with PB's update to unified frame files and VN's update to the version 3.3.

To deal with these constant updates, we've improved the system that automatically generates these mapping files based on ground-truth mappings present in PB. The PB frame files contain links from each roleset to possible VN classes: this allowed us to generate a large number of accurate mappings based purely on the information present in PB. The main update to this architecture is the development of VN class matching. We can now find if verbs have moved between classes, allowing the automated updater to find more valid instances. This system incorporates soft class matching for when verbs moved between VN subclasses, as well as exploiting available WordNet mappings in VN to identify if a verb moved to a new class.

The mappings generated by this system are not exhaustive: the ever-changing nature of the two projects makes it impossible to have all possible mappings. One of the primary goals of SemLink is to ensure that the most consistent possible mappings between resources is available, and our update helps to foster this consistency by making available our software for updating and evaluating the accuracy of these mappings. This is done by automatically generating mappings from PB to VN based on PB frame files, combining them with the previous version of manual mappings, and checking both of these mappings for consistency.

This process produces an update mapping resource from PB to VN. While these mappings don't eliminate the need for some manual annotation, as substantive changes can require new mappings to be added or deleted, it does allow the resource to be consistently and automatically updated while preserving only valid mappings.

3.2 VN to FN mappings

SemLink contains similar mapping files from VN to FN: one mapping from VN senses to FN frames, and one mapping from VN thematic roles to FN's typically more specific frame elements. As with PB and VN, FN has seen a significant update (to version 1.7) since the previous SemLink release, and these mappings files have become outdated.

Unlike PB, neither VN nor FN implicitly keeps track of mappings to the other resource: the only linking between them is in SemLink's mapping files. Therefore, for these files, we employed a semi-automated system to identify incorrect mappings and make updates. We run a script to identify whether VN class/role and FN frame/frame elements are valid. This is done by checking if the classes, roles, frames and frame elements still exist in the current version of the resource, and then checking if the roles and frame elements are still valid for the given classes and frames. We then pass them to annotators if there are errors. This was done for all of the mappings in the previous version, yielding 2,387 valid mappings, 160 of which came from manual re-annotation. These mappings were then compiled to form the new VN to FN mapping file for SemLink 2.0.

For both PB to VN and VN to FN mappings, we employed automatic procedures that allowed us to update outdated SemLink instances to match the current resources. However, these updates are necessarily not comprehensive: we only updated instances for which we could identify automatic mappings between old and new. If the resources changed in unpredictable ways (ie. a sense tag changed itself changed meanings), these mappings may still be inconsistent. We therefore include for each instance in SemLink 2.0 and indicator for each mapping whether it was derived from an automatic procedure or manually annotated.

Previous Version		**SemLink 2.0**		
Resource	Count	Count	Added	Coverage
PB	75k	148k	73k	.99
VN	75k	97k	22k	.65
FN	37k	42k	5k	.28
ON	28k	48k	21k	.33
Total	75k	149k	74k	+98%

Table 1: Summary of Annotation Updates to SemLink

3.3 Annotations

The second artifact produced for SemLink is a set of annotations. These consist of predicates annotated with PB frames, VN senses, FN frames, ON groupings, and each resource's representation of the predicates' arguments. An example of an annotation instance is shown in Figure 1.

3.3.1 Updates to Previous Annotations

All instances underwent an automatic update process based on our revision of mapping resources. The sense tags for each resource are validated, and automatically updated via mappings if errors are found. This process is repeated for role arguments.

This was done for the 74,920 instances available with the previous SemLink. In order to keep the resource as large and as flexible as possible, as long as an instance had a PB roleset, we didn't remove instances with invalid mappings: rather, we kept these instances and left the additional information (VN, FN, etc) as "None". This allows us to maintain the size of the resource and while preserving only the accurate annotations.

3.3.2 New Annotations

In addition to updating the previous annotations, we were also able to leverage additional annotation projects to expand the scope of the SemLink resource. We gathered 72,822 additional instances from the OntoNotes 5.0 release annotated with the unified PB rolesets (Weischedel et al., 2011), and employed our updated mapping files to automatically attribute VN and FN information to them. We also collected 5,300 instances that were manually

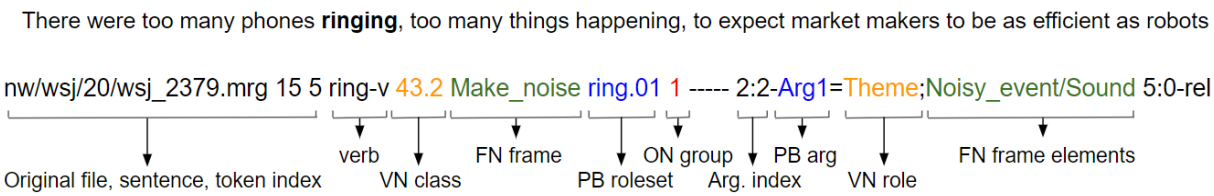


Figure 1: SemLink annotation instance for the verb "ringing" in the above sentence.

annotated with VN classes (Palmer et al., 2017), and extracted PB and FN information from these based on mapping files.

Similar to the updates above, we automatically check these instances to determine if their annotations were valid (the class, sense, or frame still exists) in the modern versions of each resource. and then added them to SemLink's annotation corpus. A summary of the update to the annotations is shown in Table 1.

From this summary we can see substantial improvements to the dataset across all resources, with the greatest impact coming from the new annotations. However, as we automatically add instances based on PB and VN annotation, they often lack mappings to the other resources. This, combined with the fact that some VN and FN annotations were removed due to inconsistency with the latest versions, leads to a decrease in the percent of instances tagged with each particular resource, despite the increase in total annotations.

3.4 VN Tools

In order to ensure the applicability of these mappings and lexical resources, we include two additional components: sense embeddings and common arguments. These are based on VN, as it directly links to PB and FN.

3.4.1 VN Embeddings

We train embeddings based on VN in a style similar to that of (Sikos and Padó, 2018). We tag a corpus of 4.5m sentences from Wikipedia with a VN class tagger (Palmer et al., 2017). We then learn embeddings for both VN classes and specific VN senses by modifying the resulting corpora. First, to generate generic VN class embeddings, we replace the verb directly with its labeled class. This allows the embedding model to learn a representation that generalizes over all instances of a particular VN class, and provides an abstraction away from the individual lexical items. Second, to generate sense-specific word embeddings, we concatenate the class information along with the verb. This yields more specific embeddings that concretely reflect contextual usages of the given verb. The resulting sentences can then be fed to a lexical embedding algorithm of choice: here we use GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) embeddings of size 100.

These embeddings have proven an effective addition to traditional embeddings for classification tasks, and even have advantages over contextual embeddings. Stowe (2019) show that incorporating VN-based sense embeddings into LSTM-based metaphor detection improves results over using ELMo embeddings alone, despite the fact that the contextualized ELMo embeddings should independently capture sense information (Peters et al., 2018).[3]

These methods for learning embeddings are broadly applicable to any lexical resource, and are adaptable to changing versions; the embeddings provided are trained using VN 3.3, and as we provide links from VN to PB and FN, we further believe that the accompanying embeddings can be directly linked to these two resources.

3.4.2 VN Common Arguments

In addition to embeddings, we also collect argument information based on VN class tagging. We collect for each class the most frequent subjects and objects of verbs tagged with that class. This is done by tagging the above Wikipedia corpus with VN classes, then using a dependency parser to extract subject and object information (Chen and Manning, 2014). This automated procedure does inherently introduce noise, but it allows us to form a general idea of kind of arguments that typify the semantic roles and to better understand the syntactic and collocational properties of verb classes. Practitioners who are researching verb classes can use these to better understand from a quantitative perspective what kinds of subjects and objects are likely to ap-

[3]Note that these results are from embeddings trained on VN version 3.2; they have since been updated to version 3.3

pear with given verb classes, further facilitating research into lexical semantics.

3.5 Software

In order to manage these updates, we've built a substantial number of infrastructure components to support the interaction between these resources. This includes interfaces to each resource, to SemLink, and tools for making automatic updates based on different versions. The SemLink scripts have the flexibility to use and compare various different versions of each resource; this allows us to quickly update SemLink to new versions.

This software will be released along with the new version via GitHub, with the hope that the community can maintain and improve its functionality as necessary, and to allow researchers to be able to easily interact with both the resources linked and the SemLink resource itself. Critically, this resource will mitigate the damage of future changes to each individual resource, as SemLink can painlessly be updated to accommodate new versions.

4 Conclusions and Future Work

Our updates to SemLink consist of four main components. (1) We update SemLink data to match the current versions of each resource through automatic and manual methods. (2) We add annotations to improve the coverage of the resource. (3) We add sense embeddings and argument information. (4) We provide automatic tools to allow the SemLink resource to be consistently updated. As these lexical resources are always changing, these tools are necessary for the resource to remain viable, and while the process of linking semantic resources can likely never be fully automated, these tools can assist in this process. This work then comes with two artifacts: the new SemLink resource (mapping files and annotations) as well as architecture for updating and managing SemLink.

The coverage is by no means complete and many lexical items in each resource contain no viable mappings. Manual annotation of links between resources is essential for the success of the SemLink resource: while we can automatically filter out inaccurate mappings when resources change, this leaves blind spots where we have incomplete mappings, and manual annotation is currently the most accurate way to cover these gaps.

Another direction of future work is evaluating the usefulness of these linked resources. While there have been evaluations comparing the three semantic role labelling frameworks provided via PB, VN, and FN (Hartmann et al., 2017), a full-scale evaluation of the links between them is yet to be done, and may provide valuable insight not only into how to best improve SemLink, but also into how these kinds of linked resources can be best employed. While modern NLP focuses largely around end-to-end models that implicitly capture semantic relations, there is still a role for hand-curated lexical resources to play, and we believe SemLink can be an effective resource for those studying computational lexical semantics, word sense disambiguation and semantic role labelling, and other tasks requiring linked lexical resources.

5 Acknowledgements

We gratefully acknowledge the support of DTRAl-16-1-0002/Project 1553695, eTASC - Empirical Evidence for a Theoretical Approach to Semantic Components and DARPA 15-18-CwC-FP-032 Communicating with Computers, C3 Cognitively Coherent Human-Computer Communication (sub from UIUC) and Elementary Composable Ideas (ECI) Repository (sub from SIFT), and DARPA FA8750-18-2-0016-AIDA – RAMFIS: Representations of vectors and Abstract Meanings for Information Synthesis. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, DTRA, or the U.S. government.

References

Fillmore C.J. Baker, C. F. and J.B. Lowe. 1998. The Berkeley FrameNet project. pages 86–90, Montreal, QC. COLING-ACL '98.

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.

Claire Bonial, Kevin Stowe, and Martha Palmer. 2013. Renewing and revising SemLink. In *Proceedings of the 2nd Workshop on Linked Data in Linguistics (LDL-2013): Representing and linking lexicons, terminologies and other language data*, pages 9 – 17, Pisa, Italy. Association for Computational Linguistics.

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. Verbnet representations: Subevent semantics

for transfer verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163.

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)*.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Silvana Hartmann, Éva Mújdricza-Maydt, Ilia Kuznetsov, Iryna Gurevych, and Anette Frank. 2017. Assessing SRL frameworks with automatic training data expansion. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 115–121, Valencia, Spain. Association for Computational Linguistics.

Daisuke Kawahara, Daniel W. Peterson, and Martha Palmer. 2014. A step-wise usage-based method for inducing polysemy-aware verb classes. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1030–1040, Baltimore, Maryland. Association for Computational Linguistics.

K Kipper-Schuler. 2005. VerbNet: A broad-coverage, comprehensive verb lexicon.

Ilia Kuznetsov and Iryna Gurevych. 2018. Corpus-driven thematic hierarchy induction. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 54–64, Brussels, Belgium. Association for Computational Linguistics.

Ilia Kuznetsov and Iryna Gurevych. 2020. A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Gildea D. Palmer, M. and P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. volume 31, pages 71–106.

M. Palmer. 2009. Semlink: Linking PropBank, VerbNet and FrameNet. Pisa, Italy. Proceedings of the Generative Lexicon Conference.

Martha Palmer, James Gung, Claire Bonial, Jinho Choi, Orin Hargraves, Derek Palmer, and Kevin Stowe. 2017. The Pitfalls of Shortcuts: Tales from the word sense tagging trenches. *Essays in Lexical Semantics and Computational Lexicography - In honor of Adam Kilgarriff*.

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.

Matthew 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.

Daniel Peterson, Jordan Boyd-Graber, Martha Palmer, and Daisuke Kawahara. 2016. Leveraging VerbNet to build corpus-specific verb clusters. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 102–107, Berlin, Germany. Association for Computational Linguistics.

Daniel Peterson, Susan Brown, and Martha Palmer. 2020. Verb class induction with partial supervision. In *Proceedings of the Thirty-fourth AAAI Conference on Artificial Intelligence*, New York City, NY.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488.

Jennifer Sikos and Sebastian Padó. 2018. Using embeddings to compare FrameNet frames across languages. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 91–101, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kevin Stowe. 2019. Syntactic and semantic improvements to computational metaphor processing.

R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradan, L. Ramshaw, and X. Nianwen. 2011. OntoNotes: A Large Training Corpus for Enhanced Processing. *Handbook of Natural Language Processing and Machine Translation: Global Automatic Language Exploitation*, pages 53–63.

Variation in framing as a function of temporal reporting distance

Levi Remijnse, Marten Postma, Piek Vossen
Vrije Universiteit Amsterdam
De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
l.remijnse,m.c.postma,piek.vossen{@vu.nl}

Abstract

In this paper, we measure variation in framing as a function of foregrounding and backgrounding in a co-referential corpus with a range of temporal distance. In one type of experiment, frame-annotated corpora grouped under event types were contrasted, resulting in a ranking of frames with typicality rates. In contrasting between publication dates, a different ranking of frames emerged for documents that are close to or far from the event instance. In the second type of analysis, we trained a diagnostic classifier with frame occurrences in order to let it differentiate documents based on their temporal distance class (close to or far from the event instance). The classifier performs above chance and outperforms models with words.

1 Introduction

To understand streams of news and blogs in terms of the ways in which events can be framed, we need to model how these streams develop over time in relation to the common ground that is created. The common ground between interlocutors plays an essential role in how they refer to real-world event instances.[1] Following pragmatic theory (Grice, 1975; Horn, 1998; Clark et al., 1977), when this common ground is low, the speaker, in an attempt to be cooperative, needs to be as informative as possible, using detailed and marked descriptions of the main event instance. When the common ground is high, the speaker can optimally use less marked expressions and hence background the main event instance in order to foreground related events with a higher informative value (see also the grounding principles of Grimes (2015). The less marked expression then implicates prior knowledge of the event instance, which has become unnecessary to explicate. This is shown in the next two examples that report on instances of the same event type at different points in time (the-day-before versus a-week-ago). In example (1a), reference to the event instance is marked by using multiple indefinite expressions of different syntactic categories in reference to subevents: a *shooting* in which a man *died*. In example (1b), reference to the event instance is restricted to one definite expression *last week's murder*, which presupposes the event instance as shared knowledge and implicates its details. The rest of the text in the example focuses on other events.

(1) a. One man died in a shooting early Thursday morning in southwest Houston. [2]

b. One of the four suspects wanted in last week's murder of Keith Thompson was arrested Wednesday morning at a home in Springfield, according to the Jacksonville Sheriff's Office. [3]

Given this theory about variation in referential expressions, we can expect that, from the onset of an unexpected real-world event instance (e.g., a shootout), the constantly developing narrative of related events (e.g. pursuits, arrests, trials) will enforce these mechanics of foregrounding and backgrounding based on growing mutual knowledge. In other words, the common ground determines the extent to which the speaker is able to background (i.e., use minimal expressions or implicatures) the main event in order to foreground related subjects.

[1] In this paper we use the term *event instance* for event instances of a specific event type, e.g., an instance of shooting

[2] https://www.chron.com/houston/article/Shooting-levaes-man-dead-in-SW-Houston-6688587.php, published on the same day as the event instance.

[3] http://www.news4jax.com/news/crime/1-arrest-in-westside-murder, published a week after the event instance.

Proceedings of the 14th International Conference on Computational Semantics, pages 228–238
June 17–18, 2021.

Suppose we want to test these principles empirically by examining references in a large dataset, e.g., a referentially grounded corpus. This requires a large collection of documents all referencing single event instances, with a large spread of temporal distance between the publication dates and the event instance date. However, most of the available co-referential corpora hardly contain multiple reference texts for the same event instance, let alone with a strong range of publication dates (Ilievski et al., 2016; Postma et al., 2016).

In this paper, we propose to overcome the data sparsity by merging data of event instances of the same event type to study foregrounding and backgrounding phenomena. We assume that it takes approximately the same amount of time for information, on for instance shooting events, to become common ground between members of a society. Furthermore, such a specific event type activates a coherent set of conceptual properties typically used in reference (Vossen et al., 2020; Morris and Murphy, 1990). Yet, this use of reference might depend on mutual knowledge. Based on the discussed pragmatic principles, we claim that both referential expressions and their meanings vary across documents with different temporal distances to event instances: over time, relevant information about the event instance is left implicit as a means to background reference to the event instance and foreground reference to novel information.

In order to find evidence for our claim, we use FrameNet (Fillmore et al., 2003) as a proxy to characterize event semantics. Our prediction is that those frames typically associated with an event type, called *typical frames*, will also show a different foregrounding and backgrounding distribution as a function of the increased common ground. We expect that subevents of the event instance are foregrounded in texts with little temporal distance, whereas related disjoint events are expected to be foregrounded in texts with large temporal distance. This difference should be reflected by their frames.

To test this hypothesis, we applied a method based on Grootendorst (2020) to learn frame typicality rates for event types from a large collection of news reports that were processed with an automatic frame-labeler (Swayamdipta et al., 2017). Furthermore, we trained a Linear Support Vector Machine classifier to distinguish between referential texts with close temporal distance and further temporal distance on the basis of the typical frames evoked by the texts. We contrast this classifier against models trained on words. We provide evidence that frame distributions are learned by the classifier to perform the task, whereas this is lesser the case for word based models. Our analysis of the results shows that the typical frames evoked in texts with a short temporal distance are backgrounded in texts of larger temporal distance by means of implicature.

The main contributions of our work are:

- We present *HDD* (*Historical Distance Data*), an extensive corpus of reference texts for event instances grouped under event types, with a large spread of temporal distance to the event instance;
- We derive a ranking of typical frames cross-event types;
- We show that frames are more informative than their predicates in training a Linear Support Vector to predict the temporal distance class given a document;
- We show that when contrasting frames for an event type between temporal distance classes, the top ranked frames reflect foregrounded topics.

Our results will help future systems in detecting events in texts and their framing but also help the computational modeling of pragmatics and implicatures.

This paper is structured as follows. We first describe relevant past work in Section 2. We then introduce our methodology in Section 3. Section 4 provides the results of our experiments, which we discuss in Section 5. We conclude in Section 6.

2 Background

In this section, we discuss previous work that has been done with respect to event foregrounding, (2.1 FrameNet (2.2), temporal distance (2.3) and event corpora (2.4).

2.1 Event foregrounding

Different studies have focused on the recognition and characterization of foregrounded events. On the sentence level, foregrounded events show high probability of appearing in main clauses, being actively voiced and having a high transitivity (Kay

and Aylett, 1996; Decker, 1985). These observations are applied by Upadhyay et al. (2016) to identify the most significant event in a news article.

On the discourse level, it has been observed that normalized frequencies of co-referential event mentions play an important role in detecting the central event of a document (Filatova and Hatzivassiloglou, 2004a,b). According to Choubey et al. (2018), another crucial factor is the scope of the chain of co-referential mentions throughout the document. These mentions foreground subevents in reference to the central event. The discussed examples in Choubey et al. (2018) show that backgrounded events scarcely occur throughout the document, supporting the reader in grounding the foregrounded central event in a commonly known prior event. In line with their proposal, both *died* and *shooting* in (1a) form a chain of foregrounded subevents that make reference to the central event of the document. In (1b), *arrested* is the foregrounded central event, but *murder* is a backgrounded event.

In this paper, we propose that the mentions that foreground the central event instance also activate a coherent set of FrameNet frames typically used in reference to the event type. In analyzing HDD, we find that this set of typical frames is different for documents written long after the event instance, as an effect of backgrounding that event instance and foregrounding related disjoint events.

2.2 Frames as implicatures

We use FrameNet as a proxy to characterize event semantics in this paper.[4] FrameNet is a lexicographic project anchored in the paradigm of frame semantics (Fillmore et al., 2003; Fillmore and Baker, 2010; Baker et al., 2003). Its lexical database consists of over 1200 semantic frames. Each frame is considered a schematic representation of a situation involving semantic roles, and is assumed to be *evoked* by a *lexical unit*, i.e., a lemma in one of its senses. Each frame exhibits an inventory of lexical units. Below, (1) is extended with FrameNet annotations.

(2) a. One man DEATH$_{\odot}$*died* in a KILLING$_{\odot}$*shooting* [...]

b. One of the four SUSPICION$_{\odot}$*suspects* wanted in last week's KILLING$_{\odot}$*murder* of Keith Thompson was ARREST$_{\odot}$*arrested* [...]

With respect to inferential relations between frames, literature largely focuses on different types of *frame-to-frame relations*, i.e., asymmetric relations between two frames. The FrameNet database registers frame-to-frame relations between the frames to form a network. For example, the Precedes relation specifies a sequential order between two frames, e.g., ARREST shows a Precedes relation to ARRAIGNMENT (Ruppenhofer et al., 2010). Thus, when ARRAIGNMENT is evoked in a document, we can infer ARREST as an implicature. Frames connected through Precedes relations form a coherent set in which any frame implicates the "preceding" frames. The output of our experiment can be used as input for FrameNet to form more of these cohesive sets of frames with temporal relations.

2.3 Temporal Distance

The effect of the temporal distance between a reference text's publication date and the event date on variation in reference has been explored in a few studies.[5] Staliūnaitė et al. (2018) focus on co-reference to entities in the *New York Times Annotated Corpus* (Sandhaus, 2008), which contains articles spanning 20 years. They show that as a function of common knowledge, references to the same entity become definite, of shorter length, i.e., less marked, and with less use of appositives.

Cybulska and Vossen (2010) carried out a statistical analysis on a corpus of reference texts concerning the Srebrenica Massacre. The corpus consisted of 52 news articles (evenly distributed over two news journals) published within a time range of 10 days after the event, and 26 "historical" texts published years later. They created a word-based frequency distribution of references. They showed a strong discrepancy in type-token ratio between the two conditions of temporal distance: the sub-corpus written close to the event shows a higher number of word types than the sub-corpus written years later. The authors conclude that difference in temporal distance correlates with variation in language use. Short temporal distance leads to more variation in descriptions, due to focus on sub-

[4] https://framenet.icsi.berkeley.edu/fndrupal/

[5] On discourse level, referential variation as an effect of common ground has been studied more intensively. See Yoshida (2011); Markert et al. (2012); Del Tredici and Fernández (2018).

events, while longer distance leads to less variation in descriptions due to focus on the main event.

Our research aims to contribute to Cybulska and Vossen (2010) in the following ways. Our HDD is restricted to news articles under the assumption that variation in reference can also be observed within genres. Hence, the potential confounding variable of variation between text genres in their study is eliminated. Second, HDD covers reference texts of multiple event instances of a single event type. Third, we use FrameNet to measure variation in typically evoked frames on top of expressions. Finally, the dimension of temporal distance in our experiments ranges to 30 days after the event instances, instead of years.

2.4 Event corpora

In event co-reference research of the last decade, the corpus datasets show a small number of documents referencing events. Vossen et al. (2018) provide an overview of the nine governing text corpora (e.g., OntoNotes (Pradhan et al., 2007), ECB (Bejan and Harabagiu, 2010), ACE2005 (Peng et al., 2016)) and observed that their sum consists of less than four thousand documents. The number of mentions of events is small within documents (10 mentions per document on average) and only a subset of the corpora contains cross-document event co-reference. Also more recent attempts to manually create annotations for all sentences in articles did not cover a high number of documents (Cybulska and Vossen, 2014; Song et al., 2015; O'Gorman et al., 2016).

Since we need a substantial amount of event reports of the same event type for our experiment, we used the Multilingual Wiki Extraction Platform (MWEP) (Vossen et al., 2020) to obtain a large corpus of referentially grounded news texts. MWEP follows the data-to-text method and takes event types as input to query Wikidata (Vrandečić and Krötzsch, 2014) for event instances. For the obtained event instances, MWEP crawls the corresponding Wikipedia pages and their primary reference texts. These pages are processed by NLP systems, resulting in a corpus of multilayered linguistic annotation files.

3 Methodology

In this section, we describe the methodology used for both the between-event type and within-event type experiments.[6] This includes the resources, (3.1), data processing (3.2), contrastive analysis (3.3, hypotheses (3.4) and evaluation (3.5).

3.1 Resources

The model used to describe our data relies on three main concepts: event type, incident, and reference text. Let E be a set of event types, let I be a set of real-world event instances, and let R denote a registry of reference texts. Each real-world instance $L_i \in I$ is an instance of one or more event types. Also, there can be reference texts that refer to a particular real-world instance L_i. For example, the reference text *Significance of Orlando gunman calling 911 during standoff*[7] refers to the real-world event instance *Orlando nightclub shooting*[8], which is an instance of several event types according to Wikidata, including *mass shooting*[9] and *mass murder*.[10] Based on Wikidata, the incident date can be obtained.

Commonly, our pointer to a reference text is an URL. We apply the following steps to locate, retrieve, and process the reference text. First, we make use of the Internet archive Wayback Machine[11]. Please note that this step is not successful for all URLs. Second, we apply news-please (Hamborg et al., 2017) to crawl the reference text as well as the publication date. Third, we process the text using spaCy (Honnibal et al., 2020) for sentence splitting, tokenization, lemmatization, and dependency parsing. Finally, we apply Open-SESAME (Swayamdipta et al., 2017), which was retrained in order to be used. The collection process results in a document with annotations for various NLP tasks, including frame identification, and the publishing date of the document is typically known.

We make use of two routes to obtain data for HDD according to our model. We apply MWEP on three Wikidata event types: *presidential election* (Q858439), *storm* (Q81054), and *music festival* (Q868557). The second source is a Kaggle dataset called *Gun Violence Data* (Ko, 2018), which con-

[6] the code is available at https://github.com/clt1/HDD_analysis.

[7] https://www.cbsnews.com/news/orlando-shooting-investigation-gunman-omar-mateen-911-call/

[8] https://www.wikidata.org/wiki/Q24561572

[9] https://www.wikidata.org/wiki/Q21480300

[10] https://www.wikidata.org/wiki/Q750215

[11] https://web.archive.org/

tains approximately 260,000 real-world instances regarding the event type *gun violence*, containing links between reference text URLs and the real-world instances. The four event types are selected due to their differentiation of conceptual properties, which makes them suitable for a contrastive analysis. The descriptive statistics of applying our retrieval and processing software are shown in Table 1.

For each of the four selected event types, Table 1 presents the descriptive statistics. MWEP is capable of generating data for various different event types. However, the number of incidents and reference texts are limited, while the number of reference texts per incident is relatively high. The gun violence dataset, on the other hand, provides a high number of incidents for one specific event type, i.e., gun violence, but the number of texts per incident is relatively low.

Finally, we compute the *temporal distance*, which we define as the number of days between the incident date and the publishing date of a reference texts that makes reference to it. We visualize the distribution of temporal distance for the event type gun violence (Q5618454) for those reference texts for which we were able to obtain a publishing date, see Figure 1.

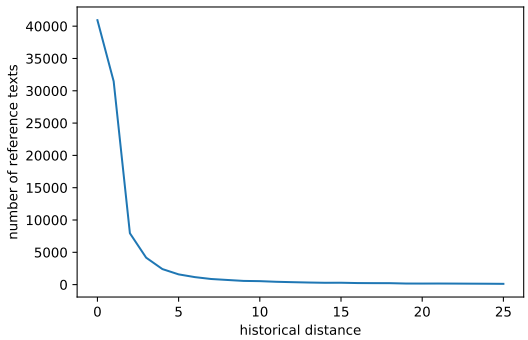


Figure 1: The distribution of the temporal distance is shown for the reference texts that are published within 25 days of the incident, which holds for approximately 90% of the reference texts for the event type gun violence (Q5618454)

Figure 1 visualizes the distribution of temporal distance for the event type gun violence (Q5618454). Most texts are published at the day of the incident. As time passes, the number of documents written about an incident decreases. Still, more than 10,000 are written after 25 days have passed.

3.2 Processing the corpus

We chose to train our diagnostic classifier on the gun violence data, since this subcorpus of HDD shares the largest volume of texts. The following steps were taken to preprocess the data for training. First, a subset of 6,290 documents containing less than 10 annotated frames were removed. These are most likely documents whose URLs were not successfully retrieved by the Wayback Machine, resulting in raw text of error messages, cookies etc. We also removed a subset of 16,237 documents for which news-please was not able to retrieve the publication date.

Next, we specified two temporal distance classes: "day 0" and "day 8-30". The remainder of documents were categorized into those classes according their publication date. After this step, day 0 covers 38,930 documents and day 8-30 covers 6,291 documents. We chose to train a Linear Support Vector model with both this unbalanced variant and a balanced variant in which the documents of day 0 are reduced to a randomized set of equal size as day 8-30.

Per document, both the frequencies of the frames and of their predicates were extracted and separately implemented as features in a data frame. A column was added with the temporal distance classes as labels. Each data frame was split into a training set (80%), a development set (10%) and a test set (10%). We ended up with data frames for both predicates and frames in a balanced and unbalanced corpus condition (4 experiments).

For each experiment, LinearSVC from Scikit Learn (Pedregosa et al., 2011) was used to train a Linear Support Vector with both the features of the experiment and the temporal distance classes as labels. This diagnostic classifier was applied to the test set and evaluated as a multi-class task per experiment.

3.3 Typical frame detection

The HDD corpus was first used for a contrastive analysis between event types and between temporal distance classes of gun violence. The aim was to derive typical frames, i.e., frames that are typically evoked in reference to a certain event type. The following steps were taken to process the data. We selected the data for the event types presidential election, storm and music festival, from which a total set of 57 documents containing less than 10 annotated frames were removed. From the event type

event type	# Li	# of Ri	Avg # of Ri per Li
presidential election (Q858439)	111	408	3.7
storm (Q81054)	60	256	4.3
music festival (Q868557)	13	205	15.8
gun violence (Q5618454)	103,090	123,659	1.2

Table 1: Descriptive statistics regarding the key data concepts of the data forming HDD, used for the experiments. The first three rows originate from using MWEP to obtain data, whereas Gun Violence Data (Ko, 2018) is used for the data of the last row. The first column indicates the event types and the Wikidata identifier of the event type. The second column, L_i, indicates the number of real-world incidents that belong to the event type. The third column, R_i, presents the total number of reference texts, each referring to one of the real-world incidents. Finally, the average number of reference texts per real-world event instance are shown.

gun violence, we used the documents for which the publication date could not be retrieved. Next, the corpus was randomly sampled by equalizing the volume of texts to the smallest collection (N=191), resulting in an equal amount of reference texts per event type.

For the analysis between event types, all frame annotations were extracted from the documents and compiled per event type. Next, we apply an **FFICF** metric (a derivative of C-TFIDF), where *FF* stands for the frame frequency in a subcorpus, and *ICF* is the inverse collection frequency (Vossen et al., 2020). This results in an FFICF score (henceforth *typicality score*) per frame per event type.

C-TFIDF was designed by Grootendorst (2020) with the purpose of determining the topic of a word cluster based on the set of highest scoring words. We have the advantage that, due to the data-to-text approach, the documents in HDD are already clustered based on predefined topics, i.e., event types. It follows that if we apply C-TFIDF to our corpus, we merely have to validate the highest scoring frames. Adapted to collections of frames, the mathematical model reads as follows:

$$FF - ICF_i = \frac{t_i}{f_i} \times log \frac{m}{\sum_j^n t_j} \quad (1)$$

where the frequency of each frame *t* is extracted for each event type *i* and divided by the total number of frames of that event type. Then, the total number of documents *m* across event types is divided by the total frequency of the frame *t* across event types *n*.

We applied this metric to our sampled subset of HDD and ranked the typicality scores per event type. Furthermore, we performed a similar FFICF procedure between the temporal distance classes of gun violence.

3.4 Hypotheses

1. FFICF between event types
We expect the frames with high typicality scores to differ between event types. The frames with the lowest typicality scores may be similar across event types, being a-typical.

2. FFICF between temporal distance classes
We expect the frames with high typicality scores to differ between texts from the same event type gun violence but from different temporal distance classes due to foregrounding and backgrounding.

3. Training and testing the Linear SVM
We expect the diagnostic classifier to perform above chance in predicting the temporal distance class given a document, when the texts are represented by the typically evoked frames. With frame frequencies as features, the model will outperform word based models.

3.5 Evaluation

In order to validate the outcome of the contrastive analysis between temporal distance classes, we presented two annotators with frames from the subcorpus of gun violence for which the publication dates could not be retrieved. Frames with three or less occurrences across this subcorpus were filtered out. For each of the remaining 282 frames, the annotators were asked to provide a binary judgment about whether it is typical in reference to an incident of gun violence at day 0. We utilized the notion *narrative container* (NC) from Pustejovsky and Stubbs (2011), i.e., the scope between the onset of the event instance and the document creation time, to estimate the possible subevents that have a high chance of being referred to in a document on day 0. The annotators had to judge whether each frame is part of the NC. We used Cohen's kappa (Cohen, 1960) to obtain a measure of inter-

annotator-agreement. We expect that the frames annotated as part of the NC of day 0, also occur in the top rank of FFICF scores for this class, whereas frames annotated as falling outside of the NC occur in the top rank of FFICF scores for day 8-30.

We evaluated the output of the diagnostic classifier in a multi-class classification report with precision, recall and F1-score in addition to accuracy, macro average and weighted average.

4 Results

For the contrastive analysis between event types, Table 2, shows the top and bottom ranked FFICF scores for the event types gun violence and music festival. The top ranked frames differentiate between event types and appear to reflect their typical properties. In contrast, the bottom ranked frames are the same for both types and reflect generic event properties.

For the contrastive analysis between temporal distance classes, Table 3 shows the top and bottom ranked FFICF scores between the two temporal distance classes of the event type gun violence. LAW_ENFORCEMENT_AGENCY, KILLING and CATASTROPHE, which were in the top ranking in Table 2, ended in the bottom ranking here. Furthermore, except for two frames, the top ranking of both classes in Table 3 is occupied by different frames.

The annotators show a Cohen's kappa of .48, which is moderate. However, their judgments on the frames in the top and bottom ranking of FFICF ratings in Table 3 show a rather high agreement (20 out of 26 frames, 77%). Half of the top ranked frames in day 0 are annotated as part of the NC of day 0, and almost all top ranked frames at day 8-30 are annotated as not belonging to that same NC. Note that the three frames that are both in the top ranking of scores between event types and at the bottom ranking of scores between temporal distance classes, are also annotated as part of the NC.

Table 4 displays the evaluation report of the experiments with the Linear SVM classifier. In the unbalanced conditions, the accuracy is above 0.85, but biased towards the performance for day 0. The model performed below chance in predicting day 8-30. For frames, the performance in this class is higher than for predicates. In the balanced conditions, the performance decreases for day 0, but increases for day 8-30. For predicates, the model performs around and above chance, with higher recall for day 0 and lower recall for day 8-30. For frames, the F1 is above 0.75, with consistent precision and recall.

5 Discussion

In Table 3, we find that SHOOT_PROJECTILES and JUDGMENT_COMMUNICATION remain in the top ranking, each in a different class. All other frames in the top ranking are typically used in reference to the events of their respective class. Many of those frames can be considered typical for gun violence (e.g., EXPERIENCE_BODILY_HARM, JUDICIAL_BODY), but their evocation is subjected to the temporal distance class. The frames on day 0 refer to subevents of the central event instance, while the frames on day 8-30 refer to related disjoint events, as is generally validated by the annotators. We interpret this variation as an effect of foregrounding and backgrounding. Most typical frames on day 0 are backgrounded in day 8-30 due to the high common ground. They are pragmatically implicated in order to foreground the frames of day 8-30, which carry the highest informative value, but are not typically used in reference to the central event instance of day 0.

Recall that in order to implicate shared knowledge, one uses minimal or less marked expressions. Thus, if the typical frames of day 0 have become shared knowledge in day 8-30, then the writer optimally uses short and definite expressions to implicate them. Such expressions then evoke a strong typical frame, an *anchor* frame, that is sufficient to both refer to the event type and implicate the typical frames as shared knowledge. Such an anchor frame should show a high typicality score for the event type, but a low score across temporal distance classes, due to its frequent usage. KILLING and CATASTROPHE in Table 2 and Table 3 meet both requirements and refer to the main event instance. These might behave as anchor frames on day 8-30, backgrounding the main event instance and implicating the typical frames of day 0 as shared knowledge. This is demonstrated in (1b), where KILLING is evoked in the backgrounded noun phrase.

Finally, the results of the diagnostic classifier in Table 4 show that frame occurrences are more informative for the model than predicate occurrences. The above-chance performance of the model in the balanced/frames condition shows that it is capable to learn temporal patterns, just by paying attention

rank	music festival	gun violence
1	PERFORMING_ARTS (1)	ARREST (1)
2	SOCIAL_EVENT (.990)	LAW_ENFORCEMENT_AGENCY (.991)
3	CREATE_PHYSICAL_ARTWORK (.984)	WEAPON (.980)
4	PARTICIPATION (.975)	HIT_TARGET (.977)
5	ORIGIN (.967)	SHOOT_PROJECTILES (.952)
6	COMMERCE_SELL (.965)	KILLING (.946)
7	LOCALE_BY_EVENT (.965)	JUDGMENT_COMMUNICATION (.929)
8	EXPERTISE (.964)	SCRUTINY (.926)
9	COMPETITION (.964)	LOCATING (.919)
10	MANUFACTURING (.960)	CATASTROPHE (.919)
...	...	...
714	PEOPLE (.862)	CARDINAL_NUMBERS (.777)
715	LOCATIVE_RELATION (.840)	POLITICAL_LOCALES (.765)
716	CARDINAL_NUMBERS (.804)	LOCATIVE_RELATION (.763)
717	LEADERSHIP (.757)	LEADERSHIP (.629)
718	POLITICAL_LOCALES (.631)	PEOPLE (.601)
719	STATEMENT (.568)	STATEMENT (.040)
720	CALENDRIC_UNIT (0)	CALENDRIC_UNIT (0)

Table 2: The top 10 highest ranked frames (FFICF score) and the 7 bottom ranked frames for the event types music festival (Q868557) and gun violence (Q5618454). The scores were remodeled from (-1,1) to (0,1)

rank	day 0	day 8-30
1	STATE_OF_ENTITY (.007566) [D]	JUDICIAL_BODY (.007431) [N]
2	EXPERIENCE_BODILY_HARM (.006752) [Y]	DOCUMENTS (.007431) [N]
3	CAUSE_HARM (.006729) [Y]	JUDGMENT_COMMUNICATION (.006781) [N]
4	EVENT (.006607) [Y]	THEFT (.006538) [D]
5	MEDICAL_CONDITIONS (.006393) [Y]	INTOXICANTS (.006307) [N]
6	TAKING_TIME (.006317) [N]	BAIL_DECISION (.00623) [N]
7	SHOOT_PROJECTILES (.006266) [Y]	ORDINAL_NUMBERS (.006139) [N]
8	DIRECTION (.006037) [D]	CATEGORIZATION (.005915) [N]
9	RESPONSE (.006009) [N]	EVIDENCE (.005842) [N]
10	INFORMATION (.006006) [D]	UNATTRIBUTED_INFORMATION (.005827) [N]
...	...	...
710	KILLING (-.00196) [Y]	KILLING (-.00229) [Y]
711	VEHICLE (-.00299) [D]	VEHICLE (-.00302) [D]
712	LEADERSHIP (-.00422) [D]	CATASTROPHE (-.00421) [Y]
713	ROADWAYS (-.00552) [N]	LEADERSHIP (-.00421) [D]
714	CATASTROPHE (-.005763) [Y]	ROADWAYS (-.00457) [N]
715	AWARENESS (-.00763) [N]	AWARENESS (-.00656) [N]
716	BUILDINGS (-.0093) [Y]	BUILDINGS (-.0072) [Y]
717	LAW_ENFORCEMENT_AGENCY (-.01465) [Y]	LAW_ENFORCEMENT_AGENCY (-.01063) [Y]
718	PEOPLE (-.0379) [Y]	PEOPLE (-.03166)[Y]
719	CALENDRIC_UNIT (-.06463) [Y]	CALENDRIC_UNIT (-.05501) [Y]
720	STATEMENT (-.09892) [N]	STATEMENT (-.08964) [N]

Table 3: The top 10 highest ranked frames (FFICF score)[annotators' score: Y = yes, N = no, D = disagreement] and the 11 bottom ranked frames for the classes "day 0" and "day 8-30" within the event type gun violence. The scores range between -1 and 1.

	precision	recall	F1	support
1. predicates/unbalanced				
day_0	0.861	0.998	0.925	3896
day_8-30	0.357	0.008	0.016	630
Accuracy			0.860	4526
macro avg	0.609	0.503	0.470	4526
weighted avg	0.791	0.860	0.798	4526
2. frames/unbalanced				
day_0	0.891	0.974	0.931	3896
day_8-30	0.627	0.267	0.374	630
Accuracy			0.880	4526
macro avg	0.759	0.620	0.653	4526
weighted avg	0.855	0.876	0.854	4526
3. predicates/balanced				
day_0	0.562	0.676	0.614	630
day_8-30	0.594	0.473	0.527	630
Accuracy			0.575	1260
macro avg	0.578	0.575	0.570	1260
weighted avg	0.578	0.575	0.570	1260
4. frames/balanced				
day_0	0.746	0.789	0.767	630
day_8-30	0.776	0.732	0.753	630
Accuracy			0.760	1260
macro avg	0.761	0.760	0.760	1260
weighted avg	0.761	0.760	0.760	1260

Table 4: Classification reports providing, precision, recall, F1 and support for the performance of the Linear SVM on the test sets of four different experiments: 1. predicate frequencies/unbalanced corpus; 2. predicate frequencies/balanced corpus; 3. frame frequencies/unbalanced corpus; 4. frame frequencies/balanced corpus. Accuracy, macro average and weighted average are also provided per condition.

to frame occurrences.

We performed a model analysis to derive a ranking of the most important frames that the model used as margins to derive the hyperplane. The top 5 reads: TEMPORAL_SUBREGION, BECOMING_SILENT, SELF_MOTION, STORE and ENFORCING. None of these frames get a high typicality score in Table 3. Although the typical frames in the FFICF analysis show strong effects of foregrounding and backgrounding, idiosyncratic generic frames in the data seem more informative for the model in finding the most optimal separating hyperplane. TEMPORAL_SUBREGION might be a strong generic contender across event types due to its inherent temporal properties.[12] BECOMING_SILENT[13], SELF_MOTION[14] and ENFORCING[15] might show a significant frequency in a specific class in reference to the main event instance or subevents.

We assume that the results of our analysis can be generalized over unpredicted event types. From the onset, the common ground increases over time, affecting the pragmatic principles of foregrounding and backgrounding. Thus, if we would be able to obtain enough texts for the event type storm, we would expect the variation in framing between temporal classes to only occur with this event type as well. Since presidential election and music festival are rather anticipated events, the common ground at day 0 is at maximum height and build up from texts in preceding days. Thus, for these event types, temporal distance classes should be determined from preceding days up until the event itself.

6 Conclusion

In this paper, we measured variation in framing as a function of pragmatic foregrounding and backgrounding. We hypothesized that difference in common ground determine the extent to which the writer is able to background frames typically used in reference to the main event instance. We presented HDD, a corpus consisting of reference texts grouped under event types and enriched with publication dates. HDD was used to both perform FFICF between event types and between temporal distance classes, and train a diagnostic classifier. The former resulted in a ranking of typical frames per event type and between classes. The Linear SVM to a large extent was able to differentiate documents of different temporal distance classes. Frames turned out to be more informative than their predicates in training the model. Yet, The diagnostic classifier prefers idiosynchratic frames for learning the hyperplane.

In future work, we extend our experiments to more event types and we want to learn the specific frame-to-frame relations from the typical frames for event types. We expect to learn subevent relations from texts with short temporal distance and (causal) sequence relations from typical frames in texts with larger temporal distance.

Acknowledgments

This research was funded by the Dutch National Science Organisation (NWO) under the project number VC.GW17.083, Framing Situations in the Dutch language.

[12]Examples of lexical units: *later.a*, *earlier.a*, *early.a*.
[13]Examples of lexical units: *quiet.v*, *silence.v*
[14]Examples of lexical units: *walk.v*, *run.v*, *rush.v*
[15]Examples of lexical units: *enforcement.n*, *enforce.v*

References

Collin F. Baker, Charles J. Fillmore, and Beau Cronin. 2003. The Structure of the FrameNet Database.

International Journal of Lexicography, 16(3):281–296.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422, Uppsala, Sweden. Association for Computational Linguistics.

Prafulla Kumar Choubey, Kaushik Raju, and Ruihong Huang. 2018. Identifying the most dominant event in a news article by mining event coreference 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 340–345, New Orleans, Louisiana. Association for Computational Linguistics.

Herbert H Clark, S Haviland, and Roy O Freedle. 1977. Discourse production and comprehension.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Agata Cybulska and Piek Vossen. 2010. Event models for historical perspectives: Determining relations between high and low level events in text, based on the classification of time, location and participants. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Nan Decker. 1985. The use of syntactic clues in discourse processing. In *23rd Annual Meeting of the Association for Computational Linguistics*, pages 315–323, Chicago, Illinois, USA. Association for Computational Linguistics.

Marco Del Tredici and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. pages 1591–1603.

Elena Filatova and Vasileios Hatzivassiloglou. 2004a. Event-based extractive summarization. In *Text Summarization Branches Out*, pages 104–111, Barcelona, Spain. Association for Computational Linguistics.

Elena Filatova and Vasileios Hatzivassiloglou. 2004b. A formal model for information selection in multi-sentence text extraction. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 397–403, Geneva, Switzerland. COLING.

Charles J Fillmore and Collin Baker. 2010. A Frames Approach to Semantic Analysis. In *The Oxford handbook of linguistic analysis*. Oxford University Press.

Charles J Fillmore, Christopher R Johnson, and Miriam RL Petruck. 2003. Background to framenet. *International journal of lexicography*, 16(3):235–250.

H Paul Grice. 1975. Logic and conversation. *Cole, P., and Morgan, J.(Eds.)*, 3.

Joseph E Grimes. 2015. *The thread of discourse*, volume 207. Walter de Gruyter GmbH & Co KG.

Maarten Grootendorst. 2020. Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *15th International Symposium of Information Science (ISI 2017)*, pages 218–223.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Lawrence R Horn. 1998. Toward a new taxonomy for pragmatic inference: Q-based and r-based implicature. *Pragmatics: Critical Concepts*, 4:383–417.

Filip Ilievski, Marten Postma, and Piek Vossen. 2016. Semantic overfitting: what 'world' do we consider when evaluating disambiguation of text? In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1180–1191, Osaka, Japan. The COLING 2016 Organizing Committee.

Roderick Kay and Ruth Aylett. 1996. Transitivity and foregrounding in news articles: Experiments in information retrieval and automatic summarising. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 369–371, Santa Cruz, California, USA. Association for Computational Linguistics.

James Ko. 2018. Gun Violence Data. `https://www.kaggle.com/jameslko/gun-violence-data`.

Katja Markert, Yufang Hou, and Michael Strube. 2012. Collective classification for fine-grained information status. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 795–804, Jeju Island, Korea. Association for Computational Linguistics.

Michael W Morris and Gregory L Murphy. 1990. Converging operations on a basic level in event taxonomies. *Memory & Cognition*, 18(4):407–418.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56, Austin, Texas. 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.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas. Association for Computational Linguistics.

Marten Postma, Filip Ilievski, Piek Vossen, and Marieke van Erp. 2016. Moving away from semantic overfitting in disambiguation datasets. In *Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods*, pages 17–21, Austin, TX. Association for Computational Linguistics.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160, Portland, Oregon, USA. Association for Computational Linguistics.

Josef Ruppenhofer, Michael Ellsworth, Miriam R L Petruck, Christopher R Johnson, and Jan Schefczyk. 2010. FrameNet II: Extended theory and practice.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Ieva Staliūnaitė, Hannah Rohde, Bonnie Webber, and Annie Louis. 2018. Getting to "hearer-old": Charting referring expressions across time. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4350–4359, Brussels, Belgium. Association for Computational Linguistics.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv preprint arXiv:1706.09528*.

Shyam Upadhyay, Christos Christodoulopoulos, and Dan Roth. 2016. "making the news": Identifying noteworthy events in news articles. In *Proceedings of the Fourth Workshop on Events*, pages 1–7, San Diego, California. Association for Computational Linguistics.

Piek Vossen, Filip Ilievski, Marten Postma, Antske Fokkens, Gosse Minnema, and Levi Remijnse. 2020. Large-scale cross-lingual language resources for referencing and framing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3162–3171, Marseille, France. European Language Resources Association.

Piek Vossen, Filip Ilievski, Marten Postma, and Roxane Segers. 2018. Don't annotate, but validate: a data-to-text method for capturing event data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Etsuko Yoshida. 2011. *Referring expressions in English and Japanese: patterns of use in dialogue processing*, volume 208. John Benjamins Publishing.

Automatic Classification of Attributes in German Adjective-Noun Phrases

Neele Falk[†,‡*], **Yana Strakatova**[†*], **Eva Huber**[†,¶], **Erhard Hinrichs**[†]
[†]SfS, University of Tübingen / Tübingen, Germany
[‡]IMS, University of Stuttgart / Stuttgart, Germany
[†]firstname.lastname@uni-tuebingen.de
[‡]neele.falk@ims.uni-stuttgart.de
[¶]eva.huber@uzh.ch

Abstract

Adjectives such as *heavy* (as in *heavy rain*) and *windy* (as in *windy day*) provide possible values for the attributes `intensity` and `climate`, respectively. The attributes themselves are not overtly realized and are in this sense implicit. While these attributes can be easily inferred by humans, their automatic classification poses a challenging task for computational models. We present the following contributions: (1) We gain new insights into the attribute selection task for German. More specifically, we develop computational models for this task that are able to generalize to unseen data. Moreover, we show that classification accuracy depends, inter alia, on the degree of polysemy of the lexemes involved, on the generalization potential of the training data and on the degree of semantic transparency of the adjective-noun pairs in question. (2) We provide the first resource for computational and linguistic experiments with German adjective-noun pairs that can be used for attribute selection and related tasks. In order to safeguard against unwelcome memorization effects, we present an automatic data augmentation method based on a lexical resource that can increase the size of the training data to a large extent.

1 Introduction

There is ample evidence that humans decompose the meaning of objects and events into a set of prototypical semantic relations and their values. These relations, referred to in different frameworks as *attributes* (Barsalou, 1992), *frame elements* (Fillmore, 1982), *thematic relations* (Gruber, 1965), or *thematic roles* (Jackendoff, 1972), serve as an effective means to cluster classes of objects and events by degrees of semantic similarity. For example, thematic roles such as `buyer` and `seller` help distinguish among different participants in a financial transaction, and adjectives, such as *young* and *old*, group individuals into different equivalence classes for the relation `age`. Likewise, adjectives such as *heavy* (as in *heavy rain*) and *windy* (as in *windy day*) provide possible values for the attributes `intensity` and `climate`, respectively. The attributes themselves are not overtly realized and are in this sense implicit. While these attributes can be easily inferred by humans, their automatic classification poses a challenging task for computational models, as shown in the recent study by Shwartz and Dagan (2019) for English data. Compared to automatic role assignment for verbal arguments, attribute selection for adjective-noun pairs has received relatively little attention in computational semantics.

Attribute selection is highly relevant in different NLP tasks, such as information retrieval, topic modelling, and sentiment analysis. Consider a sentiment analysis task. If there is positive/negative sentiment expressed about something or someone, it is useful to know what triggers that sentiment. This requires from a system the ability to generalize over specific adjectives to more abstract attributes:

(1) *I* {*like/don't like*} *her siblings. They are*
   a. {*bright/stupid*} *people.*
      Attribute: `intelligence`
   b. {*friendly/rude*} *people.*
      Attribute: `behaviour`

For polysemous adjectives, the attribute selection task can be viewed as a coarse-grained word sense disambiguation. For instance, the adjective *bright* in example (1a) may acquire different meanings when it combines with different nouns, e.g. *bright room*, where the attribute is not `intelligence`, but `perception`.

In this paper, we frame the attribute selection task as a multiclass classification problem. We conduct experiments on the German dataset GerCo (Strakatova et al., 2020) of adjective-noun phrases. To the best of our knowledge, this is the first at-

[*] denotes equal contribution

Proceedings of the 14th International Conference on Computational Semantics, pages 239–249
June 17–18, 2021.

tribute analysis for German. Our main contributions are the following: (1) We gain new insights into the attribute selection task for German. More specifically, we develop computational models for this task that are able to generalize to unseen data. Moreover, we show that classification accuracy depends, inter alia, on the degree of polysemy of the lexemes involved, on the generalization potential of the training data and on the degree of semantic transparency of the adjective-noun pairs in question. (2) We provide the first resource for computational and linguistic experiments with German adjective-noun pairs that can be used for attribute selection and related tasks. In order to safeguard against unwelcome memorization effects, we present an automatic data augmentation method based on a lexical resource that can increase the size of the training data to a large extent.

This paper is structured as follows. We discuss related work in section 2. Section 3 describes the dataset in more detail. In section 4, we present the experiments and their results. Finally, we draw conclusions and give directions for future work in section 5.

2 Related work

Earlier studies of attribute selection focus primarily on English data. Hartung (2015) and Hartung et al. (2017) investigate the attributes in AN phrases and create a dataset for English adjective-noun phrases and their corresponding attributes based on the English WordNet. Hartung et al. (2017) try to model the task of selecting underlying attributes such as `age` for a phrase such as *old car* with representation learning: they experiment with different composition models to construct a single vector for the adjective-noun combination from the embeddings of the adjective and the noun. This composed vector is then used as a proxy for the underlying attribute, e.g. `age` and ranked with possible alternative values for other candidate attributes. Shwartz and Dagan (2019) evaluate different types of word embeddings on a number of lexical semantics tasks, including attribute selection and probe their ability to model lexical composition. For that purpose they reformulate the task of attribute selection into a binary classification: given an adjective-noun pair and an attribute, the classifiers predict whether the target attribute is selected for the pair in question. Their findings on the English dataset reveal that this task remains a challenge for all embedding types, though contextualized embeddings clearly outperform static embeddings.

Our work differs this from previous work in several aspects: we create the first dataset for the annotation of attributes in adjective-noun pairs for German. The taxonomy of 16 attributes is not as fine-grained as in Hartung (2015), who distinguishes between 254 attribute labels. Our more compact label set is thus more coarse-grained and more suitable for automatic modeling. We test the automatic models in a multiclass-classification setup with the adjective and noun embedding as input.

Unlike previous work on attribute selection, we take into account whether the semantics of an adjective-noun pair is transparent or not. Since the GerCo dataset contains both collocations and free phrases, we can partition the data accordingly and can compare the results obtained by a given classifier for the two classes. In earlier work (Strakatova et al., 2020), we report on binary classifiers for collocational and free adjective-noun pairs, which did not include prediction of the target attributes. In the present paper, the relevant attributes are taken into account. Therefore, our research contributes to a growing number of studies of semantic transparency, which up to now have focused on multiword expressions and nominal compounds (Reddy et al., 2011; Bell and Schäfer, 2013; Jana et al., 2019; Shwartz and Dagan, 2019) in particular, and extends this body of literature to the empirical domain of adjective-noun pairs. Our ability to distinguish between free phrases and collocations, allows us to test the finding of Espinosa Anke et al. (2019), who show that semantic relations in collocations are more difficult to predict in comparison to other types of relations such as hyponymy, meronymy, etc.

In sum, previous studies confirm that (i) revealing lexical relations in compounds and AN phrases is a challenge in NLP and (ii) relations found in collocations are more difficult to predict than other types of lexical relations. We combine these two findings in our study and model the lexical-semantic relations, which we call *attributes*, for both collocations and free phrases.

3 Data

In our experiments, we use the German dataset of adjective-noun phrases GerCo (Strakatova et al., 2020) which we annotate with additional seman-

tic information.[1] This dataset is suitable for our study due to several reasons: (1) it contains highly polysemous adjectives; (2) half of the dataset is represented by collocations; (3) it is based on a lexical resource – the German wordnet GermaNet (Hamp and Feldweg, 1997; Henrich and Hinrichs, 2010) which can assist us in augmenting the data and obtaining attribute information about it.

The original GerCo dataset contains 3,652 AN phrases manually annotated as "collocations" and "free phrases". The distinction between the two types is based on the transparency of the adjective in the phrase that is operationalized as literality (Reddy et al., 2011). For instance, in the phrase *grober Sand* 'coarse sand', the adjective has its literal sense of "rough in texture" – it is annotated as *free phrase*. In the phrase *grober Fehler* 'gross mistake', the meaning of the adjective is shifted: it does not describe texture in combination with the noun *Fehler* 'mistake', but refers to its intensity.

The adjectives in GerCo have been chosen on the basis of the semantic classes that they are assigned to in GermaNet. The advantage of GermaNet as a lexical resource is that, in contrast to the English WordNet, it models adjectives in a hierarchical manner similarly to nouns and verbs. From each of the 16 semantic classes for German adjectives, three adjectives have been selected. Each adjective is paired with the most frequent co-occurring nouns, thus all adjective-noun pairs in the dataset have a strong association.[2] In the present study, we excluded two relational adjectives from the data: *barock* 'baroque' and *steinig* 'stony'. Out of the remaining 46 adjectives, 44 have at least two senses (Strakatova et al., 2020). The top nodes of the GermaNet hierarchy of adjectives represent the 16 semantic classes and the direct hyponyms of the top nodes represent more fine-grained classes of adjectives.[3] Figure 1 shows a part of the taxonomy for one sense of adjectives *tief* 'deep' and *salzig* 'salty'. The top nodes are used as attribute labels to annotate the data (see section 3.1).

We make use of this hierarchical structure for adjectives in GermaNet in two ways: extracting attribute information (subsection 3.1) and automatic augmentation of the dataset (subsection 3.2).

[1]The dataset, the splits and the code for running the models on the data are available at https://github.com/Blubberli/IWCS-attributes.git

[2]Based on the logDice score (Rychly, 2008); 75% of the data has a logDice > 4.14.

[3]Based on the semantic classification of German adjectives proposed by Hundsnurscher and Splett (1982).

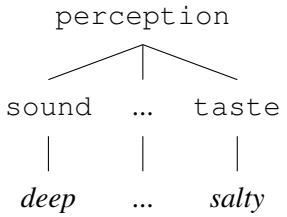


Figure 1: A part of the taxonomy of adjectives in GermaNet for *tief* 'deep' and *salzig* 'salty'. The top node is used as attribute label to annotate the GerCo dataset

3.1 Gold standard

For the present study, we add two layers of semantic annotation to the GerCo dataset: (1) by manual annotation: word sense IDs in GermaNet for all the adjectives and nouns in the dataset; (2) by automatic annotation: attributes for all the phrases.

Manual annotation. Manual annotation has been performed by two advanced students of computational linguistics with a solid background in lexical semantics and lexicography. Each adjective and noun from the GerCo dataset has been disambiguated and annotated with the corresponding sense IDs in GermaNet. We need these annotations for two reasons: to obtain attribute information about the phrases and to augment the data automatically.

Automatic annotation. To add the attribute annotations, we made use of the hierarchical structure of adjectives in GermaNet. Based on the manually annotated sense IDs of the adjectives, we assign an attribute label to each phrase automatically. For instance, *tief* 'deep/low' in *tiefe Stimme* 'deep voice' has been annotated with the sense "having a low pitch". The top node in the hierarchy for this sense is `perception` (see figure 1) – the phrase is assigned this label as an attribute. In *tiefe Liebe* 'deep love', the adjective is annotated with a different sense – "very strong, intense", the attribute label for this sense is `intensity`. Table 1 provides an overview of all the 16 labels with examples from the dataset (codenamed GerCo+).

Collocations. Half of the GerCo+ dataset is represented by collocations. Their distribution, however, is not balanced for each attribute. It concurs with the previous observations in literature that certain meanings tend to be expressed collocationally and certain meanings are usually found in free phrases. For instance, `intensity` is usually expressed in collocations whereas `color` in free

attribute	example
behaviour	*frecher Bursche* 'rude guy'
body	*blindes Kind* 'blind child'
climate	*windiger Tag* 'windy day'
evaluation	*herrliches Wetter* 'wonderful weather'
feeling	*bitteres Lachen* 'bitter laugh'
intensity	*leichter Regen* 'light rain'
location	*tiefer See* 'deep lake'
manner	*wilder Tanz* 'wild dance'
intelligence	*schlauer Junge* 'smart boy'
motion	*starres Gesicht* 'rigid face'
quantity	*karger Lohn* 'meager salary'
perception	*schwarzer Rock* 'black skirt'
relation	*sicherer Tod* 'certain death'
society	*reiche Verwandten* 'rich relatives'
substance	*grober Sand* 'coarse sand'
time	*alter Freund* 'old friend'

Table 1: Attributes in the GerCo+ dataset.

phrases (van der Wouden, 1997). Figure 2 shows the frequency distribution of collocations and free phrases in GerCo+. Four labels (`intensity`, `relation`, `manner`, `feeling`) are represented to a large extent by collocations, for `perception`, `substance`, on the other hand, the number of free phrases is very high. We expect collocations to be more challenging for the models.

Additional adjectives. The number of distinct adjectives in the original GerCo dataset is small. For some attributes (e.g. `evaluation`), very few adjectives are available. To be able to test each attribute for at least three distinct adjectives, we added 8 adjectives. We manually combined them with suitable nouns from the original dataset and annotated the phrases with the corresponding attributes. The adjectives in the final dataset can select between one and six different attributes (see figure 3). Most of the adjectives can select more than one attribute: this ambiguity is expected to pose another challenge for the automatic modelling.

3.2 Automatic augmentation

Lexical memorization is the tendency of a classifier to memorize the relations between words it has seen in training and corresponding labels (Levy et al., 2015). The generalisation ability of classifiers and the phenomenon of *lexical memorization* in classifying lexical inference relations and relations in noun compounds have been investigated by Levy et al. (2015); Dima (2016); Shwartz and Waterson (2018). Since the GerCo+ dataset is rather small, the danger of the classifier falling into the trap of lexical memorization effects needs to be safeguarded against. We therefore propose an automatic data augmentation to be able to create different training and test splits: either with modifier overlap, with head overlap or no overlap. We also expect a larger dataset to have positive effects on the precision of the machine-learning models. In order to increase the amount of training data, we perform automatic data augmentation relying on lexical and conceptual relations in GermaNet.

In GermaNet, senses of words are grouped into sets of synonyms (synsets). Synsets are connected to each other via conceptual relations, the main type of such relations is hyponymy/hypernymy as in *pie→pastry→baked goods*. Apart from that, some lexical units are interlinked via lexical relations, such as synonymy and antonymy. Attributes are expected to carry over to adjectives and nouns linked in GermaNet via lexical and conceptual relations. Knowing the sense IDs of all the words in the dataset, we therefore only have to extract the semantically related adjectives and nouns to generate new phrases. The new phrases are annotated automatically with the attribute from the original phrase. For instance, the original dataset contains the phrase *tiefer Ton* 'low-pitched sound' (collocation) with the attribute `perception`. Both words are provided with the corresponding sense IDs from GermaNet. The antonym of *tief* in this sense is *hoch* 'high-pitched' and a co-hyponym of *Ton* is *Pfeifen* 'whistle'. This results in a new phrase *hohes Pfeifen* 'high-pitched whistle' with the attribute `perception`.

Further phrases can be extracted via the adjectival top nodes in GermaNet: by combining non-ambiguous adjectives under those nodes with nouns that can select the corresponding attribute. Selecting only non-ambiguous adjectives, i.e. only adjectives that select a single possible attribute ensures that the resulting phrases is annotated with the correct attribute. For example, a new phrase for the attribute `perception` can be constructed by combining the adjective *salzig* 'salty' which can only express this attribute with other nouns that can have `perception`, e.g. *Suppe* 'soup'. We create two augmented datasets:

1. **small** Augment only the adjectives by adding synonyms, antonyms, direct hypernyms, all hyponyms and co-hyponyms

2. **large** Augment the adjectives and nouns by

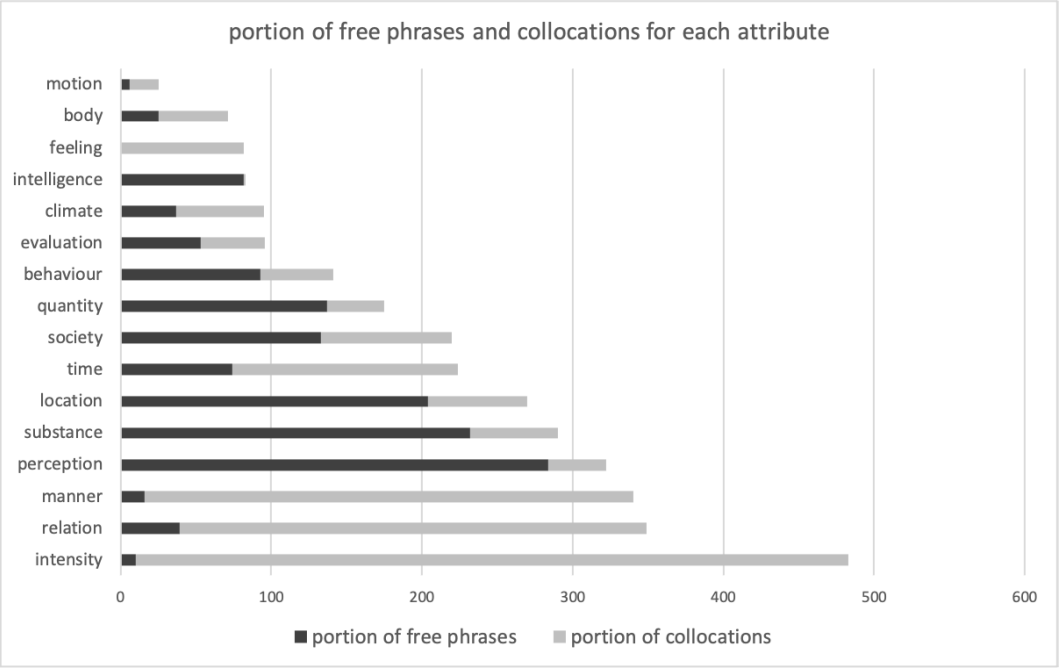


Figure 2: Distribution of free phrases and collocations in the GerCo+ dataset for each attribute.

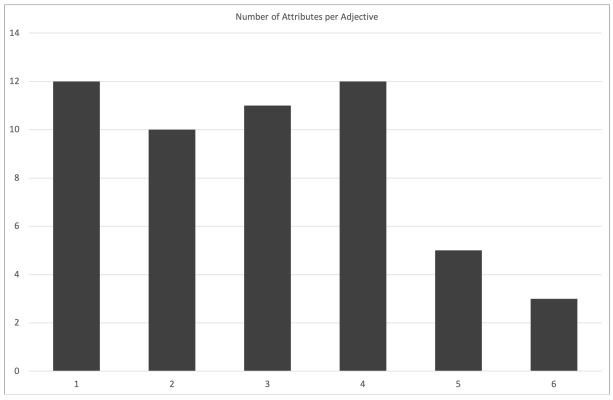


Figure 3: Distribution of the number of different attributes per adjective.

adding synonyms, antonyms, direct hypernyms, all hyponyms and co-hyponyms. Augment the attributes by combining all non-ambiguous hyponyms with suitable nouns.

In order to eliminate nonsensical phrases, the automatically created AN phrases are filtered by their bigram frequencies (>3) in a large corpus consisting of several German treebanks.[4]

Automatically augmented data is expected to be noisy to some extent. To estimate the amount of noise, we randomly extract 100 examples from each augmented dataset and manually assess the examples and the corresponding attributes. This study of random samples shows that around 20% of the automatically gained data is labeled incorrectly. Table 2 gives an overview of the data.

data	size	adj	nn	correct
gold standard	3,093	46	2,030	-
small	21,498	1,980	2,538	80%
large	232,389	4,630	36,659	79%

Table 2: Data overview: the amount of phrases, unique adjectives, unique nouns and the amount of correct phrases in the random sample extracted from each augmented dataset and evaluated manually.

3.3 Dataset splits

We create two test set ups: mixed and balanced. In the mixed setting, we test all the attributes and all the adjectives from the gold standard dataset. In the balanced setting, we use a subset of seven attributes with a balanced distribution of collocations and free phrases to compare the performance on the two types of phrases. The balanced attributes are `climate, quantity, time, society, location, behaviour, evaluation`.

The models are trained on the two automatically augmented datasets: small and large.

We create three splits of validation/test data from the gold standard GerCo+ dataset. Each test set contains roughly 700 phrases. To investigate the role of lexical memorization in the attribute selection task,

[4]TüBa-D/DP (de Kok and Pütz, 2019) and the corpus DECOW16AX (Schäfer, 2015; Schäfer and Bildhauer, 2012)

we create different lexical settings in the training data: (1) **No overlap** The validation/test and training have distinct vocabulary. (2) **Modifier overlap** The validation/test and training share modifiers (adjectives). (3) **Head overlap** The validation/test and training share heads (nouns).

4 Automatic classification

In the following experiment, we investigate to what extent attribute-selection can be computationally modeled. For that purpose, we use the data described in section 3.3 and train a simple neural network to predict one of the 16 possible attributes given the adjective and noun as input.

4.1 Modelling

We train a feed-forward non-linear classifier with one hidden layer. For each adjective-noun phrase, we extract the embedding for each constituent and apply a linear transformation to the concatenated input embeddings, followed by a `ReLU` non-linearity.

We experiment with two different embedding types:

- **fastText** (Bojanowski et al., 2017) non-contextualized German word embeddings with subwords trained on Common Crawl (Grave et al., 2018).
- **BERT** (Devlin et al., 2019) contextualized embeddings produced by a bidirectional transformer trained on Wikipedia, the EU Bookshop corpus, Open Subtitles, CommonCrawl, ParaCrawl and News Crawl.[5] We treat the adjective-noun phrase as the context sentence, thus the embedding of the adjective is only contextualized given the noun (and the other way around respectively).

The size of the hidden layer corresponds to the embedding dimension of one constituent (300 for fastText, 768 for BERT), the output layer has size 16 which corresponds to the number of different attributes.

We optimize the cross-entropy loss with Adam and use class weights, with higher weights for the less frequent attributes because the distribution of the attributes is imbalanced. As BERT comes with 12 layers, we learn a scalar-weighted combination of them. We always apply a dropout of 0.8. As the best model, we pick the one that achieves the best macro F1 score on the validation set after not improving for 5 epochs.

We use two baselines: We train each model with either using only the adjective or only the noun embedding as input. For the contextualized embeddings, we use the respective embedding after contextualization.

Note that our goal was not to find the best model for the task but to investigate how well a simple model can generalize for the task if it has been trained on a sufficient amount of data.

4.2 Results and Evaluation

(i) Generalization One of the research questions we want to answer with the experiment is in which way the automatic models can learn abstractions only on the basis of semantically related adjective-noun pairs. If the model has seen phrases like *black limousine* and *yellow truck* in training, is it able to learn the abstract attribute `perception` and predict correctly for test phrases, such as *red car*? In the best case, although the model has neither seen *red* nor *car* in the training set, it can arrive at the correct solution via lexical similarities: it has learned that colors express `perception` when combined with e.g. artifacts.

As mentioned in Section 3.2, , it has been shown for other tasks in lexical semantics that the abstraction ability of automatic models in supervised learning is diminished if constituents of the phrase in the test set have already occurred in training. It may then be easier for the model to memorize the most frequent or only class label for specific words to solve the task. We investigate to what extent that phenomenon applies to attribute selection. Especially for adjectives that occur with only one attribute, this effect would be expected. This phenomenon could have a particularly negative effect for ambiguous adjectives: In the worst case, lexical memorization overwrites the less frequent sense as only the most dominant attribute is predicted.

Table 3 shows the results for both embedding types for the different training data and the adjective and noun baseline. We report the average macro F1 score for all attributes, so each attribute is scored equally, regardless of the number of test instances.

First, it becomes clear that both models are capable of abstracting to some degree with fastText outperforming BERT by 6%. It is particularly interesting that there is hardly any difference between

[5] https://github.com/dbmdz/berts

the small and the large data set, although the large data set contains ten times more training instances. This demonstrates that it is not the size of the training data alone that matters for the generalization ability of the models. A sufficient lexical variety is much more important. This variety seems to be covered in the smaller training data set, such that an increase in size does not have a large effect on the general result. It is also evident that a partial overlap of adjectives and nouns leads to a significant improvement especially for BERT. This effect is similar on the smaller data set for modifier and head overlap, on the larger one a modifier overlap brings more advantages. The number of unique nouns is much higher in this data set, so it is less likely that lexical memorization can occur with the head overlap.

The results for the adjectives and noun baseline illustrate that while it is necessary to have both constituents as input for the models with fastText embeddings, the contextualization of the BERT embeddings is sufficient to convey almost the same information via one of the two contextualized vectors. In both cases the adjective baseline is stronger, indicating that the adjective plays a more important role for the task than the noun.

(ii) Attributes Figure 4 and Figure 5 show the performance for each attribute on the large dataset, for no overlap, modifier overlap and head overlap. The attributes `time`, `climate`, `perception` and `evaluation` can be learned particularly well without overlap. A possible explanation is that adjectives and nouns selecting these attributes have a high semantic similarity. For example, adjectives selecting `time` are more similar to each other than adjectives selecting `intensity`. For such attributes, the generalization is more difficult. For instance, `manner` and `intensity` are not easy to predict despite a high amount of training data (14,084 and 8,714 training instances). Attributes that benefit most from lexical overlap are `body`, `feeling`, `behavior`, and `motion`.

(iii) Polysemy With respect to lexical memorization, the findings here are mixed. While across-the-board improvements for each attribute with modifier or head overlap indicate that this phenomenon takes place, the partial overlap does not automatically lead to predicting the attribute for the polysemous adjectives that has the highest frequency in the training data. Table 4 depicts how many of all the possible attributes for the ambiguous adjectives in the test set are covered. We sum the number of correctly recognized attributes for each adjective. Out of the total of 144, roughly two thirds are recognized by the models for each setup, the number is even higher for the modifier overlap. For instance, in the case of the adjective *zart* 'tender', `substance`, `intensity` and `manner` were recognized without overlap, while `body` was additionally recognized with the modifier overlap. Table 5 shows the average accuracy for adjectives with different degrees of ambiguity regarding their possible attributes. A lower degree of ambiguity leads to better results. For a higher degree of ambiguity the modifier overlap brings significant improvements so the models can learn to better distinguish the different senses for the adjectives based on the training data. It is also worth noting that there is a considerable jump in accuracy when we compare adjectives that co-occur with four or more attributes with those that select at most three attributes.

training data	**fastText**			**BERT**		
			small			
	both	*adj*	*noun*	*both*	*adj*	*noun*
no overlap	0.50	*0.42*	*0.29*	0.44	*0.44*	*0,33*
modifier overlap	0.66	*0.45*	*0.38*	0.61	*0.61*	*0.49*
head overlap	0.67	*0.45*	*0.46*	0.61	*0.59*	*0.56*
			large			
no overlap	0.53	*0.45*	*0.24*	0.45	*0.41*	*0.38*
modifier overlap	0.68	*0.49*	*0.26*	0.71	*0.68*	*0.62*
head overlap	0.60	*0.47*	*0.31*	0.57	*0.53*	*0.52*

Table 3: Average Macro F1 Score over all attributes for each training set. The results are presented for training on the adjective and noun (both), and for the two baselines: trained only on adjectives (adj) and only on nouns (noun)

training set	**no overlap**	**modifier overlap**	**head overlap**
fastText	97	105	99
BERT	95	105	99

Table 4: Number of correctly predicted senses of polysemous adjectives for each embedding type and each training setup trained on the large dataset; the total number of different senses in the test data: 144.

(iv) Transparency To investigate the difference in the performance between collocations and free phrases, we use a smaller balanced test set (described in Section 3.3). Table 6 presents the results as the average of the Macro F1 scores of all 7 attributes in the test set.

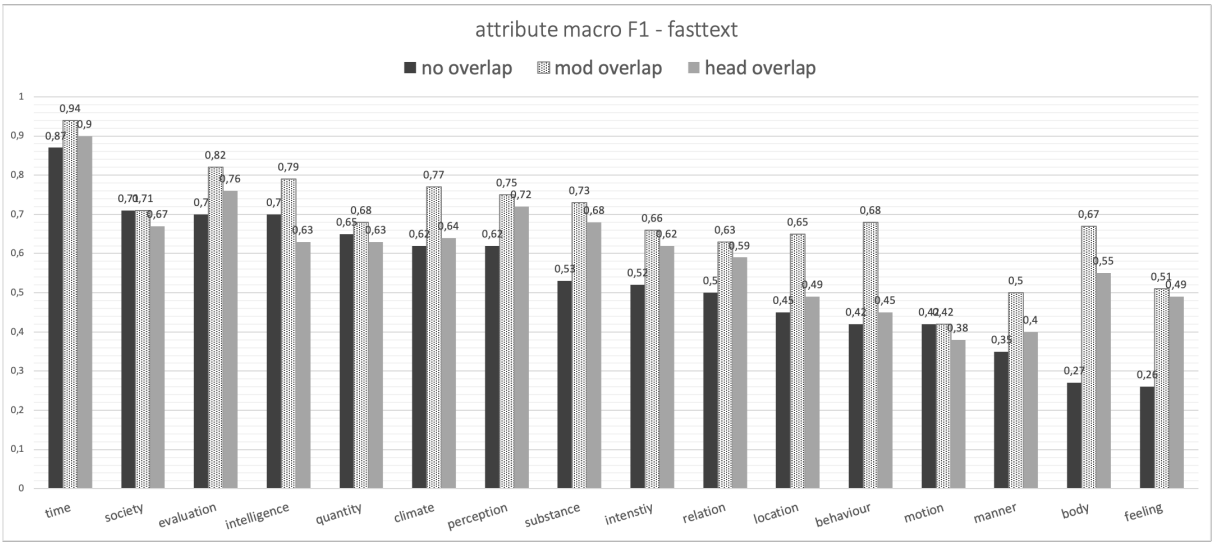


Figure 4: General Macro F1 for each attribute for fastText – each training set

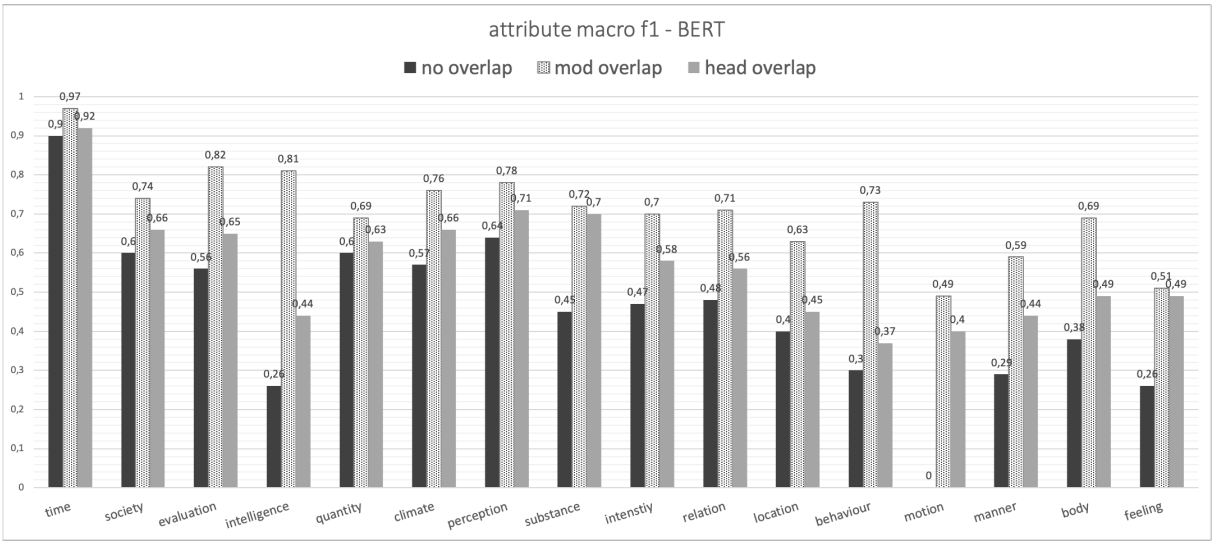


Figure 5: General Macro F1 for each attribute for BERT – each training set

no. attr	fastText			BERT		
	no	**mod.**	**head**	**no**	**mod.**	**head**
6	0.40	0.47	0.45	0.34	0.60	0.37
5	0.29	0.48	0.41	0.26	0.57	0.35
4	0.33	0.56	0.48	0.32	0.60	0.51
3	0.70	0.75	0.70	0.58	0.78	0.69
2	0.61	0.72	0.71	0.48	0.78	0.60
1	0.80	0.93	0.84	0.77	0.95	0.80

Table 5: Average accuracy for all adjectives with a specific number of possible attributes (no. attr) for the setup with no overlap (no), modifier overlap (mod) and head overlap (head).

Overall, there is a consistent difference between collocations and free phrases across all training data: free phrases are more accurately predicted in all cases. Contextualized embeddings were expected to yield better results for collocations because they are dynamically conditioned on the local context. Therefore, adjective and noun are represented by different vectors for different phrases. However, the model with BERT embeddings is worse if no lexical overlap is present. One reason for this may be that the contextualization of BERT does not give an advantage for a word-based task. It is more difficult to find regularities because the similarities between words could become blurred due to contextualization.

Although the performance for collocations is worse than for free phrases in general, for some attributes, the models are successful. This finding confirms the hypothesis that there are regularities also for collocations in spite of the general assump-

tion of their idiosyncrasy. For instance, the attribute `climate` has a high F1 score for collocations in all experimental settings (between 0.67 and 0.87). It indicates that meaning shifts of the adjectives selecting this attribute are regular. Another example of such a regular meaning shift is provided by the polysemous adjective *süß* 'sweet'. In its literal meaning, it refers to the attribute `perception` as in *süße Torte/Tee* 'sweet cake/tea'. However, *süß* can also refer to the attribute `evaluation` when it is combined for instance with nouns from the semantic field 'person', as in *süßes Kind* 'sweet child'.

By contrast, other collocations are highly lexicalized. These cases are hard to classify and remain a challenge. For instance, the models fail to predict the attribute `evaluation` for examples such as *helle Zukunft* 'bright future'.

training data	**fastText**		**BERT**	
	free phrase	collocation	free phrase	collocation
		small		
no overlap	0.66	0.53	0.59	0.44
modifier overlap	0.74	0.57	0.67	0.59
head overlap	0.80	0.73	0.73	0.67
		large		
no overlap	0.73	0.61	0.62	0.58
modifier overlap	0.84	0.73	0.87	0.72
head overlap	0.75	0.61	0.67	0.63

Table 6: Average Macro F1 score for the balanced set in terms of collocations and free phrases for each training set.

5 Conclusion and future work

In this paper we present a study on attribute selection in German adjective-noun phrases. Experiments in different training settings with and without lexical overlap show that it is possible to learn attribute selection patterns based on semantically related adjectives and nouns: abstract attributes such as `perception`, `time`, or `society` can be learned and predicted for new, unseen data.

The results of the experiments with different lexical overlap settings are in line with previous research: partial lexical overlap leads to better results on this task. However, this is not only due to lexical memorization. The models are still able to decide which attribute to select for an ambiguous adjective in the test set if it appears in training with all its possible meanings, based on the nouns combined with.

The experiments confirm that attributes are more difficult to predict for collocations than for free phrases. However, not all types of collocations are equally difficult. Attributes can be learned correctly for collocations when the meaning shift occurs systematically. Strongly lexicalized collocations cannot benefit from these regularities.

As future work it would be interesting to investigate attribute-selection in other languages, e.g., in Russian. Compounding in Russian is not as productive as in German and the function of compounds is often taken over by adjective-noun phrases, so a higher degree of lexicalization would be expected. This could result in an even greater difference between collocations and free phrases. Secondly, it would be interesting to investigate how using a full sentence as context impacts the results, especially in ambiguous cases. For instance, the phrase *stürmischer Tag* 'stormy day' can either express the attribute `climate` when the adjective is used in its literal sense or the attribute `manner` when *stormy* = *chaotic*. For such phrases, disambiguation is only possible in context. Finally, it would be useful if a model could learn a general intuition about whether a phrase is a collocation or a free phrase and which attributes are selected by an adjective in its literal and collocational senses.

Acknowledgments

We would like to thank our student assistants Daniela Rossmann, Alina Leippert and Mareile Winkler for their help with the annotations. We are also very grateful to the anonymous reviewers for their insightful and helpful comments that helped us to improve the paper. Financial support of the research reported here has been provided by the grant *Modellierung lexikalisch-semantischer Beziehungen von Kollokationen* awarded by the Deutsche Forschungsgemeinschaft (DFG).

References

Lawrence W. Barsalou. 1992. Frames, concepts, and conceptual fields. In *Frames, fields, and contrasts: New essays in semantic and lexical organization*, pages 21–74. Lawrence Erlbaum Associates, Inc.

Melanie J. Bell and Martin Schäfer. 2013. Semantic transparency: challenges for distributional semantics. In *Proceedings of the IWCS 2013 Workshop Towards a Formal Distributional Semantics*, pages 1–10, Potsdam, Germany. Association for Computational Linguistics.

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.

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.

Corina Dima. 2016. On the compositionality and semantic interpretation of English noun compounds. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 27–39, Berlin, Germany. Association for Computational Linguistics.

Luis Espinosa Anke, Steven Schockaert, and Leo Wanner. 2019. Collocation classification with unsupervised relation vectors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5765–5772, Florence, Italy. 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.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jeffrey S. Gruber. 1965. *Studies in Lexical Relations*. Ph.D. thesis, MIT. Distributed by: Indiana University Linguistics Club, Bloomington, Indiana.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a Lexical-Semantic Net for German. In *Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Matthias Hartung. 2015. *Distributional Semantic Models of Attribute Meaning in Adjectives and Nouns*. Ph.D. thesis, Heidelberg University, Germany.

Matthias Hartung, Fabian Kaupmann, Soufian Jebbara, and Philipp Cimiano. 2017. Learning compositionality functions on word embeddings for modelling attribute meaning in adjective-noun phrases. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 54–64, Valencia, Spain. Association for Computational Linguistics.

Verena Henrich and Erhard Hinrichs. 2010. GernEdiT - The GermaNet Editing Tool. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, pages 2228–2235.

Franz Hundsnurscher and Jochen Splett. 1982. *Semantik der Adjektive des Deutschen*. VS Verlag für Sozialwissenschaften.

Ray S. Jackendoff. 1972. *Semantic Interpretation in Generative Grammar*. MIT Press.

Abhik Jana, Dima Puzyrev, Alexander Panchenko, Pawan Goyal, Chris Biemann, and Animesh Mukherjee. 2019. On the compositionality prediction of noun phrases using poincaré embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3263–3274, Florence, Italy. Association for Computational Linguistics.

Daniël de Kok and Sebastian Pütz. 2019. *Stylebook for the Tübingen Treebank of Dependency-parsed German (TüBa-D/DP)*. Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, Germany.

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.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Pavel Rychly. 2008. A Lexicographer-Friendly Association Score. In *Sojka, Petr /Horák, Aleš (Hg.): Proceedings of the 2nd Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2008*, pages 6–9, Brno.

Roland Schäfer. 2015. Processing and Querying Large Web Corpora with the COW14 Architecture. In *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, pages 28–34, Lancaster, UK. UCREL, IDS.

Roland Schäfer and Felix Bildhauer. 2012. Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, Turkey. European Language Resources Association (ELRA).

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Vered Shwartz and Chris Waterson. 2018. Olive oil is made *of* olives, baby oil is made *for* babies: Interpreting noun compounds using paraphrases in a neural model. 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 218–224,

New Orleans, Louisiana. Association for Computational Linguistics.

Yana Strakatova, Neele Falk, Isabel Fuhrmann, Erhard Hinrichs, and Daniela Rossmann. 2020. All that glitters is not gold: A gold standard of adjective-noun collocations for German. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4368–4378, Marseille, France. European Language Resources Association.

Ton van der Wouden. 1997. *Negative Contexts. Collocation, polarity, and multiple negation*. Routledge.

Author Index