DMR 2023

The 4th International Workshop on Designing Meaning Representations

Proceedings of the Workshop

June 20 - 23, 2023 Nancy, France

Order copies of this and other ACL proceedings from:

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

ISBN: 978-1-959429-65-4

Preface

While deep learning methods have led to many breakthroughs in practical natural language applications, most notably in Machine Translation, Machine Reading, Question Answering, Recognizing Textual Entailment, and so on, there is still a sense among many NLP researchers that we have a long way to go before we can develop systems that can actually "understand" human language and explain the decisions they make. Indeed, "understanding" natural language entails many different human-like capabilities, and they include but are not limited to the ability to track entities in a text, understand the relations between these entities, track events and their participants, understand how events unfold in time, and distinguish events that have actually happened from events that are planned or intended, are uncertain, or did not happen at all. "Understanding" also entails human-like ability to perform qualitative and quantitative reasoning, possibly with knowledge acquired about the real world. We believe a critical step in achieving natural language understanding is to design meaning representations for text that have the necessary meaning "ingredients" that help us achieve these capabilities.

This workshop intends to bring together researchers who are producers and consumers of meaning representations and through their interaction gain a deeper understanding of the key elements of meaning representations that are the most valuable to the NLP community. The workshop will also provide an opportunity for meaning representation researchers to critically examine existing frameworks with the goal of using their findings to inform the design of next-generation meaning representations. A third goal of the workshop is to explore opportunities and identify challenges in the design and use of meaning representations in multilingual settings. A final goal of the workshop is to understand the relationship between distributed meaning representations trained on large data sets using network models and the symbolic meaning representations that are carefully designed and annotated by CL researchers and gain a deeper understanding of areas where each type of meaning representation is the most effective, and how they can be linked.

These proceedings include papers presented at the 4th Designing Meaning Representation workshop on June 20, 2023, held in conjunction with the15th International Conference on Computational Semantics (IWCS 2023) in Nancy, France. DMR4 received 20 submissions, out of which 13 papers have been accepted to be presented at the workshop as talks. The papers address topics ranging from meaning representation methodologies to issues in meaning representation parsing, to the adaptation of meaning representations to specific applications and domains, to cross-linguistic issues in meaning representation. In addition to oral paper presentations, DMR4 also featured invited talks by Alain Polguère (Université de Lorraine) and Juri Opitz (Heidelberg University), entitled "A graph approach to representing lexical semantics" and "Metrics of Graph-Based Meaning Representations with Applications from Parsing Evaluation to Explainable NLG Evaluation and Semantic Search", respectively.

We thank our organizing committee for its continuing organization of the DMR workshops, and the IWCS 2023 workshop chairs for their support. We are grateful to all of the authors for submitting their papers to the workshop and our program committee members for their dedication and their thoughtful reviews. Finally, we thank our invited speakers for making the workshop a uniquely valuable discussion of linguistic annotation research.

Workshop Chairs

Julia Bonn, University of Colorado Boulder
Nianwen Xue, Brandeis University

Organizing Committee

Omri Abend, Hebrew University of Jerusalem
Johan Bos, University of Groningen
William Croft, University of New Mexico
Jan Hajič, Charles University
Chu-Ren Huang, Hong Kong Polytechnic University
Stephan Oepen, University of Oslo
Alexis Palmer, University of Colorado Boulder
Martha Palmer, University of Colorado Boulder
James Pustejovsky, Brandeis University
Nathan Schneider, Georgetown University

Program Committee

Omri Abend, Hebrew University of Jerusalem
Daisuke Bekki, Ochanomizu University
Claire Bonial, US Army Research Lab
Alastair Butler, Hirosaki University, Japan
Ido Dagan, Bar-Ilan University, Israel
Lucia Donatelli, Saarland University
Katrin Erk, University of Texas, Austin
Anette Frank, University of Heidelberg
Kira Griffit, UPenn/LDC
Jan Hajič, Charles University
Daniel Hershcovich, University of Copenhagen
AiKaterini-Lida Kalouli, University of Konstanz
Paul Landes, University of Illinois at Chicago
Alex Lascarides, University of Edinburgh
Lori Levin, CMU
Bin Li, Nanjing Normal University, China
Jan Tore Lønning,, University of Oslo
Joakim Nivre, Uppsala Universitet
Tim O'Gorman, Thorn
Stephan Oepen, University of Oslo
Martha Palmer, University of Colorado
Lilja, Øvrelid, University of Oslo
Jacob Prange, Hong Kong Polytechnic University
Weiguang Qu, Nanjin Normal University
Nathan Schneider, Georgetown University
Maite Taboada, Simon Fraser University
Uresova, Zdenka, Charles University

Invited Speakers

Alain Polguère, University of Lorraine
Juri Opitz, Heidelberg University

Publicity Chairs

Kristine Stenzel, University of Colorado Boulder
Haibo Sun, Brandeis University

Table of Contents

Workshop Program

9:00–9:50 ***Invited Talk by Alain Polguère: A graph approach to representing lexical semantics***

9:50–10:10 *Structural and Global Features for Comparing Semantic Representation Formalisms*
Siyana Pavlova, Maxime Amblard and Bruno Guillaume

10:10–10:30 *Evaluation of Universal Semantic Representation (USR)*
Kirti Garg, Soma Paul, Sukhada Sukhada, Fatema Bawahir and Riya Kumari

10:30–11:00 **break**

11:00–11:20 *Comparing UMR and Cross-lingual Adaptations of AMR*
Shira Wein and Julia Bonn

11:20–11:40 *Abstract Meaning Representation for Grounded Human-Robot Communication*
Claire Bonial, Julie Foresta, Nicholas C. Fung, Cory J. Hayes, Philip Osteen, Jacob Arkin, Benned Hedegaard and Thomas Howard

11:40–11:55 *Annotating Situated Actions in Dialogue*
Christopher Tam, Richard Brutti, Kenneth Lai and James Pustejovsky

11:55–12:15 *From Sentence to Action: Splitting AMR Graphs for Recipe Instructions*
Katharina Stein, Lucia Donatelli and Alexander Koller

12:15–12:30 *Meaning Representation of English Prepositional Phrase Roles: SNACS Supersenses vs. Tectogrammatical Functors*
Wesley Scivetti and Nathan Schneider

No Day Set (continued)

12:30–14:00 lunch

14:00–14:50 Invited Talk by Juri Optiz: Metrics of Graph-Based Meaning Representations with Applications from Parsing Evaluation to Explainable NLG Evaluation and Semantic Search

14:50–15:10 *QA-Adj: Adding Adjectives to QA-based Semantics*
Leon Pesahov, Ayal Klein and Ido Dagan

15:10–15:30 *The long and the short of it: DRASTIC, a semantically annotated dataset containing sentences of more natural length*
Dag Haug, Jamie Yates Findlay and Ahmet Yildirim

15:30–16:00 break

16:00–16:20 *UMR Annotation of Multiword Expressions*
Julia Bonn, Andrew Cowell, Jan Hajič, Alexis Palmer, Martha Palmer, James Pustejovsky, Haibo Sun, Zdenka Uresova, Shira Wein, Nianwen Xue and Jin Zhao

16:20–16:40 *MR4AP: Meaning Representation for Application Purposes*
Bastien Giordano and Cédric Lopez

16:40–17:00 *Claim Extraction via Subgraph Matching over Modal and Syntactic Dependencies*
Benjamin Rozonoyer, David Zajic, Ilana Heintz and Michael Selvaggio

17:00–17:20 *Which Argumentative Aspects of Hate Speech in Social Media can be reliably identified?*
Damián Ariel Furman, Pablo Torres, José A. Rodríguez, Laura Alonso Alemany, Diego Letzen and Vanina Martínez

Structural and Global Features for Comparing Semantic Representation Formalisms

Siyana Pavlova, Maxime Amblard, Bruno Guillaume
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{firstname.lastname}@loria.fr

Abstract

The area of designing semantic/meaning representations is a dynamic one with new formalisms and extensions being proposed continuously. It may be challenging for users of semantic representations to select the relevant formalism for their purpose or for newcomers to the field to select the features they want to represent in a new formalism. In this paper, we propose a set of structural and global features to consider when designing formalisms, and against which formalisms can be compared. We also propose a sample comparison of a number of existing formalisms across the selected features, complemented by a more entailment-oriented comparison on the semantic phenomena of the FraCaS corpus.

1 Introduction

Over the past decades, various semantic representation formalisms have emerged, focusing on different features of semantics. New formalisms and extensions are continuously developed, highlighting a dynamic field, but few works have been carried out on their comparison. Abend and Rappoport (2017) provide a high-level summary of semantic features and existing formalisms. Žabokrtský et al. (2020) provide an overview and comparison of eleven deep-syntactic graph-based formalisms, focusing largely on their formal graph features. Insights into the difference between encoding some semantic phenomena in different formalisms can also be found in empirical work based on rule-based (Hershcovich et al., 2020; Pavlova et al., 2022) and machine learning (Kuznetsov and Gurevych, 2020; Wu et al., 2021; Prange, 2022) techniques.

Our goal is to provide a theoretical overview of various features of semantics and what choices are available for including them in the design of a new semantic representation formalism. The set of features can also serve for comparing different formalisms. In this spirit, we present some existing formalisms[1] and compare them against the outlined features. For a more entailment-balanced view and an empirical comparison, we also compare these formalisms against Cooper et al. (1994)'s FraCaS corpus. We focus on sentence-level semantics, but provide a short discussion on multi-sentence awareness for semantic representation formalisms.

The rest of the paper is organised as follows: in §2, §3 and §4 we present some global and structural features to be taken into consideration when comparing or designing a semantic representation formalism. In §5, we briefly present the following formalisms: Conceptual Graphs (CG) (Sowa, 1984), Montague Semantics (MS) (Montague, 1970; Montague et al., 1970; Montague, 1973), Discourse Representation Theory (DRT) (Kamp and Reyle, 1993), Minimal Recursion Semantics (MRS) (Copestake et al., 2005), Abstract Meaning Representation (AMR) (Banarescu et al., 2013), Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013), Universal Decompositional Semantics (UDS) (White et al., 2016), and Uniform Meaning Representation (UMR) (Van Gysel et al., 2021). In §6, we compare the formalisms across the selected features, and FraCaS features.

2 Pre-Semantics Issues

In this section, we outline a few aspects that we consider to not be constituent parts of what a semantic representation formalism (hereupon referred to as "formalism") is as such, but put it in a more global perspective and are nonetheless important to consider when designing one.

Scalability. Semantic representation formalisms vary in terms of complexity and expressive power.

[1]The list is not exhaustive, though we have attempted to cover a wide range of families.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 1–12
June, 2023.

While more complex ones may be more robust and encode a wider range of phenomena, complexity is negatively correlated to readability and therefore scalability. To be able to make use of various Machine Learning methods for parsing and generation, we need large amounts of manually (or at least semi-manually) annotated data. However, if the representation formalism is more complex, the skills required for annotation are more specialised. This makes the pool of potential annotators smaller and requires a longer and more in-depth training. A balance is necessary to ensure that a formalism covers a wide range of phenomena, yet it is not overly complex in order to keep the threshold for annotators relatively low. Alternatively, formalisms could propose lattices (Van Gysel et al., 2019)[2] for different phenomena, similarly to what UMR does. This would allow for a more coarse-grained annotation by less specialised annotators, and a more in-depth one by more specialised annotators.

This balance is also beneficial for analysis and comparison of different formalisms, as getting comfortable with reading and interpreting the representations is more straightforward.

Universality. The general intuition when talking about multi-linguality, is that meaning is preserved across languages. Thus, the semantic representation for a given text should be the same in all languages. In reality, many semantic representation formalisms are built upon the syntactic structure of sentences, which can differ greatly, especially between pairs of languages from distant families[3].

Furthermore, syntax-based semantic formalisms (and syntax-agnostic ones too) have historically been developed with well-resourced languages (mostly Indo-European, and in particular English) in mind. Thus, formalisms are likely skewed towards better representing phenomena that occur in those languages, and might even miss phenomena that do not appear in them.

Finally, similarly to scalability, using lattices could be beneficial for universality as the same semantic phenomenon may contain a more fine-grained set of categories in some languages than in others. Using lattices allows smoother annotation in different languages, while still keeping the possibility for cross-language comparison.

Unicity. Unicity addresses whether a formalism has a unique representation for a given meaning. In AMR inverting the direction of relations changes the focus and thus the meaning of the representation. On the other hand, if we take the formal logic representation of a sentence containing negation and conjunction, and apply De Morgan's Laws, we end up with a different, but logically equivalent representation. If equivalent representations are allowed for a certain formalism, it may be necessary to establish what constitutes the canonical form and how members of an equivalence class relate to it.

Flavor. We use the term "flavor" as used by Koller et al. (2019) in relation to the level of abstraction from surface form for graph-based semantic representations. Koller et al. (2019) define three levels: flavor 0 – a one to one correspondence between graph nodes and surface tokens; flavor 1 – all tokens are present as nodes, but there are additional nodes in the graph too; flavor 2 – not necessarily all tokens correspond to nodes, and there may be nodes that do not correspond to a specific token.

Use of lexical resources. Some formalisms rely on lexical resources for predicate and concept senses, or argument structure of predicates. This works well for languages where these resources already exist and are well-developed. However, for languages where this is not the case, there may be the need to produce them in parallel with producing annotations for the formalism, like the creators of UMR propose (Van Gysel et al., 2021). While viable, this makes the process longer and more complex and should be taken into consideration for the design. It is also tied to the *Universality* aspect: in order to enable cross-language comparison, for formalisms that do use lexical resources, there needs to be a link between said resources for multiple languages. While efforts exist in this direction (Bond and Foster, 2013; Bond et al., 2020), for most languages, this link is not there yet. Ideally, when creating datasets for a new language, the linking to other languages can be created in the process too. This, again, entails more effort, but we believe the cost of that is worth the benefits of having a more complete resource.

3 Semantic Features

In this section we discuss aspects of semantics that constitute what a semantic representation is.

Predicate-argument structure. The most

[2]E.g. *number* can be coarsely annotated as *singular* or *non-singular*, while the latter can be further broken down into *paucal* and *plural* where the distinction exists.

[3]If we assume that it is possible for two representations to have the same meaning, then the different representations stemming from different underlying syntactic structures in different languages would be less of an issue.

prominent feature of many formalisms is that they are centered around the predicate-argument structure of the events occurring in a sentence. Events are usually represented as predicates that take a certain number and kind of arguments. The relation between a predicate and an argument is expressed via a semantic role, which can be predicate specific (in the spirit of PropBank (Palmer et al., 2005)) or from a generic closed set (like VerbNet (Kipper et al., 2008)), with varying granularity.

Practical issues here arise from the fact that different formalisms use different lexical resources, making comparison and transformation more challenging. For English, work has been done to align (Palmer, 2009) and continue to improve the alignment (Stowe et al., 2021) of these resources. However, English is one of few languages where lexical resources are comparatively well developed. Thus, the use of language-specific frames for predicates comes with the cost of developing such resources. This is an argument against their use and for adopting methods that do not encode such senses, making the *Universality* point more easily attainable, similar to what UCCA does (Abend and Rappoport, 2013). That may, however, make the formalism less expressive.

Temporality. Temporal information deals with when an event occurs. We consider two aspects of this - when it occurrs relative to other events in the text, and when it occurred relative to the moment of speaking. Temporal information can be encoded in a variety of ways – via grammatical tense, from the lexicon with certain adverbs, or specific words or phrases, or may even be implicit. Combined with the fact that different languages have a stronger preference for some approaches over others, the task of encoding it is challenging. Formalisms need to decide whether temporal information will be encoded at all, and whether all kinds, that is, whether grammatical tense will be considered or only information present on the surface.

Aspect. Complementary to grammatical tense, grammatical aspect expresses how an event develops over time – whether it is one-time, whether it is continuous, whether it ended or is still ongoing. Here, again, formalisms have a choice – whether to encode aspect, and which features of it.

Spatial information. As Abend and Rappoport (2017) point out, spatial information in semantics is considered mainly for domains such as geographical information systems and robotics navigation. From a more theoretical perspective, we consider the resolution/interpretation of location-related deictics (*here*/*there*) and demonstrative pronouns to be an important aspect of the representation.

Encoding spatial information is especially relevant for sign languages, where its semantics is richer. For example, the handshape can express a distinction in an object's shape (e.g. curved or flat object) (Supalla, 1986), and the orientation of the handshape can express an object's orientation (Brozdowski et al., 2019).

Reification. Reification in semantics is the process of transforming events, actions and concepts so that they are expressed with (quantifiable) variables. This facilitates the translation of the so transformed representation into first-order logic and is therefore an important consideration if we want to use a formalism for logical inference.

Scope. The scope of semantic operators (such as those of quantification or negation) shows to which entities or events that operator applies. Some formalisms choose to not encode scope at all, making consistent logical inference impossible.

Scope does not directly relate to word order, which gives rise to *scope ambiguity* – a single sentence containing more than one scope operator can be interpreted in more than one way depending on how the operators combine. In case of scope ambiguity, the question for formalisms is whether to force a specific interpretation or to leave the representation underspecified. The latter allows that restrictions are added at a later stage when the correct interpretation becomes obvious from the context.

Negation. Negation, similarly to many of the other phenomena, can be expressed in different ways – overtly as a separate token, or as a morpheme of a token, presenting a challenge of whether to encode the two in the same way. We believe that meaning-wise, they should be equivalent and semantic representations should abstract away from the difference between the two. This is especially important when we consider logical inference and scope. Indeed, this is what many of the formalisms in section 5 do. There are some exceptions, notably UCCA, where, for example, the phrases "not clear" and "unclear" would be encoded differently despite having the same meaning.

Modality. Modality is used to express the reality of an event: realis – whether is it actually realised, or irrealis – whether it is a possibility or neces-

sity. Modal expressions are often expressed on the surface as modal auxiliaries, adverbs or adjectives. They get special treatment for most formalisms, be it as specially dedicated predicates (as in AMR), or operators between boxes (as in some realisations of DRT).

Modal expressions are also categorised in *flavors* (different from the flavor we discussed in section 2), showing how the possibility under discussion is linked to reality. *Epistemic* flavor covers possibilities based on some knowledge or belief, while *deontic* flavor expresses that the possibility is in accordance with what is required in reality.

Evidentiality is a phenomenon that encodes the type of evidence the speaker has for a statement. For example, one may differentiate between the speaker having direct (e.g. visual) or reportative (e.g. having heard about it and merely repeating what they have been told) evidence. In most languages this is expressed lexically with specific phrases (such as "reportedly" in English). However, in about a quarter of the world's languages, these differences are expressed grammatically (Aikhenvald, 2004), which formalisms do not address.

Logical inference. If we want to be able to use a semantic representation for reasoning, it is important that the formalism used permits logical inference. Not all formalisms are equally well equipped for this. For example, as Bos (2020) points out, with AMR, we are able to draw inferences, as long as there is no negation. That is, we can infer "it rained" from "it rained heavily", but we can also infer "it rained" from "it didn't rain". According to Bos (2020), this is due to negation in AMR being expressed as a predicate rather than an operator that takes scope. This highlights the importance of formalisms expressing scope-relevant phenomena (such a quantification and negation) in the appropriate way if we want to permit logical inference.

4 Semantics Interfaces

In this section we outline structural features of formalisms that are linked to their applications or to interfaces of semantics with syntax and pragmatics.

Generation and Analysis. When designing a new formalism, it is worth considering whether there are specific intended uses and applications for the formalism. Some tasks may rely more on parsing or on generation, so it is important to consider whether there are aspects that can be encoded into the design of the formalism to make parsing and/or generation more robust.

A lot of effort has gone into the parsing of text into various semantic representations as the amount of works on the topic suggests (Oepen et al., 2019, 2020). Challenges for parsing may come from the various types of ambiguities (e.g. lexical, scope) and, if we assume equivalent representations, which one to produce.

Similarly to parsing, generation from meaning representations has gathered much attention (Ribeiro et al., 2021; Hajdik et al., 2019). When keeping track of word order as part of the representation and without lemmatizing or otherwise modifying the original tokens, "generation" from semantic representation is straightforward for formalisms of flavor 1[4]. On the other hand, generation is a more interesting problem when working with flavor 2 formalisms, where the question is what to generate for a structure which may have more than one interpretation within the formalism.

Evaluation. For parsing, for most formalisms there are established methods for evaluating the produced representation against a gold one (Cai and Knight, 2013; Hershcovich et al., 2017; Oepen et al., 2014)[5]. Regardless, there are difficulties when using lexical resources and there is ongoing work on how to score closely related (but not perfectly overlapping) concepts in the representation (Opitz et al., 2020). Finally, if a formalism allows for multiple equivalent representations, similarity metrics will need to take this into account when evaluating a representation that is not in the canonical form for its equivalence class.

Evaluating generation is not straightforward when we consider that for some flavor 2 formalisms, many sentences have the same representation (e.g. AMR does not encode tense, so "I went to Paris" and "I will go to Paris" have the same representation). In such cases, it is necessary to consider whether it is enough to generate only one of the correct sentences in order to consider the process successful, or we need all the possible ones. Paraphrases pose a further issue, as they may have a (nearly) identical meaning to the original sentence, but look very different on the surface, with paraphrase evaluation being a subfield in its own right (Shen et al., 2022).

Compositionality. The meaning of a sentence

[4]Still, if the representation was automatically produced, the process may not be as direct.

[5]These metrics are also often used to compute inter-annotator agreement for manual annotation.

(or a phrase) is generally thought to be a function of the meanings of its composite parts. Historically, producing the semantic representation for a given sentence has passed through the syntactic one first, necessarily making compositionality a feature of the final representation. The Machine Learning revolution, however, has enabled the parsing of text directly into a semantic representation, rather than relying on the syntax-semantics interface, thus many of the newer formalisms have a choice to make about whether to preserve compositionality as a feature of the design.

A broader question is whether we consider the semantic representation to be only the final structure (e.g. graph or logical formula) that we obtain, or also the process of building that structure (in the spirit of derivation vs. derived trees for TAGs (Joshi et al., 1975)). If we take the latter view, then, necessarily, compositionality becomes a core aspect of the representation. We note, however, that this adds additional complexity to the annotation process, especially if the syntactic structure is not used as an underlying component.

Syntax-semantics interface. As mentioned above, semantic representation formalisms were built in such a way that the semantic structure of a sentence can be constructed from its syntactic one. Many of the newer formalisms are syntax-independent. While the first method may work for well-resourced languages with developed grammars, the latter one might be more beneficial for languages where these resources do not exist. This ties to the *Universality* point.

Multi-sentence. Many formalisms focus on representing sentences but do not necessarily employ means to go beyond the sentence boundary. The considerations we describe here can appear within a single sentence too, but are frequently seen when dealing with multiple sentences, namely anaphora and co-reference resolution, and the representation of discourse markers and relations.

When it comes to anaphora and co-reference, formalisms may choose to annotate the referents with the same variable, or with different ones. In the latter case, they may choose to employ a way to indicate that the variables refer to the same object or not do so. We note here the interesting case of AMR which includes a way, albeit somewhat superficial, to encode multiple sentences in the same representation. For referents occurring in the same sentence, AMR uses the same variable, but different ones when they occur in different sentences. Finally, similarly to scope ambiguity, formalisms need to take into consideration anaphoric ambiguity (in "John told Tom his brother left." it is ambiguous who "his" refers to) – whether to select one of the options, produce all different version, or leave the representation underspecified.

When treating discourse markers, formalisms have the choice to represent them in the same way as other relations, or give them a special status, thus adding a layer that sits on the boundary between semantics and pragmatics.

Questions. A distinction is usually made between Wh-questions and yes/no questions. For Wh-questions, a common approach is to maintain the structure of a declarative sentence and introduce a special concept or symbol (e.g. *amr-unknown* in AMR) to put in place of the entity or predicate that is being asked about. Yes/no questions usually need an additional relation to indicate that the whole statement is a question. It is interesting to note that in the case of DRS (at least in the version implemented in the Parallel Meaning Bank (Abzianidze et al., 2017), yes/no questions are ignored altogether and annotated in the same way as their declarative counterparts. This can be explained with the fact that DRT is designed to deal with discourse, as opposed to dialogue, and takes the stance that questions are only part of the latter.

5 Semantic formalisms

In this section, we describe existing formalisms and their core features, with a more exhaustive comparison in section 6. We strongly believe in the benefits of data-driven analysis and comparison. Therefore, if existent beyond a toy-corpus size, we also point to existing datasets.

Various extensions have been proposed for many of the formalisms. However, we do not know, for every extension, how it combines with the other ones and whether it does not interfere with the properties we explore. For example, adding scope interferes with compositionality. Thus, for this study we work with the original formalism, unless the extensions have been combined in a standalone one (as is the case with UMR).

Montague Semantics (MS) (Montague et al., 1970; Montague, 1970, 1973) introduced mathematical methods, namely higher-order predicate logic and lambda calculus, to semantics. Its core features are the use of model theoretic semantics,

and compositionality.

Conceptual Graphs (CG) (Sowa, 1984) are based on semantic networks and C.S. Peirce's existential graphs. Aside from natural language semantics, CGs have also been influential in knowledge representation. CGs' most apparent difference from modern semantic graphs is that they encode all events, entities, and relations as nodes, whereas edges are unmarked. Original CGs do not encode scope, but later versions provide that, along with ways to deal with temporal and modal logic (Sowa, 2003, 2006) and work has been done to combine CGs with generalized quantifiers (Cao, 2001).

Discourse Representation Theory (DRT) (Kamp and Reyle, 1993) is a "dynamic semantics" formalism, i.e. the meaning of a sentence is considered with respect to its potential to update context. It was designed to deal with anaphora and tense, but has since evolved to treat other semantic aspects such as presupposition, and propositional attitudes. DRT expressions are called Discourse Representation Structures (DRS). They are usually presented as nested boxes, but those can be transformed into graphs (Abzianidze et al., 2020). The Parallel Mearning Bank (PMB) (Abzianidze et al., 2017) is a large DRS corpus with gold annotations in English, German, Italian and Dutch.

Minimal Recursion Semantics (MRS) (Copestake et al., 2005) is a formalism from the Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994) family. As such, it has a strong link to syntax, but is also meant to be universal. MRS annotates a large range of phenomena, but is a rather complex formalism for annotators without linguistic knowledge. A distinguishing feature is its underspecifiability, which allows for encoding scope ambiguity. A medium-sized parallel dataset has been annotated for 15 languages[6]

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is meant to be a simple formalism to increase the ease of annotation. This is achieved by ignoring features such as tense, plurality and definiteness. AMR's core is the predicate-argument structure of events, with additional non-core roles specified for predicate-independent relations. For English, AMR relies on PropBank (Palmer et al., 2005) for predicate senses and semantic roles. A multitude of extensions have been proposed for AMR for various aspects such as tense (Donatelli et al., 2018), scope (Pustejovsky et al., 2019; Bos, 2020), spatial information (Bonn et al., 2020), multi-sentence information (O'Gorman et al., 2018). Despite its being designed with English in mind, AMR has also been used for Chinese, Czech, and Korean, among others. Larger corpora are available for English under a paid license, smaller ones are freely available[7].

Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013) is likewise designed to be simple for annotators. UCCA's *Foundational Layer (FL)* uses a set of 14 broad semantic role categories (e.g. `Participant`, `Adverbial`) and does not rely on lexical resources. The latter point makes it easier to adopt for multiple languages. Extension layers exist for UCCA that deal with semantic roles (Shalev et al., 2019; Prange et al., 2019a), co-reference (Prange et al., 2019b) and implicit arguments (Cui and Hershcovich, 2020). There are datasets for the FL for English, German, French, Hebrew and Russian.[8]

Universal Decompositional Semantics (UDS) (White et al., 2016) adds a number of semantic layers on top of the syntactic Universal Dependencies (UD)[9]. UDS follows the principle of decomposition, e.g. for semantic roles, they take Dowty (1991)'s view on decomposing notions such as `Agent` into finer properties like `volition` and `awareness`, allowing a single predicate to be assigned multiple of these categories. The currently existing layers address semantic roles; irrealis vs realis distinction on events; predicate senses and entity types; genericity; and duration and relative order of events. Annotated datasets are available for English[10].

Uniform Meaning Representation (UMR) (Van Gysel et al., 2021) is a proposal that extends AMR with aspect, temporal information, scope, co-reference and modal dependencies. UMR takes into account the morphosyntactic differences between languages and, to the best of our knowledge, is the first formalism to propose concrete steps on how to proceed with annotation for low-resource languages.

[6] https://github.com/delph-in/docs/wiki/MatrixMrsTestSuite

[7] https://amr.isi.edu/download.html

[8] https://github.com/UniversalConceptualCognitiveAnnotation

[9] https://universaldependencies.org/

[10] http://decomp.io/data/

6 Comparison and Discussion

In this section, we compare the formalisms from section 5 across the features outlined in §2, §3 and §4, as well as the phenomena covered by the FraCaS corpus (Cooper et al., 1994).

6.1 Feature Comparison

Table 1 provides an overview of how the frameworks compare across the features described in §2, §3 and §4 with the following exceptions: we add rows for dataset size and for the number of languages in which data is available, as these can be indicative of *Scalability* and *Universality* respectively; we add a row to show whether a formalism leaves the representation underspecified or not in case of *scope ambiguity*; we consider *Generation* and *Analysis* separately; we omit *Evaluation*, because while it is an important aspect to talk about, evaluation metrics are not formalism specific (e.g. Smatch (Cai and Knight, 2013) is typically associated with AMR, but can be used to evaluate any graph-based formalism).

Table 1 should be read as follows: for most features we indicate whether a formalism encodes it (✔) or not (✗). Dataset size is divided into three categories: toy (< 100 sentences for any language), medium (between 100 and $1,000$ sentences for at least one language), large ($> 1,000$ sentences for at least one language). For predicate-argument structure, we indicate whether semantic roles are predicate-specific or generic. For *Temporal* and *Evidentiality* we distinguish three categories: (0) not encoded with a dedicated structure / relation type; (1) encoded with a dedicated structure, but only if present on the surface, and not when grammatical; (2) encoded in all cases. For *Negation* and *Modality*, we distinguish three categories: (0) not encoded; (1) encoded, but without scope; (2) encoded with scope. For *Questions*, we distinguish between: (0) not encoded at all, (1) only wh-questions are encoded, or (2) all questions are encoded.

From the table, we can see that MRS is the most expressive formalism across the chosen features. However, this comes at the cost of it being complex to annotate, making it scale poorly. Similarly, MS and CG require some specialised knowledge for annotation and do not scale well, but while MS is close to MRS in terms of expressive power, CG lags behind. Original DRT, likewise, requires some specialised knowledge for annotation. However, recent work on simplifying the representation (Bos, 2021) and the existence of a large corpus (Abzianidze et al., 2017) lead us to consider DRT scalable. On the scalable side are also the newer formalisms, which have been designed for ease of annotation. However, for AMR, UCCA and UDS this means that they are not well-equipped to encode many of the semantic features we consider. For AMR and UCCA, extensions exist to address some of these issues. UDS, being a layered formalism, with new layers being added continuously, also has the potential to address the missing aspects. Finally, we take a look at UMR, which incorporates many of the proposed extensions of AMR, while preserving the latter's features. As we can see from the table, UMR is almost as expressive as MRS and DRT, while remaining syntax-independent, which its creators consider to be a strong point for scalability.

Looking across the features, we can notice that all formalisms can be used for *Generation* and *Analysis*. However, they all lack tools to deal with spatial information, especially the kind that is present in sign languages. Similarly, grammatically-expressed evidentiality is not annotated by any formalism. This opens a broader discussion regarding the encoding of features which are expressed only grammatically. We notice that surface information tends to be encoded, while for certain phenomena the grammatical side is ignored altogether. Thus, there is the risk of underrepresenting grammatical phenomena that are more prevalent in low-resource languages, but not in the well-resourced languages used as the basis for the design of formalisms.

6.2 FraCaS Comparison

FraCaS (Cooper et al., 1994) is a corpus of 346 textual inference problems, each consisting of one to five premises and a hypothesis. For each example, it is indicated whether it is true, false or unknown that the hypothesis follows from the premises. The problems are split into nine categories, relating to semantics (leftmost column of Table 2), however their distribution is not uniform. Some work on evaluating formalisms against FraCaS can be found in (Abzianidze, 2016; Haruta et al., 2019).

In Table 2, we provide a high-level comparison across the phenomena present in the FraCaS corpus. The table shows whether a formalism should be able to encode all (✔), at least half but not all (0.5), or less than half (✗) of the examples for a phenomenon. We want to highlight that this is

	MS	CG	DRT	MRS	AMR	UCCA	UDS	UMR
Scalability	✗	✗	✔*	✗	✔	✔	✔	✔†
Datasets (size)	toy	toy	large	medium	large	large	large	toy†
Universality	✔	✔	✔	✔	✗‡	✔	✔	✔†
Datasets (# languages)	-	-	4	> 10	> 6	6	1	-†
Unicity	✗	✗	✗	✗	✔	✔	✔	✗
Flavor	2	2	2	2	2	1	1	2
Lexical Resources	✗	✗	✔	✗	✔	✗	✗	✔
Pred-arg	generic	generic	generic	generic	specific	generic	generic	specific
Temporality	0	0	2	2	1	1	2¶	2
Aspect	✗	✗	✗	✔	✗	✗	✗	✔
Spatial	✗	✗	✗	✗	✗	✗	✗	✗
Reification	✔	✗	✔	✔	✔	✗	✗	✔
Scope	✔	✗	✔	✔	✗	✗	✗	✔
Scope ambiguity	✗	✗	✗	✔	✗	✗	✗	✗
Negation	2	1	2	2	1	1	1	2
Modality	2	1	2	2	1	1	1	2
Evidentiality	1	1	1	1	1	1	1	1
Logical Inference	✔	✗	✔	✔	✗	✗	✗	✔
Generation	✔	✔	✔	✔	✔	✔	✔	✔
Analysis	✔	✔	✔	✔	✔	✔	✔	✔
Compositionality	✔	✔	✗	✔	✔	✔	✔	✗
SSI	✔	✗	✔	✔	✗	✗	✔	✗
Multi-sentence	✗	✔	✔	✗	✗	✗	✗	✗
Questions	0	0	1	2	2	0	0	2

Table 1: Feature comparison. ✔ = yes, ✗ = no.
Temporal, *Evidentiality*: 1 = only surface, but not grammatical; 2 = yes.
Negation, *Modality*: 1 = encoded, but without scope; 2 = encoded with scope.
Questions: 0 = no special way to encode; 1 = only wh-questions encoded; 2 = all types of questions encoded.
* Original DRT requires some specialised knowledge, but given the recent proposal for simplification (Bos, 2021) and the existence of a large annotated corpus, we consider it scalable;
† UMR is designed with scalability and universality in mind, but it is a young formalism and both aspects remain to be verified;
‡ AMR does not claim to be universal, but corpora have been made available in a variety of languages.
¶ UDS encodes duration and relative occurrence of events, but does not specify when an event occured relative to the moment of utterance.

		MS	CG	DRT	MRS	AMR	UCCA	UDS	UMR
Quantifiers	23%	✔	✗	✔	✔	✗	✗	✗	0.5
Plurals	10%	✔	✗	✗	✔	✗	✗	✗	✗
Anaphora	8%	✗	✔	✔	✔	✗	✗	✗	✗
Ellipsis	16%	✗	✔	✔	✔	✗	✗	✗	✗
Adjectives	7%	✗	✗	✗	✗	✗	✗	✗	✗
Comparatives	9%	0.5	✗	0.5	0.5	✗	✗	✗	✗
Temporal	22%	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Verbs	2%	✗	✗	✗	0.5	✗	✗	0.5	0.5
Attitudes	4%	✔	✔	✗	✗	✗	✗	✔	✗
Total	100%	$\geq 52.5\%$	$\geq 39\%$	$\geq 62.5\%$	$\geq 73.5\%$	$\geq 11\%$	$\geq 11\%$	$\geq 16\%$	$\geq 23.5\%$

Table 2: Coarse-grained FraCaS comparison. ✔ = the formalism should be able to cover all examples for that feature; 0.5 = the formalism should be able to cover at least half, but not all examples for that features; ✗ = the formalism can cover less than half of the examples for that feature.

a very coarse-grained comparison[11] meant to balance the one in subsection 6.1 in providing a more entailment-based view and an empirical comparison of the formalisms. It should serve as a starting point for a more detailed, sentence-by-sentence comparison on this and other corpora.

We have made a few assumptions before our decision-making process. Where multi-sentence capabilities are relevant, namely for *Anaphora* and *Ellipsis*, we have taken the formalisms' ability to encode that into account. For categories where the focus is not on multi-sentence capabilities, we assume a conjunction of the premises to give a fairer chance to formalisms that only deal with single sentences. Table 2 is split in two parts, the lower one highlighting the sections where lexical information is necessary to resolve some of the examples. In such cases, we have taken the conservative view that formalisms are unable to encode the example. If we assume that with the help of the lexical resource the hypothesis can be deemed true, false or unknown, then the estimates for the lower part of the table would be higher. In what follows, we highlight the challenging areas in each category.

Quantifiers. For full coverage here, a formalism should be able to deal with scope, but also with definiteness, which is why while UMR covers scope, it cannot cover all examples in this section.

Plurals. While the majority of formalisms should be able to cover many of the *Conjoined Noun Phrases* examples, most will struggle with some of the bigger subsections, namely *Bare Plurals* and *Definite Plurals* due to inability to encode distinctions in definiteness.

Anaphora. While most formalisms can cover intra-sentential anaphora, for full coverage, they need to be able to also deal with inter-sentential one, which constitutes the larger part of this section.

Ellipsis. Similarly, if the ellipsis is in the same sentence, most formalisms perform well. However, since most examples in this section use multiple premises, only the formalisms that can deal with multiple sentences can get to full coverage.

Adjectives. Examples in this category rely heavily on lexical information (e.g. "former" implying that the phrase it is modifying is not necessarily up-to-date) and even some world knowledge (knowing that a "small elephant" is larger than a "large mouse"). Thus, none of the formalisms are equipped to deal with the majority of examples.

Comparatives. Two main difficulties arise here: similarly to *Adjectives*, lexical information is necessary for some examples, meaning none of the formalisms can reach full coverage. Furthermore, a large portion of the examples use quantification, making the formalisms that do not encode quantifiers well unable to cover even half of the examples.

Temporal. While some FraCaS temporal examples rely on tense or lexical semantics, for many there is temporal information present as separate surface tokens ("before", "for two years", "in 1991"). While most formalisms would be able to deal with these, many examples also include time spans which only UDS is explicitly equipped to encode. A few examples rely on lexical information as well ("started", "lasted", "was over" in example #259) which the formalisms will struggle with.

Verbs. For full coverage here, distinction between tenses, some lexical information, and capabilities to work with time spans are needed. Thus, none of the formalisms can encode all sentences.

Attitudes. To get a full coverage for this part, a formalism needs to either rely on lexical information (to distinguish between "managed to win" and "tried to win", for example) or employ specific ways to address epistemicity within its structure.

From Table 2, our general observation is that MRS, again, is the most expressive formalism, followed by DRT, while AMR, UCCA, UDS and UMR manage to fully encode only a few of the features. We remind the reader again that this is a very coarse-grained study. An in-depth sentence-by-sentence study is necessary to confirm our observations and provide an exact percentage of the FraCaS corpus by various formalisms.

7 Conclusion

In this paper we proposed a set of structural and global features to use when comparing semantic representation formalisms. We hope this set of features can be helpful for the community, both in the design of new formalisms and extensions, and in the selection of formalisms to use for specific tasks. The list of features is by no means complete, and extending it as well as the number of formalism can be the subject of future works. Similarly, we believe a more fine-grained study on the expressivity of formalisms with respect to the FraCaS corpus would be beneficial for the community.

[11] E.g. ✗ in the *Anaphora* row is different for UDS, which does not encode anaphora at all, and AMR, which encodes only intra-sentential anaphora, but still does not cover at least 50% of the *Anaphora* examples of FraCaS.

Acknowledgments

We would like to thank the anonymous reviewers for their feedback and comments. Part of this work has been funded by *Agence Nationale de la Recherche* (ANR, fr: National Research Agency), grant number ANR-20-THIA-0010-01.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–89, Vancouver, Canada. Association for Computational Linguistics.

Lasha Abzianidze. 2016. Natural solution to FraCaS entailment problems. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 64–74, Berlin, Germany. Association for Computational Linguistics.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze, Johan Bos, and Stephan Oepen. 2020. DRS at MRP 2020: Dressing up discourse representation structures as graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 23–32, Online. Association for Computational Linguistics.

Alexandra Y Aikhenvald. 2004. *Evidentiality*. OUP Oxford.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Francis Bond, Luis Morgado da Costa, Michael Wayne Goodman, John Philip McCrae, and Ahti Lohk. 2020. Some issues with building a multilingual Wordnet. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3189–3197, Marseille, France. European Language Resources Association.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

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.

Johan Bos. 2020. Separating argument structure from logical structure in AMR. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 13–20, Barcelona Spain (online). Association for Computational Linguistics.

Johan Bos. 2021. Variable-free discourse representation structures. *Semantics Archive*.

Chris Brozdowski, Kristen Secora, and Karen Emmorey. 2019. Assessing the comprehension of spatial perspectives in asl classifier constructions. *The Journal of Deaf Studies and Deaf Education*, 24(3):214–222.

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.

Tru H Cao. 2001. Generalized quantifiers and conceptual graphs. In *Conceptual Structures: Broadening the Base: 9th International Conference on Conceptual Structures, ICCS 2001 Stanford, CA, USA, July 30–August 3, 2001 Proceedings 9*, pages 87–100. Springer.

Robin Cooper, Richard Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspers, Hans Kamp, Manfred Pinkal, Massimo Poesio, Stephen Pulman, et al. 1994. Fracas–a framework for computational semantics. *Deliverable D6*.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3:281–332.

Ruixiang Cui and Daniel Hershcovich. 2020. Refining implicit argument annotation for UCCA. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 41–52, Barcelona Spain (online). 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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.

Valerie Hajdik, Jan Buys, Michael Wayne Goodman, and Emily M. Bender. 2019. Neural text generation from rich semantic 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 2259–2266, Minneapolis, Minnesota. Association for Computational Linguistics.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2019. A ccg-based compositional semantics and inference system for comparatives. *arXiv preprint arXiv:1910.00930*.

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.

Daniel Hershcovich, Nathan Schneider, Dotan Dvir, Jakob Prange, Miryam de Lhoneux, and Omri Abend. 2020. Comparison by conversion: Reverse-engineering UCCA from syntax and lexical semantics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2947–2966, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Aravind K Joshi, Leon S Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of computer and system sciences*, 10(1):136–163.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Dordrecht. Kluwer.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42:21–40.

Alexander Koller, Stephan Oepen, and Weiwei Sun. 2019. Graph-based meaning representations: Design and processing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 6–11, Florence, Italy. 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.

Richard Montague. 1970. English as a formal language.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In *Approaches to natural language: Proceedings of the 1970 Stanford workshop on grammar and semantics*, pages 221–242. Springer.

Richard Montague et al. 1970. Universal grammar. *1974*, pages 222–46.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics.

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.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Martha Palmer. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*, pages 9–15. GenLex-09, Pisa, Italy.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2022. How much of UCCA can be predicted from AMR? In *Proceedings of the 18th Joint ACL - ISO Workshop on Interoperable Semantic Annotation within LREC2022*, pages 110–117, Marseille, France. European Language Resources Association.

Carl Pollard and Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Jakob Prange. 2022. *Neuro-Symbolic Models for Constructing, Comparing, and Combining Syntactic and Semantic Representations*. Ph.D. thesis, Georgetown University.

Jakob Prange, Nathan Schneider, and Omri Abend. 2019a. Made for each other: Broad-coverage semantic structures meet preposition supersenses. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 174–185, Hong Kong, China. Association for Computational Linguistics.

Jakob Prange, Nathan Schneider, and Omri Abend. 2019b. Semantically constrained multilayer annotation: The case of coreference. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 164–176, Florence, Italy. Association for Computational Linguistics.

James Pustejovsky, Ken Lai, and Nianwen Xue. 2019. Modeling quantification and scope in Abstract Meaning Representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33, Florence, Italy. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.

Adi Shalev, Jena D. Hwang, Nathan Schneider, Vivek Srikumar, Omri Abend, and Ari Rappoport. 2019. Preparing SNACS for subjects and objects. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 141–147, Florence, Italy. Association for Computational Linguistics.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3178–3190.

John F Sowa. 1984. *Conceptual structures: information processing in mind and machine*. Addison-Wesley Longman Publishing Co., Inc.

John F Sowa. 2003. Laws, facts, and contexts: Foundations for multimodal reasoning. *Knowledge Contributors*, pages 145–184.

John F Sowa. 2006. Worlds, models and descriptions. *Studia Logica*, 84(2):323–360.

Kevin Stowe, Jenette Preciado, Kathryn Conger, Susan Windisch Brown, Ghazaleh Kazeminejad, James Gung, and Martha Palmer. 2021. SemLink 2.0: Chasing lexical resources. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 222–227, Groningen, The Netherlands (online). Association for Computational Linguistics.

Ted Supalla. 1986. The classifier system in american sign language. *Noun classes and categorization*, page 181.

Jens E. L. Van Gysel, Meagan Vigus, Pavlina Kalm, Sook-kyung Lee, Michael Regan, and William Croft. 2019. Cross-linguistic semantic annotation: Reconciling the language-specific and the universal. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 1–14, Florence, Italy. Association for Computational Linguistics.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3-4):343–360.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.

Zhaofeng Wu, Hao Peng, and Noah A. Smith. 2021. Infusing finetuning with semantic dependencies. *Transactions of the Association for Computational Linguistics*, 9:226–242.

Zdeněk Žabokrtský, Daniel Zeman, and Magda Ševčíková. 2020. Sentence meaning representations across languages: What can we learn from existing frameworks? *Computational Linguistics*, 46(3):605–665.

Evaluation of Universal Semantic Representation (USR)

Kirti Garg
IIIT Hyderabad
kirti.garg@gmail.com

Soma Paul
IIIT Hyderabad
soma@iiit.ac.in

Sukhada
IIT(BHU), Varanasi
sukhada.hss@iitbhu.ac.in

Riya Kumari
IIT(BHU), Varanasi
riyatomar912@gmail.com

Fatema Bawahir
IIIT Hyderabad
bawahir.fatema@gmail.com

Abstract

Universal Semantic Representation (USR) is designed as a language-independent information packaging system that captures information at three levels: (a) Lexico-conceptual, (b) Syntactico-Semantic, and (c) Discourse. Unlike other representations that mainly encode predicates and their argument structures, our proposed representation captures the speaker's vivakṣā - how the speaker views the activity. The idea of "speaker's vivakṣā" is inspired by Indian Grammatical Tradition. There can be some amount of idiosyncrasy of the speaker in the annotation since it is the speaker's viewpoint that has been captured in the annotation. Hence the evaluation metrics of such resources need to be also thought through from scratch. This paper presents an extensive evaluation procedure of this semantic representation from two perspectives (a) Inter-Annotator Agreement and (b) Utility for downstream task of multilingual Natural Language Generation. We also qualitatively evaluate the experience of natural language generation by manual parsing of USR, in order to understand the readability of USR. We have achieved above 80% Inter-Annotator Agreement for USR annotations and above 80% semantic similarity in multi-lingual generation tasks suggesting reliability of USR annotations and utility for multi-lingual generations. The qualitative evaluation also suggests high readability and hence utility of USR as a semantic representation.

1 Introduction

Semantic Representations (SemRep henceforth) generally encode predicate-argument structure of a verb (Propbank (Kingsbury and Palmer, 2002) and Palmer(OnlinePalmer et al., 2005), FrameNet (Baker et al., 1998) along with some other grammatical information ranging from lexico-syntactic level information such as tense-number-person (AMR (Banarescu et al., 2013), MRS (Copestake et al., 2005) to discourse level information such as topic-focus, co-referencing and discourse relations (PDT (Sgall et al., 1992) (Böhmová et al., 2003), UCCA (Abend and Rappoport, 2013). However, no semantic representation, that we are aware of attempts to capture what we term as the speaker's vivakṣā - how the speaker views the activity. We design a Universal Semantic Representation (USR) that encodes "speaker's vivakṣā". The idea is inspired from the Indian Grammatical Tradition (IGT henceforth). IGT views languages as a holistic phenomenon. Words are not derived as isolated units in Pāṇini's grammar, but as units that are semantically connected with other words in the sentence (Raster, 2015). Sentences are connected across the discourse. This is explicitly recognized by the Paninian rule (A 2.1.1) : *samarthaḥ padavidhiḥ*[1]. Keeping in tune with IGT, USR is designed as a representation that encodes information ranging from lexico-conceptual to discourse level in a connected structural format. Since this is a very new kind of representation, the evaluation of such a resource also requires special attention.

This paper presents the evaluation metrics of USR from two perspectives: (a) the Reliability of USRs (b) the utility of USR in the domain of multi-lingual generation. Sentences are generated in multiple languages to check the universality of information encoded in USRs. We use simple reliable measures to evaluate and understand these properties.

The **quantitative evaluation** metrics are presented from two perspectives: (a) the Reliability of USRs, (b) the utility of USR. The utility is evaluated for multilingual generation measured through Semantic Textual Similarity (STS). We use simple

[1]An operation on words [takes effect only] when the words are semantically connected.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 13–22
June, 2023.

reliable measures to evaluate and understand these properties.

The **qualitative evaluation** focuses on the usability of USRs in terms of readability of USR to generate natural language is examined. We also verify the adequacy of USR by manually generating natural language sentences from USRs.

A detailed analysis validates the proposed system as well as indicates areas of improvement. The feedback from these analyses is important for improving the information content and representation strategy of USRs.

Section 2 presents the design of USR. Section 3 studies Evaluation metrics in the context of other related works. Section 4 describes the quantitative evaluation metrics with results; while the qualitative measure is recorded in Section 5.

2 Design of USR

Unlike other representations that mainly encode predicates and their argument structures, the proposed representation captures the speaker's *vivakṣā*[2] - how the speaker views the activity. The idea of "speaker's *vivakṣā*" is inspired from Indian Grammatical Tradition (IGT henceforth). For example, how does the speaker's view differ when (s)he says 1 vis-a-vis 2? In Hindi, two different verb roots are used and the post-position on the seer also indicates different roles as shown in 1 and 2. In 1, Mira is an experiencer while in 2, the volitionality of Mira is maintained.

(1) *mīrā ko* *kala* *cāṃda* *dikhā*
Mira.exprncr yesterday moon see.int.pst

'Mira happened to see the moon yesterday'

(2) *mīrā ne* *kala* *cāṃda* *dekhā*
Mira.kartā yesterday moon see.tr.pst

'Mira saw the moon yesterday'

The activity of 'seeing' licenses[3] an animate *seer* and a *seen entity*. That is the *semantic frame* for the verb that every human being who knows the meaning of 'seeing' knows. But in communication, along with choosing the appropriate semantic frame, there occur two other important factors: (a) how the speaker conceptually cognizes the situation? (b) which linguistic expressions are available to translate that cognition into languages. For example, in the above examples, does the speaker want to express Mira's agency, or does (s)he want to foreground the appearance of the moon over the seer's agency? This is termed as the speaker's *vivakṣā*. Depending on that, the speaker would choose the best appropriate linguistic expressions to convey his/her thoughts. Our application task, namely Natural Language Generation (NLG) also motivates the requirement of capturing the speaker's *vivakṣā* in SemRep.

In order to generate a coherent and cohesive text, we require generative cues. Speaker's *vivakṣā* motivates those cues and we have decided to capture them in USRs through morphosemantics and dependency relations intra-sententially and also through discourse-level information.

USR encodes information at three basic levels: (a) Lexico-Conceptual (b) Syntactico-Semantic and (c) Discourse. This semantic information in USR is organized as features (in rows) and values, where the discourse relation and discourse co-referencing are accomplished through inter-USR linking which is established through Sentence_ID. Word_Index anchoring as shown in table 1. This representation is close to the Attribute Value matrix (AVM), but is easier to read and write manually, as well as process computationally.

Lexico-conceptual level: Conceptual Information which is generally expressed in terms of atomic words, multiword expressions or derived words are captured at this level. Currently, this level has information at 4 layers in USR. These layers (or rows) are (i) Concept row; (ii) Semantic Category; (iii) Morpho-semantic and (iv) speaker's view. Each entry to the concept row is an unambiguous representation of a concept. The ambiguity of a word is resolved in a very unique way in USR. Many SemReps use WordNet sense id as concepts. We propose to represent a concept in a multilingual set-up. For example, the lexeme in Hindi paḍha expresses two concepts: 'study' (as in *The boy studies in 7th standard*) and 'read' (*'the boy reads a book'*). This kind of ambiguity is handled at the conceptual level in the Concept Dictionary. This dictionary

[2] *śabdeṣvarthadānābhiprāyo vivakṣā* "vivakṣā is the intention of the speaker with regard to the meaning to be conveyed by the words" (Bhojaraja, 2007; Abhyankar, 1977). Abhyankar (1977) has also defined the term "vaktur-vivakṣā", in the same sense . As per "vivakṣātaḥ kārakāṇi" (Tripathi et al. 1986) kāraka roles in a sentence also apply according to the desire of the speaker.

[3] Selectional restriction of the verb which in IGT is known as a verb's *yogyatā*.

Concept Row	Sanskrit	Hindi	English	Bangla
paḍha_1	paṭha_1	paḍha_1	read_1	paṛa_1
paḍha_2	adhi+ī_1	paḍha_2	study_1	paṛa_2

Table 1: Concept Dictionary

has concept labels and equivalent concept labels in the languages under consideration. Currently, our lexicon has concepts in English, Hindi, Tamil and Bangla. The entry of a concept dictionary for the concept paḍha is the table 1.

USR has the Concept Label entry in the concept row which during generation selects concepts from the respective language cell depending on which language to be generated. In the current concept dictionary, there are 142037 labels for which Hindi and English concept labels are specified. For, 130948 concepts, Sanskrit labels are also attested in the dictionary. At the Lexico-conceptual level, the Semantic category row specifies the semantic category of a concept. Currently, four generic named entity categories are being annotated, namely- *per*(son), *org*(anisation), *place* and *other*. Apart from that, we mark *animacy* and *mass* categories.

Syntactico-Semantic level: Two types of relations capture information at this level: *kāraka* and *kāraketara* ('other than kāraka') (Kulkarni 2010) at the Dependency row. Pāṇini's system of knowledge representation is based on *kāraka* theory. There are six *kārakas* pointing out the relations between an event (or state) and its participants. They are *kartā*, *karma* (object), *karaṇa* (instrument), *sampradāna* (beneficiary), *apādāna* (source) and *adhikaraṇa* (time and location of action). *kāraketara* relations include relations between (a) noun and its modifiers; (b) verb and its verbal modifiers. There are a total of 42 dependency relations postulated till now in our work.
Discourse level: Language as a mode of communication always occurs as a discourse in which a sentence generally has a connection or trace with the previous and following sentence. Discourse relations map such inter-sentential information which forms a coherent text. Co-reference is another discourse strategy to indicate two entities within a sentence or across sentences having the same referent. In USR, all intra-sentential discourse information is encoded in the Dependency row and inter-sentential discourse information is captured in the Discourse row. Currently, we are representing a few inter-sentential discourse relations as described in Das (2016) following IGT. They are *pratibandha* (If...then), *samānkāla* (when...then), *kāraṇa-kāryabhāva* (although), *hetu-hetumadabhāva* (because), *asāphalaya* (but), *anantarkālinatva* (then). More relations are being identified and a contrastive study with RST and PDTB tagsets are also being carried out. At present, if no explicit relation across USRs is marked, the default relation 'and' is presumed.

2.1 Example of USRs

Table-2 and Table-3 present examples of USRs that generate the discourse given in the following discourse 3.

(3) a. śāma ko eka yā do camakate tāre najara āte haiṃ.
'One or two shining stars come to our notice in the evening'

b. lekina kucha hī samaya meṃ unakī saṃkhyā baḍha jātī hai.
'But, within a short time, their numbers increase.'

Every sentence is given a unique sentence id. The first and second sentences are related with *asāphalaya* relation which is marked on the verb of the second sentence as Sentence_ID.Word_Index:Relation_name.

USR is designed to facilitate language generation tasks. USR is a text-based data structure and hence can be parsed both by the machine as well as humans effectively. The Sentence type row records the type of the sentence. Concepts specified in the Concept Row along with information from Morpho-semantic row, Semantic Category row determine the correct word forms. Speaker's View row information is used to postulate discourse particles that convey the speaker's view. The TAM information on the verb determines verbal inflection. Information specified in Dependency, Construction and Discourse level determines syntagmatic relation among the words. Finally Agreement rules adjust the final word forms as and when necessary.

R(ow)2	Concept	Śāma_1 / evening_1	eka_1 / one_1	do_1 / two_1	camaka_1 / shine_1	tārā_1 / star_1	najara+ā_1-tā_hai_1 / appear_1-pres
R3	index	1	2	3	4	5	6
R4	Sem Category	time					
R5 R5	Morpho- semantics	[- sg a]				[- pl a]	
R6	Dependency	6:k7t	5:card	5:card	5:rvks	6:k1	0:main
R7	Discourse						
R8	Speaker's view						
R9	Sentence type	affirmative					
R10	Construction	disjunct:[2,3]					

Table 2: Sent-1: USR for Sentence: 3a. In the USR -k7t = temporal, card = cardinal, rvks = relation vartamān kāl samānādhikarana-(present simultaneous time), k1 = kartā (close to agent but not completely equivalent)

R2	Concept	kucha_1	samaya_1	tārā_1	saṃkhyā_1	baḍha_1- tā_hai_1
R3	index	1	2	3	4	5
R4	Sem Category					
R5	Morpho-semantics		[- sg a]		[- sg a]	
R6	Dependency	2:quant	5:k7t	4:r6	5:k1	0:main
R7	Discourse			Sent-1.5:coref		Sent-1.6:contrast
R8	Speaker's view	1:emph				[shade:completion]
R9	Sentence type	affirmative				

Table 3: Sent-2: USR for Sentence: 3b. In the USR - quant:quantity, r6 = genitive, emph= emphasis,Light verb jā (go) adds a sense of completion to the main verb

3 Related Works on Evaluation

Evaluation of Semantic Representations is a multi-dimensional task due to many qualitative parameters that need to be evaluated. Usual parameters of interest are the utility of the semRep, invariance, Universality (cross-lingual potential), usability, computational efficiency etc (Abend and Rappoport, 2017).

Human evaluation is one of the important methods for measuring the accuracy of generation tasks. A human evaluator can determine the accuracy, give a qualitative ranking based on the naturalness/fluency as well as completeness of information encoded in a given semantic representation. Several human evaluation based methods are in practice such as the WMT tasks (Bojar et al., 2016), Direct Assessment (Graham et al., 2017), HUME (Birch et al., 2016) for UCCA, HTER (Snover et al., 2006), or SMATCH (Cai and Knight, 2013) applicable to AMRs.

Human evaluations, besides being more accurate for SemRep evaluations, can also mark strengths and weaknesses of the generation, further indicating possible improvements. However, human evaluation would usually require skilled annotators as well as well-designed annotation guidelines to ensure objectivity. Hence, human evaluation is effective but can be resource and time-inefficient (Sai et al., 2020). Human evaluation reliability and consistency are measured through Inter-Annotator Agreement (IAA). Automated evaluations are the alternative to human evaluations, as they can be consistent, as well as resource efficient. However, the notion of semantic similarity is still not fully captured by the machine. Several word based, vector based and embeddings based measures are available for the same (Sai et al., 2020).

In this paper, we attempt to strike a balance between both human and automatic evaluation of USR and propose two kinds of evaluation: (a) Qualitative and (b) Quantitative. Table 4 summarizes our evaluation.

4 Quantitative Evaluation

This paper presents the quantitative evaluation metrics of USR from two perspectives: (a) the Reliability of USRs; (b) the utility of USR in the domain of multi-lingual generation.

The reliability is evaluated through Inter-Annotator Agreement. The utility of USR is evaluated by examining the textual similarity between the reference sentence and the manually generated

Type	Exp Name	Quality parameter	Dataset	Measure
Quantitative	IAA	Reliability	Geo_simple	Human Evaluation - Agreement %, Cohen's kappa
Quantitative	NLG utility	Correctness, completeness	Geo_6	Pairwise cosine with embeddings
Qualitative	Generation experience	Usability/ Readability	Geo_6 + verified_sentences	Human evaluation - effort, difficulty level

Table 4: USR Evaluation Framework

sentence. Essentially this becomes an evaluation of the generation task (Abend and Rappoport, 2017). Further, the generation task can be used to examine the utility of USR for multi-lingual generation, This is an important quality to evaluate as USR is designed to facilitate Natural Language Generation in multiple languages by using the multi-lingual concept dictionary to find equivalent concepts and can generate the same thought in multiple target languages.

We have extensively used the idea of semantic textual similarity (STS) in our evaluations, measured through human evaluation as well as by standard measures like pairwise cosine similarity. Here, we build a USR for a reference sentence R, then use that USR to either manually or automatically generate a sentence (G). If R and G are semantically close, we can say that the USR correctly and adequately captures the reference sentence meaning. Table 4 summarizes our evaluation framework.

4.1 Measuring Reliability of USR

This section describes the Reliability i.e. Inter annotator Agreement experiment.

4.1.1 Dataset

Geo_simple is a corpus of 90 simple sentences (with a total word count 928) created from the Indian NCERT Geography textbook for grade 6 and grade 7. The average length of these sentences is 11 words. These sentences are simple sentences, with one finite verb and zero or more non-finite verbs. Complex sentences are manually simplified to create simple sentences with proper connectives.

4.1.2 Experiment Setup

An annotation guideline document (USR Guidelines) is provided to two expert annotators with more than 6 months of experience with USR and its annotation. Geo_simple_0 is a set of base USRs automatically generated from sentences in Geo_simple dataset. Annotators independently develop their own versions of the USRs by editing the USRs in Geo_simple_0. Inter-annotator agreement (IAA) for different semantic features (the rows of the USRs) is calculated and then aggregated for the three levels of semantic information captured in USR.

For certain type of sentences, the annotators can differ in the number of concepts they identify. One case is the annotation of complex predicates. A complex predicate is a Noun+Verb construction. There can be disagreement among the annotators on when to call a Noun followed by Verb construction a complex predicate and when verb-object construction. Depending on that decision, the number of concepts identified for a given USR changes among annotators such that the concepts and their indices may differ partially, resulting in two very different looking, but valid USRs. To handle these kinds of situations, IAA is calculated for two different cases: a) Match cases - the number of concepts match (b) Not match - the number of cases differ. About 25% of our Geography data exhibits a difference in the number of concepts identified for the same reference sentence.

Inter Annotator Agreement (IAA) is measured using Agreement Percentage as well as Cohen's Kappa for Match cases (Cohen, 1960), but only Agreement Percentage (Given as Partial Agreement) for Not Match cases as Cohen's Kappa will be appropriate for such cases. IAA is interpreted using the agreement schema given by Landis and Koch (Landis and Koch, 1977) for sentences. The result is given in the next section.

4.1.3 IAA Results and Discussion

We have calculated the Inter Annotator agreement (IAA) separately for 'Match cases' and 'Not match cases'. The 'match' and 'Not match' cases for both

Type	Match cases		Non match Cases
Feature category	**Cohen's Kappa**	**Agreement %**	**Partial Agreement %**
lexico-conceptual	0.898	92.13	73.74
Syntactico-semantic	0.758	92.50	43.85
Discourse	0.869	95.52	77.78
Sentence type	0.929	95.588	76.00

Table 5: A summary of agreements for Match and Non match cases.

data are given in Table 5 .

Maximum impact of 'Not Match' concepts is seen at the syntactico-semantic level mainly for dependency attachments (Table 5) due to change in index numbers of concepts, as number of concepts is different. For Match cases, the Cohen's kappa scores for gender and number are comparatively low (0.76, in Table 5). A detailed analysis shows that the disagreement in the lexico-conceptual category are mainly seen in the semantic category and GNP information. The GNP information shows disagreement mostly for pronominal concepts. It can be attributed to the lack of context. For example, in the following case, Annotator1 chose to consistently not mark the gender for pronominal terms while annotator2 has decided otherwise. See the following example: 2nd person pronoun *tuma* (you)

original_sentence	Annotator1	Annotator2
maiṃ bhī jāūṃgā	[- sg u],	[m sg u],

Table 6: GNP annotation differences in USR annotation

can be both singular and plural in number. In such cases, annotators can overlook larger discourse information and tend to mark either singular(sg) or plural (pl) thus resulting in a disagreement in the annotation. Another low score in Table 5 is related to discourse relation. For this case, the agreement % is high while the Kappa score is comparatively low. Kappa is reducing the scores by assuming a probability of chance agreement, which itself has a low probability in our annotation exercise owing to the experience and expertise of our annotators. Hence, we feel that agreement % is a better measure of IAA for our annotations as compared to Cohen's Kappa. Results from the IAA experiment establish that the USR Guidelines is a reliable document and following that annotators with some training can reliably create USRs.

4.2 Measuring utility of USR for Multi-lingual generation

The utility of USR for multi-lingual generation is evaluated through a detailed experiment, where human generators manually parse the USRs to generate corresponding natural language sentences in Hindi, Bangla and Telugu by the aid of the multilingual concept dictionary. The underlying idea is as follows: If a generated sentence G (from USR U) and reference sentence R exhibit a high semantic textual similarity (STS), such that the USR U is created from R and is used to generate G, then it can be inferred that the semantic information captured by the USR is correct as well as adequate. The concept dictionary provides the corresponding concept in the desired output language. The generated sentences are evaluated manually and automatically for Semantic Textual Similarity.

4.2.1 Datasets

Geo_6 - The dataset consists of a corpus of 125 sentences from a Geography textbook of grade 6. These are simple sentences and do not contain any connectives. Complex sentences, if any are manually simplified to create simple sentences. The average length of these sentences is 11 words. Sentences from Geo_6 are used to programmatically generate a set of USRs (USR_0). The USRs are verified and edited by the experts for the correctness of content and structure (USR_1). USR_1 is used by a set of human generators to generate Hindi, Telugu and Bangla sentences.

Item	Score
Same meaning(Totally)	3
Minor difference in meaning	2
Not same at all	1

Table 7: Scoring Rubric for Human Evaluation of Semantic Textual Similarity

4.2.2 Multi-lingual generation Experiment Setup and Measures

All human generators, who are native speakers of their respective languages, are pre-trained to read USRs and decode the semantic information. The basic process for sentence generation in a target language is simple. For every reference sentence R the corresponding USR is made available to the human generator who manually parses the USR text structure. Human generators were asked to pay more attention to preserving information as it is (from USR) in the generated sentences and not to worry too much about maintaining the naturalness/fluency of the target language.

Once the human generators manually generate the sentences, a sanity checking is done in the following way before the automatic comparison with the reference sentences.

1. Reference Sentences without a corresponding generated sentence are excluded from further analysis.
2. Spelling mistakes are ignored.
3. Generated sentences with partially matching semantics are included in the response set, as they may indicate a deficiency in the USR.

For each sentence pair (Ri and Gi), we compute the Semantic textual Similarity (STS), manually as well by using known measures such as pairwise cosine measure after embedding sentences Ri and Gi using the state of art LaBSE model (Feng et al., 2020) as well as XLM-R (Conneau et al., 2020), a popular multilingual Masked Language Model (MLM). The embeddings are the vector representations of sentences such that the semantically similar sentences are closer, even if they belong to different languages, hence providing the cross-lingual measurement of similarity. The embeddings done using LaBSE provide reliable pairwise cosine measure (Feng et al., 2020).

Human evaluation of STS is done using the following scoring rubric (Table 7):

4.2.3 Results and Analysis

Hindi sentences are generated by two human generators. Hence we computed the internal consistency/reliability of human evaluation scores. The generations are internally consistent, and are acceptable as indicated by for human_generator1 (Cronback's Alpha score 0.76) and good for human_generator2 for (Cronback's Alpha score 0.82).

Next, we compute the frequency distribution of STS scores from human evaluation across the three target languages Hindi, Bangla and Telugu (Table 8).

Next, we compute the pairwise cosine similarity, with embedding, for the four sentence pairs namely Ref-Hindi1, Ref-Hindi2, Ref-Telugu and Ref-Bangla. (Table 9) records our results of both human and automated evaluation.

As evident from the high scores given by the human evaluators (Table 8, Table 9), and by both the reasonable cosine similarity scores, (Table 9), we can conclude that the semantics are preserved in the USR by a high degree of accuracy. The scores are also reliable as we can see a similar pattern in the scores gained from the above three methods. Since the Semantic Textual Similarity is reasonably high across the three languages, we can also confirm the universal nature of USR.

The Inter-Annotator Agreement scores make it evident that USR is a reliable semantic representation. Similarly, utility of USR for multi-lingual generation is high due to the ease of rules-based parsing of USR to construct a meaningful sentence.

5 Qualitative Evaluation

It is important to understand and record the experience of people involved in creating and using USRs. We are particularly interested in the readability of USR, because the idea is to create a gold standard USR bank which is only possible when human annotators can effortlessly read USR and correct it as needed. In this paper, readability is tested in terms of correctness and ease of generating sentences from a USR. If a human generator succeeds in generating a correct sentence with minor or no assistance, that shows that the USR is readable as well as adequate for correct sentence generation. We conducted a study and the following survey to check the readability of USRs by human beings. Human generators (14) with mixed prior knowledge and experience with USRs are given the manual generation task. The experience distribution of generators is as given in Table 10

Each human generator was first trained on generating a sentence from a given USR. USR guidelines were explained to them and they practiced on 3 USRs. Then each generator was given a set of 10 USRs from Geo_6 and another dataset to independently generate Hindi sentences. They could refer to the USR guidelines as many times as required.

STS Score	Hindi		Bangla		Telugu		**Total**
Score	Count	% within Hindi	Count	% within Bangla	Count	% within Telugu	
3 (Totally)	633	84.40	70	76.92	25	60.98	728
2 (partially)	92	12.27	20	21.98	13	31.71	125
1 (not at all)	25	3.33	1	1.10	3	7.32	29
Count Sentences	750(2 sets)	100	91	100	41	100	882

Table 8: Frequency Distribution of Semantic Similarity scores (Human Evaluation)

	Ref-Hindi1	**Ref-Hindi2**	**Ref-Telugu**	**Ref-Bangla**
sentence	91	91	41	93
Human evaluation (average) - (0- 3 rating)	2.81	2.85	2.71	2.33 0.778/
Pairwise cosine with LaBSE embeddings (0-1.0)	0.884	0.9041	0.746	0.604
Pairwise cosine with XLM-R (0-1.0)	0.916	0.938	0.738	0.705

Table 9: Semantic closeness scores for Multi-lingual generation from USR

	Academic Degree in Linguistics or language		Any other Degree	
	Experience < 3 months	Experience > 3 months	Experience < 3 months	Experience > 3 months
Count of human generators	2	5	5	2

Table 10: Experience distribution of Human Generators

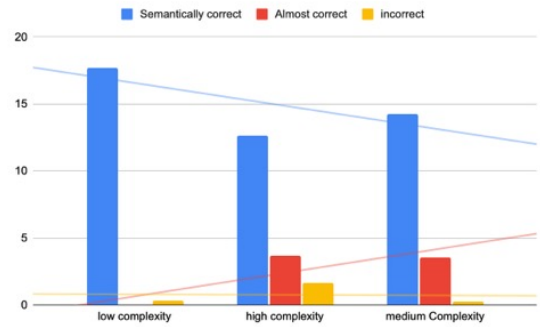


Figure 1: Generation correctness Vs. The complexity of the USR

The generators filled out a survey immediately after the Hindi generation task. The USRs were classified by the complexity level as low, medium and high, based on the number of concepts, and variations in dependencies, discourse, speaker's view information.

STS scores, measuring accuracy, for reference sentence and generated sentence were computed. A cross-sectional view of the correctness vs the complexity level is given in Figure 1. It is evident that generators could produce a high number of semantically correct (same meaning, and minor variations in meaning) sentences. The errors seen were mostly missing terms like 'almost, 'may-be', GNP and TAM (past vs present) variations. For example: For the reference sentence (Translated): *Sun is about 15 million KM away from the Earth.* Some generators did not include the word 'about'.

Figure 2 clearly indicates that the human generators could find the desired help in the USR guidelines. Most human generators found the USR Guidelines exhaustive and could use the document to clarify their doubts. The help was mostly sought for the dependency relations, as the list of dependencies is exhaustive, and remembering all can be an arduous task for a novice. Of the reported consultation of the USR guidelines, novice generators with < 3 months of exposure to USR required the most help as expected. The generators were also

		Difficulty Level					
		Very Easy	Easy	Ok	Difficult	Very Difficult	Total
My exposure to USR	<3 mth	0%	21.43&	14.29%	21.43%	0%	57.14%
	>3 mth	21.43%	7.14%	7.14%	7.14%	0%	42.86%
	Total	21.43%	28.57%	21.43%	28.57%	0%	100%

Table 11: Difficulty Level of USR Generation Process

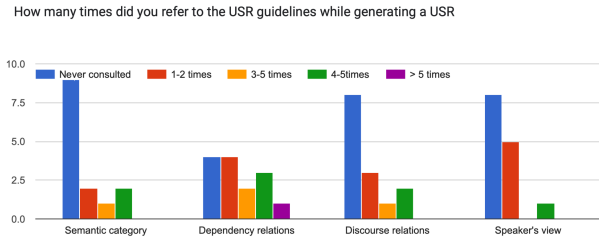


Figure 2: Frequency distribution of USR referrals while generating 10 USRs

asked to rate the difficulty of the generation process (Table 11). Majorly, the generators find the USR generation process to be very easy, easy, or OK (computed for both categories, <3 months exposure to USR; > 3 months exposure, using a Likert scale of 1-5, 5 being very difficult).

Based on the above experiences of the human generators, we can say with confidence that the readability of USRs is high as the generators could generate the USRs with ease, find the desired help in the guidelines, and could generate a high number of correct USRs. It is clear that the USR generation task is also not very difficult and gets easy with minor training. One important utility of USR readability measures is reflected in one of the tasks that we have taken up, namely training school children to read and write USR as an approach towards learning Universal Semantic Grammar through USR. The idea is that the USRs can enable children to overcome language barriers in communication.

6 Conclusion

In this paper, we have introduced a new SemRep called Universal Semantic Representation (USR). This is a very new initiative that attempts to capture the speaker's vivakṣā and is inspired from Indian Grammatical Tradition. The Lexico-Conceptual, Syntactico-Semantic and Discourse level information is encoded in a structured format in which USRs are interlinked to express the meaning of discourse as a whole. This paper presents the design of the USR and also records its detailed, multi-dimensional evaluation for reliability and its utility for natural language generation. Empirical evidence suggests high reliability as well as reliable semantic similarity scores for natural language generations done in multiple Indic languages namely Hindi, Bangla and Telugu. The qualitative evaluation strongly suggests that USR is easy to read and use with some training. Thus USRs are suitable for Natural Language Generation tasks, and can be used as a universal semantic representation.

7 Acknowledgement

We are thankful to every member of the Language Communicator team, specifically Arjun, Isma, Shweta, Bidisha and Hymavathi, for their contribution in data preparation and experiments. We are grateful to MEITY, Govt. Of India, for supporting and funding the Language Communicator project.

References

Omri Abend and Ari Rappoport. 2013. Ucca: A semantics-based grammatical annotation scheme. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 1–12.

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–89.

KV Abhyankar. 1977. A dictionary of sanskrit grammar, (1: 1961). *Baroda.(= Gaekwad's Oriental Series 134)*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation

for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Bhojaraja. 2007. *Shringaraprakasha*, volume 1. Motilal Banarsidass Publishers Pvt. Ltd. Delhi.

Alexandra Birch, Omri Abend, Ondřej Bojar, and Barry Haddow. 2016. HUME: Human UCCA-based evaluation of machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1264–1274, Austin, Texas. Association for Computational Linguistics.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The prague dependency treebank: A three-level annotation scenario. *Treebanks: building and using parsed corpora*, pages 103–127.

Ondrej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, and Lucia Specia. 2016. Ten years of wmt evaluation campaigns: Lessons learnt.

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 Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3:281–332.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Yvette Graham, Qingsong Ma, Timothy Baldwin, Qun Liu, Carla Parra, and Carolina Scarton. 2017. Improving evaluation of document-level machine translation quality estimation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 356–361, Valencia, Spain. Association for Computational Linguistics.

Paul R Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

M OnlinePalmer, D Gildea, and P Kingsbury. 2005. The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics Journal*, pages 31–1.

P Raster. 2015. *The Indian Grammatical Tradition*, volume 1. De Gruyter Mouton.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2020. A survey of evaluation metrics used for nlg systems.

Petr Sgall, Ján Horeckỳ, Alexandr Stich, and Jirí Hronek. 1992. Variation in language. *Variation in Language*, pages 1–381.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Srisa Chandra Vasu et al. 1891. *The Ashtadhyayi of Panini*, volume 2. Satyajnan Chaterji.

Comparing UMR and Cross-lingual Adaptations of AMR

Shira Wein
Georgetown University
sw1158@georgetown.edu

Julia Bonn
University of Colorado at Boulder
julia.bonn@colorado.edu

Abstract

Abstract Meaning Representation (AMR) is a popular semantic annotation schema that presents sentence meaning as a graph while abstracting away from syntax. It was originally designed for English, but has since been extended to a variety of non-English versions. These cross-lingual adaptations, to varying degrees, incorporate language-specific features necessary to effectively capture the semantics of the language being annotated. Uniform Meaning Representation (UMR) on the other hand, the multilingual extension of AMR, was designed specifically for uniform cross-lingual application. In this work, we discuss these two approaches to extending AMR beyond English. We describe both approaches, compare the information they capture for a case language (Spanish), and outline implications for future work.

1 Introduction

Abstract Meaning Representation (AMR; Banarescu et al., 2013) is a symbolic meaning representation which captures the meaning of a sentence in the form of a directed, rooted graph composed of predicate argument structures. AMR was originally designed for English, but has since been extended to many other languages. These cross-lingual adaptations of AMR vary in their approach to adapting English-centric AMR to other languages, which has posed a number of challenges.

In addition to language- or language family-specific (Heinecke and Shimorina, 2022) adaptations of AMR, Uniform Meaning Representation (UMR; Van Gysel et al., 2021a) is a recent multilingual extension of AMR which attempts to be generally cross-lingually portable.

Approach to cross-lingual adaptation has a significant impact on the utility of the annotated data. Formalisms which have similarly structured parallel annotations are better suited for incorporation into downstream applications, such as structure-aware machine translation systems (Sulem et al., 2015). Therefore, it is critical to understand the differences between UMR and cross-lingual adaptations of AMR, with regard to what linguistic information they encode, as it will impact the functionality of the annotations.

Sentence: *He denied any wrongdoing.*

AMR:

```
(d / deny-01
    :ARG0 (h / he)
    :ARG1 (w / wrong-02
        :mod (a / any)
        :ARG0 h))
```

UMR:

```
(s1d / deny-01
    :ARG0 (s1p / person
        :ref-person 3rd
        :ref-number Singular)
    :ARG1 (s1t/ thing
        :ARG1-of (s1d2/ do-02
            :ARG0 s1p
            :ARG1-of (s1w/ wrong-02)
            :MODPRED s1d))
    :ASPECT Performance
    :MODSTR FullAff)

(s1 / sentence
    :temporal ((DCT :before s1d)
                (s1d :before s1d2))
    :modal ((AUTH :FullAff s1p)
            (s1p :FullAff s1d)
            (s1d :Unsp s1d2))
    :coref (s0p :same-entity s1p))
```

Figure 1: AMR and UMR (from the guidelines, https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md) annotating the same sentence.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 23–33
June, 2023.

Though strong efforts have been made to adapt AMR to cross-lingual contexts in two directions (individual cross-lingual AMR extensions, and the more expansive UMR), there has not yet been any comparison between the effectiveness and comprehensiveness of these two different approaches.

In this work, we examine differences between these attempts at fashioning non-English-centric versions of AMR. In §2, we outline cross-lingual adaptations of AMR, including both annotation schema and generation/parsing tools, and survey select adaptations. Next (§3), we introduce UMR, the multilingual extension of AMR. In §4, we take a close look at how UMR and a cross-lingual adaptation of AMR handle linguistic features and examine cross-lingual challenges to UMR/AMR annotation. Finally, in §5, we discuss challenges for both UMR and cross-lingual extensions of AMR.

2 Cross-lingual Adaptations of AMR

AMR is designed to abstract away from the surface-form and syntactic nuance of the sentence, focusing only the basic meaning. In AMR annotations, nodes reflect concepts and the edges are labeled with relations between the concepts. Annotation of AMR concepts relies in part on PropBank lexicon of frame files[1] (Kingsbury and Palmer, 2002; Palmer et al., 2005; Pradhan et al., 2022), by annotating the frame associated with the token as the concept in the AMR graph.

Though AMR was designed exclusively for English and was not intended to be an interlingua (Banarescu et al., 2013), it has now been extended to multiple languages. Table 1 contains the cross-lingual AMR adaptations to date, with their publications as well as the underlying resources (frame files) they use and the corpus they annotate.

AMR has also been assessed as an interlingua for Czech (Urešová et al., 2014), Chinese (Xue et al., 2014; Wein et al., 2022b), and Spanish (Wein and Schneider, 2021). Xue et al. (2014) explores the adaptability of English AMR to Czech and Chinese. The authors suggest that, although it was not designed to be an interlingua, AMR may be cross-linguistically adaptable because it abstracts away from morphosyntactic differences. Cross-linguistic comparisons between English/Czech and English/Chinese AMR pairs indicate that most pairs align well, though there are some instances of divergence due to insertions, for example.

[1] https://github.com/propbank

Urešová et al. (2014) describes the types of differences between AMRs for parallel English and Czech sentences, and finds that the differences may be either due to convention/surface-level nuances which could be changed in the annotation guidelines, or may be due to inherent facets of the AMR annotation schema. One notable area of difference stems from the appearance of language-specific idioms and phrases.

Recent work has defined the types and causes of divergences between cross-lingual AMR pairs for English-Spanish parallel sentences. The causes of structural differences between parallel AMRs are identified as being due to semantic divergences, syntactic divergences, or annotation choices (Wein and Schneider, 2021).

In the subsections that follow, we consider four adaptations of AMR to individual languages.

2.1 Chinese AMR Adaptation

Li et al. (2016) suggested that AMR would be particularly well adapted to languages which vary morphosyntactically from English, because AMR abstracts away from the surface syntactic structure, motivating adaptation to Chinese. The Chinese AMR (CAMR) annotation schema largely matches that of the English annotation schema, with the concepts being tokens in Chinese instead of English. Notably, Chinese has very little inflectional morphology, so the AMR concepts more often directly correspond to tokens in the sentence than in English annotation. Extensions to the annotation guidelines are made for Chinese-specific constructions, including but not limited to (1) number and classifier construction, (2) serial-verb construction, (3) headless relative construction, (4) verb-complement construction, (5) split verb construction, and (6) reduplication. In the case where reduplication signals intensified meaning, Chinese AMR annotates this with another abstract concept, often with the role :UNIT. Discourse relations are also represented with concepts from the Chinese Discourse Treebank (DCTB; Zhou and Xue, 2015). These adaptations to the guidelines were identified during the annotation process.

2.2 Portuguese AMR Adaptations

Two distinct Portuguese AMR annotation schemata have been developed. Anchiêta and Pardo (2018) annotated the Portuguese translation of *The Little Prince*, and aligned the Portuguese sentences with the English ones (though there is one more sen-

Language	Underlying Resource(s)	Corpus	Publication
English	English PropBank	The Little Prince	Banarescu et al. (2013)
Chinese	Chinese Discourse Treebank	The Little Prince	Li et al. (2016)
Spanish	English PropBank	The Little Prince	Migueles-Abraira et al. (2018)
Spanish	AnCora	AMR 3.0 Data (news etc.)	Wein et al. (2022a)
Portuguese	FrameSet Verbo-Brasil	The Little Prince	Anchiêta and Pardo (2018)
Portuguese	FrameSet Verbo-Brasil	News, PropBank.Br	Sobrevilla Cabezudo and Pardo (2019)
Vietnamese	Vietnamese comp. lexicon	The Little Prince	Linh and Nguyen (2019)
Korean	Korean PropBank	ExoBrain	Choe et al. (2020)
Turkish	[Unspecified]	The Little Prince	Azin and Eryiğit (2019)
Turkish	Turkish PropBank	The Little Prince	Oral et al. (2022)
Persian	Perspred, English PropBank	The Little Prince	Takhshid et al. (2022)

Table 1: Comparison of characteristics of the AMR cross-lingual adaptations. "Underlying Resource(s)" for AMR reflect the lexicon or frameset used to mark roles and senses of concepts. "Corpus" indicates the corpus selected for annotation of the schema.

tence in the Portuguese corpus). This approach to Portuguese AMR annotation consists of importing the English AMR annotation for the aligned sentences, and changing the PropBank concepts to the equivalent Portuguese concepts from Frameset Verbo-Brasil (Sanches Duran and Aluísio, 2015). Any linguistic features that cause Portuguese AMR annotation to differ structurally from English AMR annotation were adjudicated upon at time of annotation for a given sentence. For example, instances of implied subjects and the particle "se".

A second Portuguese AMR annotation schema was developed shortly afterwards, which translates and fully adapts the English AMR guidelines to Portuguese. Duran and Aluísio (2011) annotated news texts from the *Folha de São Paulo* Brazilian news agency and from the *PropBank.Br* corpus. The verb senses are again determined by framesets from Verbo-Brasil. Modal verbs, which do not appear in Verbo-Brasil, are replaced by their direct Portuguese translations. Linguistic features handled specially in these new Portuguese AMR guidelines include use of the 3rd person singular and indeterminate subjects. Notably, multi-word expressions are replaced by their nearest one-word synonym.

2.3 Vietnamese AMR Adaptation

When adapting AMR to Vietnamese (Linh and Nguyen, 2019), the focus was on demonstrating relationships between entities and expanding annotation to include labels that mark function words, tense, and gender. Concepts were mapped from English to Vietnamese using the Vietnamese computational lexicon (Nguyen et al., 2006), with the addition of some new concepts. Linguistic differences between English and Vietnamese that trigger different annotation include morphosyntactic realization of manner as well as the presence of noun classifiers in Vietnamese. In English, manner is frequently expressed through *-ly* adverbs. In English AMR, *-ly* adverbs aren't included in graphs; rather, they are replaced by a related roleset or a related nominal or adjectival concept under a :MANNER relation (e.g. *quickly* in the surface form becomes :MANNER (q / quick) in the graph), Vietnamese expresses manner adjectivally, so such adjustments are unnecessary. In Vietnamese AMR, noun classifiers are omitted from the representation, except in cases where a noun classifier is alone (not directly preceding a noun phrase). Here, the co-referent needs to be included in the graph.

2.4 Korean AMR Adaptations

Choe et al. (2019) establishes a desire to make a Korean AMR annotation as similar as possible to AMR annotation in other languages so that cross-lingual annotations will be compatible and comparable, while at the same time bolstering the schema's ability to accurately reflect Korean semantics. The main areas in which special adaptations were needed include the copula and its negation, as well as case-stacking where multiple subjects or objects are involved.

Choe et al. (2020) further develops the annotation schema for Korean AMR and releases an annotated corpus for texts using Korean PropBank frames. Annotations were piloted on the ExoBrain Corpus, the Korean translation of The Little Prince, and example sentences for verbs in the Basic Korean Dictionary; the actual released corpus consists of annotations on the ExoBrain Corpus. The abstract rolesets used in English AMR (such as `have-org-role-91`) are also used for Korean AMR. For copular annotation, the use of `:domain` and `:polarity` are expanded.

3 UMR

The recent development of the Uniform Meaning Representation (Van Gysel et al., 2021a) aims to incorporate uniform treatments for linguistic diversity into the AMR annotation process.

Uniform Meaning Representation (UMR) is designed to extend AMR to a cross-linguistically viable meaning representation. Related work on BabelNet Meaning Representation (Navigli et al., 2022; Martínez Lorenzo et al., 2022) also extends AMR to a multilingual context, by moving away from English PropBank and instead using VerbAtlas (Di Fabio et al., 2019) for cross-lingual frames and BabelNet concept inventory (Navigli and Ponzetto, 2010).

To accommodate cross-linguistic diversity, UMR incorporates paradigmatic lattices to organize annotation categories from coarse-grained to more specific. Annotators are able to use the degree of granularity that is most suitable for the grammar of the language being annotated. Lattices produced for this purpose indicate degrees of granularity for discourse relations, modality, number, spatial relations, aspect, and temporality. The number of concepts associated with any given token (polysynthesis and agglutination) can also vary by language, so UMR does not require that morphologically complex words be broken down into separate morphemes when being annotated as concepts–however, it builds in the ability to do so where appropriate to support uniformity.

UMR extends AMR in 3 core ways: (1) it is capable of annotating low-resource languages, (2) it more comprehensively annotates modality, aspect, quantification, and scope for the benefit of logical inference, and (3) it annotates temporal, modal, and coreference relations across sentences.

At the sentence level, UMR adds aspect, modal strength, and quantifier scope attribute roles. Aspect is annotated for events and states at five base level values, with finer-grained values in lattice format (e.g., :ASPECT STATE). Sentence-level modal annotation comes in three strengths for both affirmative and negative (e.g., :MODSTR PRTAFF for partial-affirmative). The optional scope node augments predicates.

At the document level, UMR adds temporal and modal dependencies, plus coreference. Document-level semantic relations can be created for concepts/events within a sentence or across sentence boundaries. These document-level relations are able to be more fine-grained and provide more detailed information than their sentence-level counterparts, for instance, document-level modal relations are able to mark a conceiver in addition to the strength and polarity marked at the sentence level.

While UMR follows AMR in using existing roleset lexicons where possible (referred to as Stage 1 annotation), languages without these resources can also be annotated in UMR (Stage 0 annotation). During Stage 0 annotation, UMR-Writer (Zhao et al., 2021) allows annotators to select tokens for use as graph predicates and then add those predicates into a lexicon. Argument structures for these predicates are added using UMR's inventory of participant and non-participant roles. The predicates added to the working lexicon in combination with their participant role annotation information can be used to generate a roleset lexicon, moving a language from Stage 0 to Stage 1 annotation.

Recent work on UMR has produced small sets of annotations for four indigenous languages (Kukama, Arapaho, Sanapaná, and Navajo) (Van Gysel et al., 2021b), an online application (UMR-Writer) for producing AMR annotations (Zhao et al., 2021), automatically annotating tense and aspect in UMR (Chen et al., 2021), and incorporating non-verbal interactions into UMR annotation (Lai et al., 2021). Bonn et al. (2023) outlined deterministic conversion of AMRs to UMRs, specifically the roles, rolesets, and concepts.

4 Differences Between UMR and Cross-lingual AMR

In this section, we compare the specific linguistic features that both schemata encode, and consider two noteworthy obstacles/factors to successful annotation of UMR *or* AMR: idiomatic phrases and reliance on English concepts.

4.1 Comparison with Spanish AMR

In order to perform a language-specific comparison between a cross-lingual extension of AMR and UMR, we compare what Spanish AMR and UMR are able to capture for Spanish. We compare UMR with the Wein et al. (2022a) extension of AMR, which develops a corpus of approximately 500 sentences and guidelines for representing key linguistic features of Spanish in AMR. As depicted in Table 2, we find that most language-specific considerations in Spanish AMR are also included in UMR.

Verb Senses. Spanish AMR uses AnCora[2] verb senses, supplemented with specific senses which are not captured in the lexicon. Language-specific verb senses are used for UMR. In §5, we discuss the reliance on lexicons of both UMR and AMR.

Modality. Spanish AMR adds additional sense for *deber* (should) and *poder* (could) to mark modality. UMR marks modality through the sentence-level :MODSTR role.

Number for Persons. Spanish AMR opts against specifying number, while UMR has an additional modifying role for number of people/entities (`:ref-number`).

Pronoun Drop. Spanish AMR adds additional information for dropped pronouns by incorporating a `sinnombre` ("nameless") concept into the graph, e.g. `first-person-sing-sinnombre` for implicit entities. For example, the following AMR represents the Spanish sentence *Necesito irme* ("I need to leave"), with the first-person pronoun *"yo"* dropped.

(1) *Necesito irme.* 'I need to leave.'

AMR:
```
(n / necesitar-01
    :ARG0 (f / first-person-sing-sinnombre)
    :ARG1 (i / ir-05
        :ARG1 f)))
```
UMR:
```
(s2n / necesitar-01
    :ARG0 (s2p / person
        :ref-person 1st
        :ref-number Singular
    :ARG1 (s2i / ir-05
        :ARG1 s2p
        :ASPECT Performance
        :MODPRED s2n)
    :ASPECT State
    :MODSTR PrtAff))
```

UMR handles all pronouns–explicit, indexed, dropped, or implicit– via a generic concept (e.g., (p / person)) modified by `:ref-person` and *:ref-number*. There is no specific marking to indicate which of these methods of expression were used, however.

Politeness. Spanish AMR addresses politeness by adding a role relation for second person addressee. UMR adds a an attribute role :polite which follows the same pattern, as follows:

(2) *usted* 'you.FORM'

AMR:
```
(u / usted
    :mod-polite +)
```
UMR:
```
(s3p / person
    :refer-person 2nd
    :refer-number Singular
    :mod-polite +)
```

Affixes. Spanish AMR represents derivational suffixes as modifier concepts, and clitics are also treated as separate concepts.

How UMR handles derivational affixes depends on the type of affix and the annotation stage a language is undergoing. Languages undergoing stage 0 annotation (where there is no existing valency lexicon resource) may use an entire surface form (stem + affixes) as a graph predicate, or they may choose to systematically drop certain affixes as part of the lexicon-building process. Because the spirit of UMR (inherited from AMR) is to abstract away from syntactic manner of expression, lexical category-changing derivational affixes will likely be dropped from graph predicates by stage 1 annotation, with predicates coming from unified (part of speech-ambivalent) rolesets that will at that point have been created. Many other derivational affixes can now be dealt with through UMR graph structures (e.g., resemble-91 for similative affixes). But some will need to be resolved on a language-by-language basis as part of roleset development, as occurs in cross-lingual UMR.

UMR represents inflectional affixes via :ASPECT and :MODSTR attribute roles in the sentence-level annotation and the temporal and modal dependencies at the document level (as in figure 1). The affixes themselves may also be dropped from the graph predicate as deemed appropriate for a given stage of annotation for a language.

Examples of how AMR and UMR handle derivational suffixes and clitics can be seen in (3) and (4), respectively. In (3), the diminutive suffix /-ita/ is dropped from the head concept in the graph and represented via a :mod role in both Spanish AMR and UMR. Note that UMR doesn't have an abstract concept dedicated solely to the diminutive, and so the contents of the :mod relation will be unique to a given language, in whatever form the language deems most appropriate. The key is that the overall graph structure is the same cross-lingually.

[2] http://clic.ub.edu/corpus/en/ancoraverb_es

(3) *chiquita* 'little girl'
AMR/UMR:
(c / chica
:mod (p / pequeña))

(4) *mandarlo* 'send it'
AMR:
(m / mandar
:ARG1 (l / lo))
UMR:
(s4m / mandar :mode imperative
:ARG0 (s4p / person
:refer-person 2nd
:refer-number Singular)
:ARG1 (s4t / thing
:refer-person 3rd
:refer-number Singular)
:ASPECT Performance
:MODSTR PrtAff)

Double Negation. Double negation in Spanish can sometimes be used for emphasis, e.g. *No le dijo nada a nadie* ("She didn't say anything to anyone"). Spanish AMR specifies that double negation is treated the same as single negation (`:polarity -`). UMR guidelines do not state whether double negation receives special treatment, but one idea is to modify the polarity with :degree INTENSIFIER.

"Se" Usage. *Se* takes on many uses in Spanish. For AMR, there are three uses of note.

First, *se* can be used as a reflexive pronoun, annotated via reentrancy in English/Spanish AMR and UMR. For example, in 5, the reflexive verb *mirarse* (look at oneself) forces a reentrancy for *se* in both the Spanish AMR and the UMR.

(5) *él se miraba en el espejo* 'he looked at himself in the mirror'
AMR:
(m / mirar-01
:ARG0 (e / él)
:ARG1 e
:location (s / espejo))
UMR:
(s5m / mirar-01
:ARG0 (s5e / él)
:ARG1 s5e
:location (s5e2 / espejo)
:Aspect Activity
:MODSTR FullAff)

Second, *se* can reflect a passive marker / an omitted concept (e.g. *se vende*, for sale). In this case, Spanish AMR uses the token *se* as the argument role label. UMR would annotate these passive markers as appropriate for the language and has guidelines specifically for passives. Third, *se* can be used as an impersonal pronoun (e.g. *no se debe fumar*, one should not smoke). Given that *se* is a pronoun, the second and third uses of *se* are handled in UMR using the `:ref-persons` concept.

Document-level representation, Scope, and Aspect. UMR expands AMR by adding annotation guidelines for document-level representation, scope, and aspect, while Spanish AMR has none of the three.

4.2 Encoding Specific Linguistic Features for Other Languages

For languages which have less syntactic similarity to English than Spanish does, some language-specific features that could be accommodated by a custom monolingual AMR-adaptation may be more straightforward to handle in UMR than others. For example, numeral noun classifiers in Vietnamese are easily covered in UMR with the numeral lattice. In Korean, UMR's flexibility towards representing affixes as concepts allows handling of case-stacking. On the other hand, specifics such as reduplicatives (in Mandarin Chinese) are not currently considered in UMR. Reduplication can occur in Mandarin by repeating a lexical unit, and can be indicative of either tentative aspects of emphasized meaning (Chen et al., 1992).

5 Challenges for UMR & Cross-lingual AMR

UMR & Cross-lingual AMR face a number of challenges when adapting to various languages, most notably in the representation of idiomatic phrases. Reliance on underlying lexicons leads to graph structural inconsistencies for parallel sentences.

Idiomatic Phrases. Idiomatic phrases are a challenge for cross-lingual AMR/UMR because of the relationship between a phrase's individual tokens and its overall meaning (Urešová et al., 2014; van der Plas et al., 2010; De Clercq et al., 2012; Kara et al., 2020). Even within a single language, it can be difficult for annotators to determine the best way to incorporate predicate argument structures associated with the specific combination of individual tokens (literal expression) and the argument

Feature	Spanish AMR	UMR
In-language verb senses	✓	✓
Modality	✓	✓
Grammatical Number	✗Opted for Simplicity	✓
Pronoun Drop	✓	Not specified
Politeness	✓	✓
Affixes (Third person clitic pronouns, Suffixes)	✓	✓
Double Negation	✓Same as single negation	Not specified
Document-level representation	✗	✓
Scope	✗	✓
Aspect	✗	✓
Se Usage	✓	✓Impersonal pronoun, ✓Reflexive pronouns, ✓Passive Voice

Table 2: A selection of linguistic features relevant for capturing meaning in Spanish, showing whether they are accounted for in each of the two schemata (Spanish AMR and UMR). The specific ways in which these features are accounted for in Spanish AMR and UMR are detailed in §4.1.

structure associated with the overall (idiomatic) semantics, especially when the expression is not fully compositional. Graph structures stemming from the relationships between individual tokens are, to some extent, unavoidable, and since idiomatic expressions of the same meaning can vary greatly across languages, the graph structures associated with a single meaning can also vary. An effectively cross-lingual meaning representation needs built-in considerations for addressing this challenge as uniformly as possible during annotation and parsing.

UMR has not yet established final guidelines for uniform treatment of all idiomatic phrases (but see Bonn et al. in press for further discussion), particularly during stage 0 annotation when there are no existing lexical resources to rely on that might provide a single predicate argument structure for an expression. In addition to the difficulties posed for parallel semantic representations across languages, this can also lead to inconsistencies across annotators. Still, inter-annotator agreement for small UMR annotation studies on Kukama and Arapaho, as measured by Smatch, ranges from 0.76 to 0.92, which is similar to typical AMR inter-annotator agreement scores (Van Gysel et al., 2021b).

Given that Stage 0 UMR permits annotation of tokens into multiple concepts (e.g. compound words) or of multiple tokens into a single concepts (e.g. multi-word concepts), we expect that an altered version of Smatch (Cai and Knight, 2013) will need to be adapted in order to successfully identify parallelism in meaning when quantitatively comparing UMRs in different languages.

Reliance on English Concepts. Prior work has explored cross-lingual differences in parallel AMRs and to what extent AMR is an interlingua (Xue et al., 2014; Wein and Schneider, 2021), and suggests that the AMR annotation schema may be more compatible with certain languages than others (i.e. more compatible with Chinese than Czech).

Current cross-lingual adaptations of AMR highlight this, because some cross-lingual guidelines require more changes to handle linguistic variation than others, though the structure of arguments and concepts remain largely unchanged. The approaches which use English abstract rolesets for the cross-lingual annotation (for example, *accompany-01* as the reification for the :accompanier role) exhibit significant English bias because the arguments for concepts are determined by their English usage.

AMR adaptations vary in degree of reliance on English annotations and resources, ranging from simply working with the English AMR guidelines as a baseline and extending them, to using English PropBank for sense annotation (Migueles-Abraira et al., 2018) or aligning English and Portuguese sentences and translating English annotations to their cross-lingual framesets (Sanches Duran and Aluísio, 2015). A factor that has enabled cross-lingual AMR extensions for individual languages is the existence of lexicons in those languages, such as PropBanks. This is an obstacle to AMR annotation for low-resource languages. Because many meaning representations require additional resources to

produce annotations, the lack of *prior* non-English resource work poses an issue for *future* non-English resource work (Hovy and Prabhumoye, 2021). This issue has been handled by UMR by developing a "road map" for annotation of low-resource languages (Van Gysel et al., 2021a).

Reliance on Frame Files. The quality/extent of the lexicon of rolesets available for a given language impacts AMR/UMR annotation. For example, Spanish AMR (Wein et al., 2022a) makes use of AnCora (Taulé et al., 2008), but despite being the most comprehensive publicly available lexical resource for Spanish, it is limited in the senses it contains, so other adaptations of AMR for Spanish have opted against its use (Migueles-Abraira et al., 2018). Thus, even with the "road map" for annotation of low-resource languages in UMR, there are complexities caused by reliance on external resources that affect UMR/AMR annotation.

Spanish AMR was forced to add a supplementary database of frame files / senses when using AnCora, and Stage 1 UMR annotation will likely also need to provide additional resources when relying on external lexicons. The UMR Writer (Zhao et al., 2021) is designed to allow annotators to add lexical entries to the roleset lexicon file used for annotation as need arises during annotation, pairing the lexicon-development process with UMR annotation. Roleset development can be incredibly complicated, however–particularly for polysynthetic and agglutinating languages like Arapaho–so this feature of the UMR-writer is a vital first step out of many when it comes to establishing a robust lexical resource.

6 Conclusion

Cross-lingual adaptations of AMR use the English annotation guidelines as a baseline, and then make a set of adaptations for linguistic features specific to the other language. The linguistic phenomena incorporated into each cross-linguistic adaptation also varies by language (as described in §2), because these phenomena are language-specific.

We conclude that UMR successfully handles the vast majority of even the more language-specific features of cross-lingual adaptations of AMR. The challenges for UMR annotation in need of further investigation and consideration include the development of quantitative metrics, which will need to account for UMR's flexibility in multiword/affix annotation, and the complexities associated with the generation of roleset lexicons for low-resource languages. Future work providing general insight into the morphosyntactic strategies of AMR and UMR might provide additional insight into their cross-lingual applicability.

Acknowledgements

This work is supported by a Clare Boothe Luce Scholarship and by grants from the CNS Division of National Science Foundation (Awards no: NSF_2213805, NSF_IIS 1764048 RI) entitled "Building a Broad Infrastructure for Uniform Meaning Representations" and "Developing a Uniform Meaning Representation for Natural Language Processing", respectively. We thank anonymous reviewers as well as Andrew Cowell, Bill Croft, Jan Hajic, Alexis Palmer, Martha Palmer, James Pustejovsky, and Nianwen Xue for their helpful feedback. All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

References

Rafael Anchiêta and Thiago Pardo. 2018. Towards AMR-BR: A SemBank for Brazilian Portuguese language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Zahra Azin and Gülşen Eryiğit. 2019. Towards Turkish Abstract Meaning Representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 43–47, Florence, Italy. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Julia Bonn, Andrew Cowell, Jan Hajič, Alexis Palmer, Martha Palmer, James Pustejovsky, Haibo Sun, Zdenka Urešová, Shira Wein, Nianwen Xue, and Jin Zhao. in press. Umr annotation of multiword expressions. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, Nancy, France. Association for Computational Linguistics.

Julia Bonn, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell,

William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. 2023. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C. 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.

Daniel Chen, Martha Palmer, and Meagan Vigus. 2021. AutoAspect: Automatic annotation of tense and aspect for uniform meaning representations. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 36–45, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Feng-yi Chen, Ruo-ping Mo, Chu-Ren Huang, and Keh-Jiann Chen. 1992. Reduplication in Mandarin Chinese: Their formation rules, syntactic behavior and ICG representation. In *Proceedings of Rocling V Computational Linguistics Conference V*, pages 217–233, Taipei, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Hyonsu Choe, Jiyoon Han, Hyejin Park, and Hansaem Kim. 2019. Copula and case-stacking annotations for Korean AMR. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 128–135, Florence, Italy. Association for Computational Linguistics.

Hyonsu Choe, Jiyoon Han, Hyejin Park, Tae Hwan Oh, and Hansaem Kim. 2020. Building Korean Abstract Meaning Representation corpus. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 21–29, Barcelona Spain (online). Association for Computational Linguistics.

Orphée De Clercq, Veronique Hoste, and Paola Monachesi. 2012. Evaluating automatic cross-domain Dutch semantic role annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 88–93, Istanbul, Turkey. European Language Resources Association (ELRA).

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637, Hong Kong, China. Association for Computational Linguistics.

Magali Sanches Duran and Sandra Maria Aluísio. 2011. Propbank-br: a Brazilian Portuguese corpus annotated with semantic role labels. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*.

Johannes Heinecke and Anastasia Shimorina. 2022. Multilingual Abstract Meaning Representation for celtic languages. In *Proceedings of the 4th Celtic Language Technology Workshop within LREC2022*, pages 1–6, Marseille, France. European Language Resources Association.

Dirk Hovy and Shrimai Prabhumoye. 2021. Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8):e12432.

Neslihan Kara, Deniz Baran Aslan, Büşra Marşan, Özge Bakay, Koray Ak, and Olcay Taner Yıldız. 2020. TRopBank: Turkish PropBank v2.0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2763–2772, Marseille, France. European Language Resources Association.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Kenneth Lai, Richard Brutti, Lucia Donatelli, and James Pustejovsky. 2021. Situated umr for multimodal interactions. In *Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue*.

Bin Li, Yuan Wen, Weiguang Qu, Lijun Bu, and Nianwen Xue. 2016. Annotating the little prince with Chinese AMRs. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 7–15, Berlin, Germany. Association for Computational Linguistics.

Ha Linh and Huyen Nguyen. 2019. A case study on meaning representation for Vietnamese. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 148–153, Florence, Italy. Association for Computational Linguistics.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.

Noelia Migueles-Abraira, Rodrigo Agerri, and Arantza Diaz de Ilarraza. 2018. Annotating Abstract Meaning Representations for Spanish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki,

Japan. European Language Resources Association (ELRA).

Roberto Navigli, Rexhina Blloshmi, and Abelardo Carlos Martinez Lorenzo. 2022. Babelnet meaning representation: A fully semantic formalism to overcome language barriers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12274–12279.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

Thi Minh Huyen Nguyen, Laurent Romary, Mathias Rossignol, and Xuân Luong Vũ. 2006. A lexicon for vietnamese language processing. *Language Resources and Evaluation*, 40(3):291–309.

Elif Oral, Ali Acar, and Gülşen Eryiğit. 2022. Abstract meaning representation of turkish. *Natural Language Engineering*, pages 1–30.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31:71–106.

Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O'gorman, James Gung, Kristin Wrightbettner, and Martha Palmer. 2022. PropBank comes of Age—Larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. Association for Computational Linguistics.

Magali Sanches Duran and Sandra Aluísio. 2015. Automatic generation of a lexical resource to support semantic role labeling in Portuguese. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 216–221, Denver, Colorado. Association for Computational Linguistics.

Marco Antonio Sobrevilla Cabezudo and Thiago Pardo. 2019. Towards a general Abstract Meaning Representation corpus for Brazilian Portuguese. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 236–244, Florence, Italy. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2015. Conceptual annotations preserve structure across translations: A French-English case study. In *Proceedings of the 1st Workshop on Semantics-Driven Statistical Machine Translation (S2MT 2015)*, pages 11–22, Beijing, China. Association for Computational Linguistics.

Reza Takhshid, Razieh Shojaei, Zahra Azin, and Mohammad Bahrani. 2022. Persian Abstract Meaning Representation. *arXiv preprint arXiv:2205.07712*.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Zdeňka Urešová, Jan Hajič, and Ondřej Bojar. 2014. Comparing Czech and English AMRs. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing*, pages 55–64, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Lonneke van der Plas, Tanja Samardžić, and Paola Merlo. 2010. Cross-lingual validity of PropBank in the manual annotation of French. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 113–117, Uppsala, Sweden. Association for Computational Linguistics.

Jens E. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021a. Designing a uniform meaning representation for natural language processing. *KI - Künstliche Intelligenz*.

Jens E. L. Van Gysel, Meagan Vigus, Lukas Denk, Andrew Cowell, Rosa Vallejos, Tim O'Gorman, and William Croft. 2021b. Theoretical and practical issues in the semantic annotation of four indigenous languages. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 12–22, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shira Wein, Lucia Donatelli, Ethan Ricker, Calvin Engstrom, Alex Nelson, Leonie Harter, and Nathan Schneider. 2022a. Spanish Abstract Meaning Representation: Annotation of a general corpus. In *Northern European Journal of Language Technology, Volume 8*, Copenhagen, Denmark. Northern European Association of Language Technology.

Shira Wein, Wai Ching Leung, Yifu Mu, and Nathan Schneider. 2022b. Effect of source language on AMR structure. In *Proceedings of the 16th Linguistic Annotation Workshop (LAW-XVI) within LREC2022*, pages 97–102, Marseille, France. European Language Resources Association.

Shira Wein and Nathan Schneider. 2021. Classifying divergences in cross-lingual AMR pairs. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 56–65, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nianwen Xue, Ondřej Bojar, Jan Hajič, Martha Palmer, Zdeňka Urešová, and Xiuhong Zhang. 2014. Not an

interlingua, but close: Comparison of English AMRs to Chinese and Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1765–1772, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jin Zhao, Nianwen Xue, Jens Van Gysel, and Jinho D. Choi. 2021. UMR-writer: A web application for annotating uniform meaning representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 160–167, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuping Zhou and Nianwen Xue. 2015. The chinese discourse treebank: a chinese corpus annotated with discourse relations. *Language Resources and Evaluation*, 49:397–431.

Abstract Meaning Representation for Grounded Human-Robot Communication

Claire Bonial[1], **Julie Foresta**[1], **Nicholas C. Fung**[1], **Cory J. Hayes**[1],
Philip Osteen[1], **Jacob Arkin**[2], **Benned Hedegaard**[2], **Thomas M. Howard**[2]
[1] Army Research Lab, [2] University of Rochester
claire.n.bonial.civ@army.mil,

Abstract

To collaborate effectively in physically situated tasks, robots must be able to ground concepts in natural language to the physical objects in the environment as well as their own capabilities. We describe the implementation and the demonstration of a system architecture that supports tasking robots using natural language. In this architecture, natural language instructions are first handled by a dialogue management component, which provides feedback to the user and passes executable instructions along to an Abstract Meaning Representation (AMR) parser. The parse distills the action primitives and parameters of the instructed behavior in the form of a directed a-cyclic graph, passed on to the grounding component. We find AMR to be an efficient formalism for grounding the nodes of the graph using a Distributed Correspondence Graph. Thus, in our approach, the concepts of language are grounded to entities in the robot's world model, which is populated by its sensors, thereby enabling grounded natural language communication. The demonstration of this system will allow users to issue navigation commands in natural language to direct a simulated ground robot (running the Robot Operating System) to various landmarks observed by the user within a simulated environment.

1 Introduction

Robots are increasingly used for their potential in disaster relief and search and rescue tasks (Murphy, 2014). There is a clear benefit to this, as robots can be used to provide aid and give situational awareness of the environment to people, who can remain at a safe distance and use information gathered by the robot to knowledgeably address the situation. Using robots in this way has required advances in robotics; however, robots in the current paradigm are still treated more as tools—often requiring human teleoperation, which inhibits the operator's awareness of their own immediate surroundings in potentially dangerous situations. The ability to speak to a robot as one would another human teammate would reduce the training time and cognitive burden on the operator, making the collaborative response more efficient. While there have also been relevant advances in task-oriented dialogue systems, such as Siri and Alexa, as well as widespread interest in systems leveraging large language models such as ChatGPT, these systems are limited in their applicability to physically situated tasks because they do not address grounding natural language to the physical environment of an embodied platform. In this paper, we describe a novel system architecture that supports grounded, bi-directional human-robot dialogue. This architecture is depicted in Figure 1.

In the sections to follow, we first provide a conceptual overview of the system capabilities (§2), and then detail the components of this architecture (§3) while highlighting the novel and primary contribution of the symbol grounding components: the Abstract Meaning Representation (AMR) (Banarescu et al., 2013) parser (§3.4), which we show to be uniquely suited to distill the action primitives and their parameters in a way that can be efficiently grounded, using our updated Distributed Correspondence Graph (DCG) (Howard et al., 2014) grounding component (§3.5). We then describe the demo (§4) and detail how distinct demo modes (§4.1) allow users to experience performance differences when the grounding component receives input from either a syntactic constituency parser or the meaning-based, AMR parser. We provide a brief comparison to related work (§5) and conclude with directions for ongoing and future work (§6).

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 34–44
June, 2023.

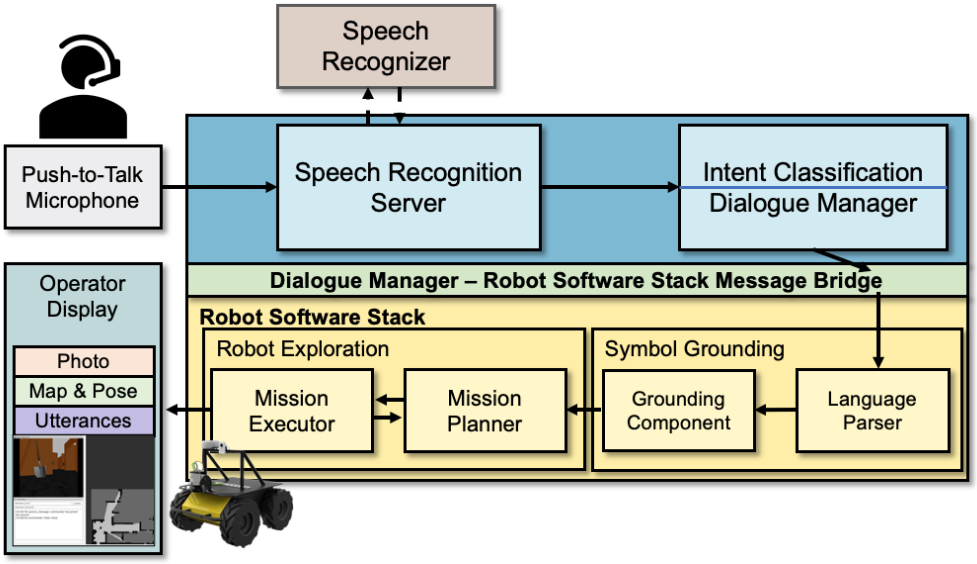


Figure 1: System architecture, supporting bi-directional, grounded communication between an operator and a remotely located robot.

2 System Capabilities

The implemented system in this research allows a human operator to speak to a remotely located robot in natural language, providing search and navigation instructions for the robot to execute. The current system has been successfully implemented for natural language control of a Clearpath Husky Unmanned Ground Vehicle (Clearpath Robotics, 2023) (shown in Figure 1), measuring about 39 inches in length and weighing about 110 pounds, which autonomously executes the natural language navigation instructions. In our implementation, the Husky is equipped with a LORD Microstrain 3DM-GDX5-25 IMU, Ouster OS1-64 Gen 1 Light Detection and Ranging (LIDAR) unit, and a Teledyne FLIR Blackfly GigE camera with a KOWA LMVZ41 high resolution camera lens. The robot computers consist of two Intel i7 equipped computers with NVIDIA 1650Ti graphics cards installed.

The robot runs on the Robot Operating System (ROS); thus, part of our research here includes creating a ROS wrapper around the AMR parsing component. The same ROS software stack can be used either within real-world robots or in simulation, and we have implemented and tested our architecture in both environments.

Because connectivity and bandwidth can be limited in disaster relief scenarios, our setup does not require internet connectivity, but it does currently require a stand-alone machine to run the natural language communication interface and dialogue management capabilities (shown in the top half of the architecture diagram in Figure 1), whereas the rest of the system architecture components run fully onboard the robot (shown in the bottom half of the architecture diagram in Figure 1).

3 System Components

In the following sections, we provide an overview of each of the architecture's components. We devote the most description to the primary novel contribution of this paper: the symbol grounding, which leverages an AMR parser together with a DCG grounding component.

3.1 Speech Recognition

The operator speaks to the robot using a microphone, currently implemented as the standard microphone capability of the computer running the user-facing dialogue interface components. The operator presses on an assigned key and speaks their instructions.

The speech recognition server listens to the user's speech and sends it to the speech recognizer component; we are currently leveraging the open-source Kaldi speech recognition toolkit (Povey et al., 2011). Kaldi provides automatic speech recognition (ASR), producing a text transcription of the user's speech. We selected Kaldi because we find that it gives relatively high-accuracy ASR but does not require internet connectivity.

3.2 Intent Classification & Dialogue Management

The text output from Kaldi is passed along to the joint intent classification and dialogue management component. This component has two elements: first, a classifier interprets the language with respect to the basic intent, and second, a dialogue manager dictates what the system should do next. For example, if the operator provides the instruction *Okay, Husky, check the path in front of you*, the system retrieves the most similar example to this seen in the training data, for example, *Scout the path in front*. The system would then provide an associated response message such as *executing* to provide feedback to the user. Finally, the system would pass the text instruction *Scout the path in front* along to the parsing component operating within the software stack for processing and eventual execution. This component is an adaptation of the Virtual Human Toolkit described in Hartholt et al. (2013), refined to support a robot platform (Marge et al., 2016).

Intent classification is treated as a retrieval problem, such that given the transcribed speech from the recognizer, the system can infer the intent by retrieving the most similar example from training data. The training data is organized into instruction-response pairs, where instructions are previously seen operator instructions, and responses are either messages sent back to the operator (such as feedback or clarification questions) or messages sent on to the robot software stack for further processing and execution. The training data instruction-response pairs are curated for a particular domain within a spreadsheet used to learn the weights of association such that a ranked list of potential matches is returned and the most similar instruction-response pair is selected (Leuski and Traum, 2011). In our implementation, the training data pairs are drawn from a corpus of human-robot collaborative dialogue for search and navigation, collected in a wizard-of-oz experimental paradigm (Marge et al., 2016) and subsequently annotated for relevant features of dialogue structure (Traum et al., 2018).

Dialogue management policies are defined based upon the matches obtained from the intent classifier, with two basic categories of response policies. The first is for actionable messages, where the robot is able to execute the instruction. For actionable commands, the basic policy is to jointly respond to the operator with feedback, demonstrating successful receipt of the instruction, and to send a simple text message of the instruction on to the robot software stack. The second policy is for non-actionable messages, which require clarification through further dialogue. The basic policy for non-actionable messages is to prompt the operator for clarification, such that any inability to infer the intent of the instruction can be overcome immediately through dialogue.

3.3 Message Bridge

A message bridge enabled by the Virtual Human Toolkit from Hartholt et al. (2013) connects the operator-facing natural language interface (which runs on a computer used by the operator) to the robot's software autonomy stack (which runs on the robot's onboard computer). The bridge enables connectivity between the two computers—sending synchronous messages from the operator-facing computer to the robot's computer and back again. Additionally, it enables the transfer between the two operating systems, where the output of the operator-facing computer is simply text, and is formatted as Robot Operating System (ROS) messages delivered to the software autonomy stack via a ROS topic for the robot to process.

3.4 Language Parser

We leverage the open-source AMR parser from Lindemann et al. (2019), specifically a model that has been retrained on a portion of the same human-robot dialogue corpus used to derive the instruction-response pairs described in §3.2. We selected this parser because the retrained model outperformed other competitive parsers retrained on the same small set of robot-directed instructions (Bonial et al., 2020), but we are working to make our implementation agnostic to any particular parser so that we can swap it out based on the current state of the art.

We implement wrapper code to interface the open-source AMR parser with ROS code that operates the automated systems aboard the robot including perception and motor control. The wrapper code takes in commands through ROS messages. These messages can be generated by the autonomy stack running on the robot, piped directly to the AMR parser as a string through ROS commands, or generated by other software. In our case, the dialogue manager generates these commands and the message bridge publishes them as a string to a

ROS topic. The command string is extracted from the ROS message and used as input for the AMR parser.

Thus, the parser accepts the text instructions output by the dialogue manager, and parses this into an AMR directed, a-cyclic graph (DAG). Because AMR abstracts away from some idiosyncratic linguistic variation in favor of representing core concepts, the AMR parse is a very effective distillation of action primitives and the parameters of that action. For example, regardless of whether the operator instructs the robot to *Drive to the barrel on the left* or *Take a drive to the left barrel*, these instructions will be encoded with identical AMR graphs, shown in Figure 2 in the textual, Penman style (Penman Natural Language Group, 1989) as opposed to a DAG.

```
(d / drive-01 :mode imperative
    :ARG0 (y / you)
    :ARG1 y
    :destination (b / barrel
        :ARG1-of (l / left-20)))
```

Figure 2: AMR graph for the input *Drive to the barrel on the left* and the alternatively-worded input *Take a drive to the left barrel.*

AMR therefore offers a level of abstraction that is suitable for a robot to act upon as it glosses over some of the linguistic complexity that does not carry any meaningful difference for execution. Furthermore, we find that AMR is well-suited as an input representation to the grounding component because the node concepts of the graph that are grounded are restricted to the action concept and its parameters (such as the destination of a movement instruction). Leveraging AMR allows us to directly associate the **meaning** of the instructions with the physical world, instead of attempting to ground all of the **words** of the instruction, which may include syntactic scaffolding, such as *take* in *take a drive*, that has no grounding in a robot's behavior or the objects in its environment. Benefits of leveraging AMR are further described in §4.1.

After parsing, the wrapper code will interpret the textual representation output from the AMR parser and generate outgoing ROS messages to be published on an established ROS topic. Any ROS software can obtain these messages by subscribing to this topic. In our case, the grounding software component running on the robot will take in these messages and ground the instruction into mission commands for the robot.

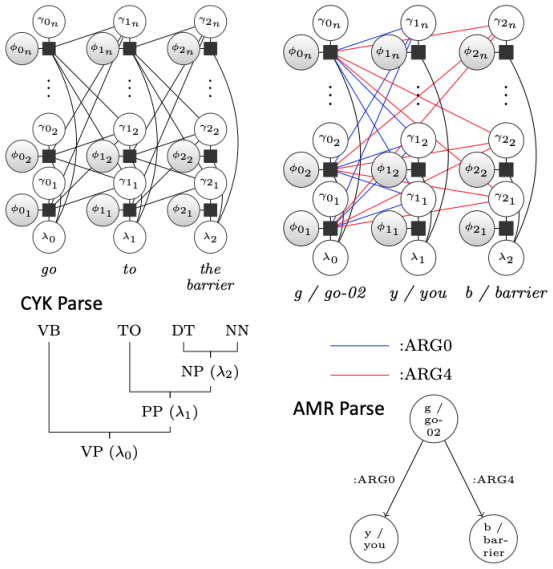


Figure 3: The constituency parse-based DCG on the left exhibits the same number of factors but lacks the informative relational structures of the AMR-based DCG on the right.

3.5 Grounding Component

We take a graphical approach to grounding using a model based on the Distributed Correspondence Graph (Howard et al., 2014). A DCG consists of a set of constituents of language $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ (e.g., phrases in a parse tree or nodes/edges in an AMR graph), a world model Υ (typically a metric-semantic object-level model), a set of grounding symbols $\Gamma = \{\gamma_1, \ldots \gamma_M\}$ that represent physical concepts (e.g., objects, spatial relationships, robot actions), and a set of binary correspondence variables $\Phi = \{\phi_{11}, \ldots \phi_{NM}\}$ representing True or False correspondence between an individual phrase and individual grounding symbol.

The formulation of DCGs assumes conditional independence of both grounding symbols and linguistic constituents excepting child constituents, resulting in a factor graph hierarchically structured according to the representation of language. Each factor computes the probability of correspondence (ϕ) between a given phrase (λ) and grounding symbol (γ), in the context of a model of the environment. The probabilities are computed by a single log-linear model (Collins, 2005) consisting of expert-designed binary features with associated optimized weights trained from a corpus of annotated data. The features jointly evaluate properties of language and the world, such as a unigram feature

for *barrier* and an indicator feature for an object grounding symbol that is True if the object is a *barrier* type, thereby allowing the log-linear model to learn to ground language in physical concepts. Inference is performed via bottom-up beam search to find the most likely True correspondences for each linguistic constituent; this process propagates up the hierarchy of the graph. The grounded interpretation of the instruction is represented by the True corresponding symbols at the root.

In previous works (Paul et al., 2018; Patki et al., 2020; Howard et al., 2021), a syntactic constituency parse tree, produced by the Cocke-Younger-Kasami (CYK) parsing algorithm (Younger, 1967), was used to represent language instructions; the resulting DCGs inherited the compositional structure of the hierarchy of phrases. A novelty of this work is that we construct a DCG from an AMR parse. A DCG constructed from an AMR parse differs than one constructed from a constituency parse tree because the edges in an AMR parse are labeled. In this work, we assume that there are no cycles in the AMR parse. Consider the example illustrated in Figure 3. For the same language, a parse tree is shown on the left and an AMR parse is on the right. The corresponding constituency parse-based DCG, also shown on the left, expresses a set of symbols for the phrases *the barrier*, *to the barrier*, and *go to the barrier*, where the symbols corresponding to the last phrase represent the grounding of the entire statement. The structure of the AMR-based DCG, shown on the right, differs. Here the AMR-based DCG expresses a set of symbols for the node concepts `y / you`, `b / barrier`, and `g / go-02`. How `y / you` and `b / barrier` are interpreted by `g / go-02` is influenced by the labels of each edge, which are `:ARG0` and `:ARG4`, respectively. To properly capture the structure of this AMR parse, the associated DCG must incorporate the labels of each edge into its own structure; this provides the edge label context to the log-linear model features at each factor, which is necessary to correctly interpret the expressed symbols at child nodes. These differently labeled edges, illustrated in red and blue respectively, are now used in the construction of DCGs so that the engineered features that compose the log-linear model-based factors can utilize this information when determining if a feature is active or inactive. AMR also differs from parse trees in that nodes are permitted to have more than one parent (reentrancy). These are naturally handled by the conditional independence of linguistic constituents that is assumed in the DCG formulation.

In this example, although both models exhibit the same number of factors, the structure of the AMR-based DCG provides richer information, including an explicit representation of who is meant to execute the command. This information is left out of the CYK-based DCG when the imperative is used, as the subject is omitted in the English imperative form.

There are other situations where an AMR-based DCG is preferable to a constituency parse tree-based DCG. For example, the approach leveraging CYK parses required training instances reflecting alternative wordings of what is semantically the same instruction, such as for light-verb constructions. In contrast, our approach enables grounding with less training data since we are grounding the deeper meaning instead of the surface word-forms of the instruction. Another benefit to grounding the meaning behind the instruction, as opposed to the words themselves, is that our implementation is able to more efficiently ground instructions involving co-reference and complex spatial relations, both of which are represented explicitly and consistently in AMR (see §4.1 for further discussion).

3.6 Mission Planner and Executor

Once the action and the action parameters, including any objects mentioned in the instruction, have been grounded, the grounding component sends the action specification to the mission planner. The grounded action includes specifications such as path end points as specified by the location of grounded objects in the robot's world model. For this implementation, we use Cohen et al. (2010)'s Search Based Planning Library global planner and Howard and Kelly (2007)'s Nonlinear Optimization (NLOPT) local planner. Once a plan has been established, the robot mission executor generates and performs the appropriate actions, taking into account real-time feedback from the robot such as the perception of moving obstacles. This completes the loop from natural language instruction to execution within the robot's current physical environment.

4 Demo Description

In the demo, audience members will be invited to interact with the system at a computer workstation

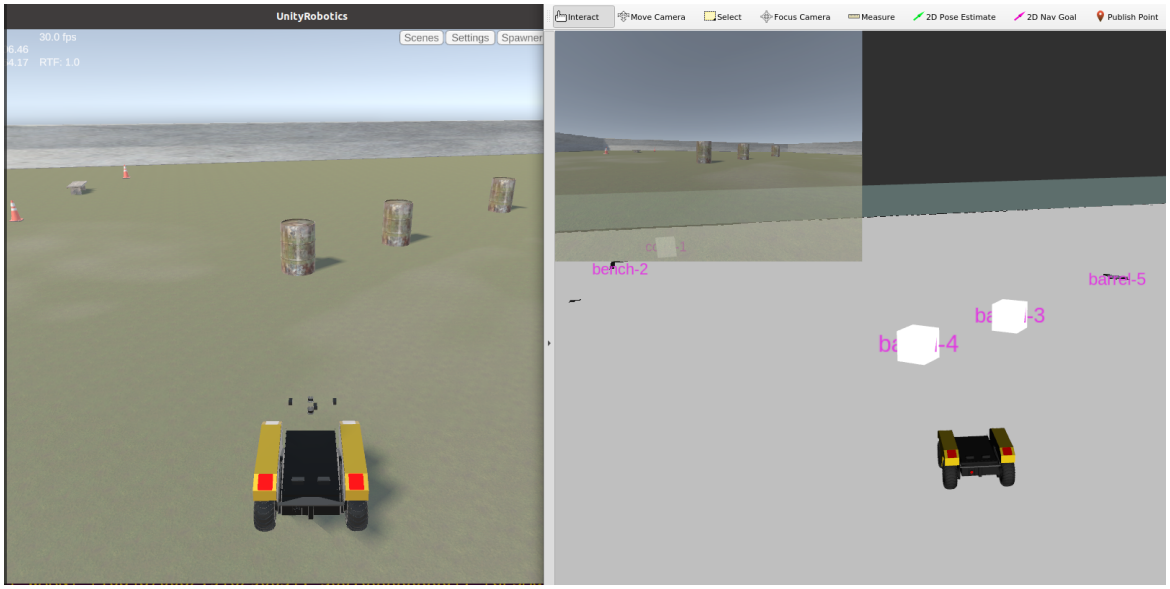


Figure 4: Screen capture of demo, where the left pane of the screen shows the robot's position within the simulation and the right pane of the screen shows the robot's world model view, populated by a LIDAR terrain map and labeled, recognized objects.

where the audience can see a view of the robot in the simulated environment on one side of the screen and what is essentially the robot's view of the world, or its world model, on the other side of the screen. The world model pane shows a LIDAR-derived map of the simulated environment where detected objects in the environment are represented by white boxes with pink labels.

Figure 4 is a screen capture of the demo workstation. In the left pane, visitors can see that the robot is approaching a set of three barrels extending out to the right, and two cones with a small bench in between them ahead of the robot and to its left. In the right pane, visitors can see a visual representation of what the robot "sees" in this same environment using its LIDAR and computer vision sensors. There is a snapshot of exactly what the robot sees from it's onboard camera in the small pane in the top left corner of the right pane. The rest of the right pane populates with light grey in the areas reached by the LIDAR that have been classified as open space; thus, there are some darker grey unknown or unexplored areas beyond the grey barrier that encloses the demo environment. The robot recognizes the three barrels, the cones and the bench. These objects are labeled with the basic object type label as well as a unique identifier number that tracks these objects in the robot's world model. For example, the robot labels the closest barrel as "barrel-4". Demo audience members will be able to direct the robot to any of the objects in the scene that the robot has identified and successfully labeled thus far.

4.1 Demo Modes Comparing AMR & CYK Parsers

In order to showcase the novel contribution of this research, the demo host can toggle the implementation back and forth between the same architecture with either the AMR parser described in Section 3.4, or the syntactic CYK parser (Younger, 1967) of previous implementations, such as Howard et al. (2021). This setup allows us to compare our architecture to comparable systems where the CYK parser was used. However, to make this a fair comparison that focuses only on the language parse and the grounding component, we hold the rest of the architecture constant while only swapping out and comparing the symbol grounding components. This will allow audience members to use different variations of navigation instructions in order to see how a small amount of complexity in the surface form can affect the grounding when using meaning-based (this work) or syntax-based (baseline) parsers.

For example, in our own preliminary comparisons, an experimenter issued the following set of three instructions, given in the same simulated environment to a robot with the same sensors and resulting world model. Only the AMR-based system was able to ground the final two instructions,

which involve a light verb construction (2), and coreference as well as a complex spatial expression (3):

1. Go to the left barrel.
2. Take a drive to the left barrel.
3. Drive to the cone and the rock closest to it.

While sufficient for the simple instruction in (1), the syntactic CYK parser output fails to be grounded for instruction (2) because the system cannot ground what it presumes to be the primary *take* action, which has not been seen in training data for either the constituency parse or AMR-based grounding. In the AMR input, *take* is abstracted away and this instruction is grounded to a driving behavior.

For instruction (3), the CYK parser output includes the words *cone* and *it*, which are co-referential expressions for the same object in the environment; thus, the constituency parse tree-based grounding component attempts to separately ground each word. Although this may eventually result in the correct grounding, it is much more computationally expensive and requires a larger space of potential groundings, including symbols for each co-referential expression, in order to compositionally build the grounded meaning of each linguistic constituent. In contrast, AMR represents co-referential expressions as a single node, which is then grounded to a single symbolic meaning.

Instruction (3) also includes the complex spatial expression *the rock closest to it*, which, combined with the coreference, causes the syntax-based grounding to fail altogether. The AMR specifies this as the `close` relation between the concepts of *rock* and *cone*, abstracting away any explicit constituent for the word *it*. Thus, the AMR enables grounding of such spatial concepts to real-world spatial relations between objects in the world model observed in training data.

5 Related Work

This research is at the intersection of NLP, including semantic parsing and dialogue systems, and robotics. We limit our direct comparison here to similarly interdisciplinary work; see Tellex et al. (2020) for a full review of research in robotics and language. Outside of the work on the DCG grounding approach that we directly augment for AMR (Howard et al., 2021), field robotics has largely focused on robots that receive an initial, static tasking and then operate autonomously (e.g., Williams et al. (2012); Arvidson et al. (2010); Camilli et al. (2010)), or robots that are tele-operated (e.g., Kang et al. (2003); Ryu et al. (2004); Yamauchi (2004)). In contrast, there is relatively little work like ours, seeking to develop robots that are able to be tasked dynamically and interactively via natural language.

There are, however, several notable exceptions. Walter et al. (2015) describe the development of a voice-controlled fork lift. In contrast to our own research, however, the natural language instructions are more constrained to particular hard-coded commands mentioning a more limited range of objects that are classified in their world model. Additionally, Heikkilä et al. (2012) develop a mobile manipulator designed for space operations that is capable of accepting spoken commands. Unlike both of the previously mentioned voice-controlled robots, it is important to note that our architecture aims to support bi-directional communication between the robot and the operator, such that ambiguities that might arise in changing environments can be resolved.

There is also relevant research leveraging large, pretrained language models to map or translate between unconstrained natural language and the controlled planning languages of robots. Song et al. (2022) utilize GPT for deciding upon the appropriate high-level plan given natural language instructions, and then use a more traditional low-level planning component to execute specific motor movements to specific grounded points in the environment. The high-level and low-level models are also able to communicate, such that the high-level model can be queried for new and updated plans if conflicts arise in the low-level planning model. Driess et al. (2023) develop their own multi-modal "embodied" language model, called PaLM-E, which accepts both sensor data, such as image data, and natural language text. The model outputs text data that can be interpreted as robot policies. In general, we see potential for leveraging language models in the future both for providing some *apriori*, zero-shot knowledge of objects that the robot might encounter in its environment, which can be used to inform the interpretation of natural language instructions, as well as for providing a likely mapping between unconstrained natural language and the constrained set of robot behaviors.

However, explainability is critical for adoption

of robotic systems in high-stakes tasks such as disaster relief; thus, further research enabling transparency and explainability of systems leveraging language models is needed. Neuro-symbolic approaches (e.g., Dipta et al. (2022)) are promising for providing greater transparency. For example, Zhang et al. (2022) develop DANLI, which symbolically represents subgoals as predicates on objects in the robot's world model.

There is a growing body of research leveraging AMR for NLU in human-agent interaction. The present research is part of a broader ongoing research effort leveraging a two-step NLU pipeline that first parses natural language into AMR, which abstracts away from some surface variation, but then in a second step converts the Standard-AMR into a formalism called Dialogue-AMR (Bonial et al., 2020). Dialogue-AMR is augmented to capture features of language found to be critical for human-robot dialogue, but not included in Standard-AMR (Bonial et al., 2019). Specifically, the Dialogue-AMR adds information on the input instruction's tense and aspect, and further normalizes varying expressions for a desired behavior (e.g., *turn, rotate, pivot*) to a single designated roleset for a particular robot behavior (e.g., `turn-01`). While the present research leverages Standard-AMR as the input to the grounding component, we will shift to using Dialogue-AMR as the input parse, as we expect that the further normalization will allow us to achieve comparable results with even less training data. Furthermore, Dialogue-AMR leverages spatial rolesets from Spatial-AMR (Bonn et al., 2020), which provides detailed relations for spatial relations for expressions such as *in front of*, which currently does not have a detailed representation with a relational concept in Standard-AMR.

Other research to augment AMR for interaction includes work to further develop multi-modal, gestural AMR (Brutti et al., 2022) as well as efforts to further develop aspect and modality in AMR to support NLU (Donatelli et al., 2020). Finally, there is research in leveraging AMR parses of image captions in order to develop scene graphs, which can help agents to summarize and process visual scenes (Choi et al., 2022a,b). Together, all of these threads of research demonstrate ways in which AMR can serve as a unified representation for making sense of multiple modalities of information.

6 Conclusions and Future Work

We are currently engaged in experimentation to evaluate the AMR-based grounding. Our ongoing extrinsic evaluation compares natural language interaction with the current paradigm of teleoperation. Specifically, we compare the time it takes for a robot operating autonomously to complete natural language instructions, using the architecture shown in Figure 1, to the time that it takes a relatively experienced person to teleoperate the robot and complete the same instruction. This comparison is made with and without the introduction of latency, which can occur when operators teleoperate a robot from distant, remote locations. The latency, or delay, between the manual teleoperation and the robot's execution of the teleoperation can be disorienting to operators (imagine, for example, if movements of your own body were delayed for some time after your brain sending the signal to move). This disorientation can cause delays in reaching the destination, an inability to reach precise locations, or even crashes. Such latency does not have a dramatic effect on natural language instructions, since although these might be delayed momentarily in getting to the robot, the robot is then navigating autonomously based upon the plan expressed in language. Our early results show that while autonomous navigation is generally slower than teleoperation, with anything over one second of latency introduced, the speed of autonomous navigation becomes comparable.

We are also carrying out an intrinsic evaluation where we compare our architecture, with the AMR parser, against an implementation with a CYK parser in order to robustly evaluate our system against the comparable system of Howard et al. (2021). We will evaluate the performance in terms of the ability of each system to successfully ground a wide variety of instructions with the same training set, and we will also compare computation time and efficiency. Once our evaluations leveraging Standard-AMR are complete, we will then turn to comparing to the use of Dialogue-AMR, where we expect even greater computational efficiency since Dialogue-AMR abstracts even further from surface variation to normalize a variety of different expressions of different behaviors into a single AMR roleset designated for a robot behavior.

Finally, although not the focus of this paper, we are also working to update our architecture such that the intent classification and dialogue manage-

ment components work more synergistically with the grounding and planning components. Therefore, the system can draw upon its knowledge of the surrounding environment to support more human-like conversational repairs in cases of ambiguities and miscommunications. For example, if the system encounters the well-formed instruction, *Move to the barrel on the right*, but there is no barrel grounded on the right and instead a barrel grounded on the robot's left, then that information from the grounding component can support generation, via AMR, of a targeted clarification question, such as *I don't see a barrel on the right; do you mean the one on the left?* This requires a level of intercommunication of the components that we currently have not achieved.

In this demonstration of our research, we show that AMR-based grounding of natural language instructions allows our system to successfully ground and execute instructions with a range of linguistic phenomena, including light verb constructions, coreference, and spatial relations. Although these phenomena are arguably complex for grounding and have proven to be challenging for the existing state-of-the-art systems, they are commonplace in natural language; thus, we simply must have systems that can handle such complexity reliably in disaster relief scenarios. In the demonstration that we offer, visitors will be able to explore this firsthand to see how our system addresses these challenges by grounding the **meaning** of the instructions, rather than just the **words** of the instructions.

References

Raymond E Arvidson, James F Bell III, P Bellutta, Nathalie A Cabrol, JG Catalano, J Cohen, Larry S Crumpler, DJ Des Marais, TA Estlin, WH Farrand, et al. 2010. Spirit mars rover mission: Overview and selected results from the northern home plate winter haven to the side of scamander crater. *Journal of Geophysical Research: Planets*, 115(E7).

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

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.

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract meaning representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583.

Richard Camilli, Christopher M Reddy, Dana R Yoerger, Benjamin AS Van Mooy, Michael V Jakuba, James C Kinsey, Cameron P McIntyre, Sean P Sylva, and James V Maloney. 2010. Tracking hydrocarbon plume transport and biodegradation at deepwater horizon. *Science*, 330(6001):201–204.

Woo Suk Choi, Yu-Jung Heo, Dharani Punithan, and Byoung-Tak Zhang. 2022a. Scene graph parsing via abstract meaning representation in pre-trained language models. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 30–35.

Woo Suk Choi, Yu-Jung Heo, and Byoung-Tak Zhang. 2022b. Sgram: Improving scene graph parsing via abstract meaning representation. *arXiv preprint arXiv:2210.08675*.

Clearpath Robotics. 2023. Clearpath Husky UGV.

Benjamin J Cohen, Sachin Chitta, and Maxim Likhachev. 2010. Search-based planning for manipulation with motion primitives. In *2010 IEEE international conference on robotics and automation*, pages 2902–2908. IEEE.

Michael Collins. 2005. Log-linear models. *Self-Published Tutorial*.

Shubhashis Roy Dipta, Mehdi Rezaee, and Francis Ferraro. 2022. Semantically-informed hierarchical event modeling.

Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2020. A two-level interpretation of modality in human-robot dialogue. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4222–4238, Barcelona, Spain (Online).

International Committee on Computational Linguistics.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model.

Arno Hartholt, David Traum, Stacy C Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. 2013. All together now: Introducing the virtual human toolkit. In *Intelligent Virtual Agents: 13th International Conference, IVA 2013, Edinburgh, UK, August 29-31, 2013. Proceedings 13*, pages 368–381. Springer.

Seppo S Heikkilä, Aarne Halme, and André Schiele. 2012. Affordance-based indirect task communication for astronaut-robot cooperation. *Journal of field robotics*, 29(4):576–600.

Thomas Howard and Alonzo Kelly. 2007. Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research*, 26(2):141 – 166.

Thomas M Howard, Nicholas Roy, Jonathan Fink, Jacob Arkin, Rohan Paul, Daehyung Park, Subhro Roy, D Barber, Rhyse Bendell, Karl Schmeckpeper, et al. 2021. An intelligence architecture for grounded language communication with field robots. In *Field Robotics, 2021*. Field Robotics.

Thomas M. Howard, Stefanie Tellex, and Nicholas Roy. 2014. A natural language planner interface for mobile manipulators. pages 6652–6659. Institute of Electrical and Electronics Engineers Inc.

Sungchul Kang, Changhyun Cho, Jonghwa Lee, Dongseok Ryu, Changwoo Park, Kyung-Chul Shin, and Munsang Kim. 2003. Robhaz-dt2: Design and integration of passive double tracked mobile manipulator system for explosive ordnance disposal. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2624–2629. IEEE.

Anton Leuski and David Traum. 2011. Npceditor: Creating virtual human dialogue using information retrieval techniques. *Ai Magazine*, 32(2):42–56.

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.

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 *Proc. of IEEE International Symposium on Robot and Human Interactive Communication*.

Robin R Murphy. 2014. *Disaster robotics*. MIT press.

Siddharth Patki, Ethan Fahnestock, Thomas M Howard, and Matthew R Walter. 2020. Language-guided semantic mapping and mobile manipulation in partially observable environments. In *Conference on Robot Learning*, pages 1201–1210. PMLR.

Rohan Paul, Jacob Arkin, Derya Aksaray, Nicholas Roy, and Thomas M. Howard. 2018. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *International Journal of Robotics Research*, 37:1269–1299.

Penman Natural Language Group. 1989. The Penman user guide. *Technical report, Information Sciences Institute*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Dongseok Ryu, Sungchul Kang, Munsang Kim, and Jae-Bok Song. 2004. Multi-modal user interface for teleoperation of robhaz-dt2 field robot system. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 1, pages 168–173. IEEE.

Chan Hee Song, Jihyung Kil, Tai-Yu Pan, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2022. One step at a time: Long-horizon vision-and-language navigation with milestones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15482–15491.

Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. 2020. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:25–55.

David Traum, Cassidy Henry, Stephanie Lukin, Ron Artstein, Felix Gervits, Kimberly Pollard, Claire Bonial, Su Lei, Clare Voss, Matthew Marge, et al. 2018. Dialogue structure annotation for multi-floor interaction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Matthew R Walter, Matthew Antone, Ekapol Chuangsuwanich, Andrew Correa, Randall Davis, Luke Fletcher, Emilio Frazzoli, Yuli Friedman, James Glass, Jonathan P How, et al. 2015. A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments. *Journal of Field Robotics*, 32(4):590–628.

Stefan B Williams, Oscar R Pizarro, Michael V Jakuba, Craig R Johnson, Neville S Barrett, Russell C Babcock, Gary A Kendrick, Peter D Steinberg, Andrew J Heyward, Peter J Doherty, et al. 2012. Monitoring of benthic reference sites: using an autonomous underwater vehicle. *IEEE Robotics & Automation Magazine*, 19(1):73–84.

Brian M Yamauchi. 2004. Packbot: a versatile platform for military robotics. In *Unmanned ground vehicle technology VI*, volume 5422, pages 228–237. SPIE.

Daniel H Younger. 1967. Recognition and parsing of context-free languages in time n3. *Information and control*, 10(2):189–208.

Yichi Zhang, Jianing Yang, Jiayi Pan, Shane Storks, Nikhil Devraj, Ziqiao Ma, Keunwoo Peter Yu, Yuwei Bao, and Joyce Chai. 2022. Danli: Deliberative agent for following natural language instructions. *arXiv preprint arXiv:2210.12485*.

Annotating Situated Actions in Dialogue

Christopher Tam, Richard Brutti, Kenneth Lai, James Pustejovsky
Department of Computer Science
Brandeis University
Waltham, MA, USA
{christophertam,brutti,klai12,jamesp}@brandeis.edu

Abstract

Actions are critical for interpreting dialogue: they provide context for demonstratives and definite descriptions in discourse, and they continually update the common ground. This paper describes how Abstract Meaning Representation (AMR) can be used to annotate actions in multimodal human-human and human-object interactions. We conduct initial annotations of shared task and first-person point-of-view videos. We show that AMRs can be interpreted by a proxy language, such as VoxML, as executable annotation structures in order to recreate and simulate a series of annotated events.

1 Introduction

In recent years, there is an increasing interest in dialogue systems that interact with humans in a natural and sophisticated manner. ChatGPT (OpenAI, 2022) and other large language models (LLMs) show a remarkable ability to generate fluent responses to textual prompts. However, these systems lack two key capabilities which are necessary for naturalistic interaction. First, they lack the ability to communicate in multiple modalities beyond written language, including gesture, gaze, and facial expression; LLMs, even ones like GPT-4 that accept both text and image input (OpenAI, 2023), are limited to text output. Second, these models do not have a notion of the "world" as such. They do not track actions and objects in an environment, and therefore are unable to perform *situated grounding* (Pustejovsky and Krishnaswamy, 2021).

Much work has addressed the importance of non-linguistic modalities in communication (Cassell et al., 2000; Wahlster, 2006; Foster, 2007; Kopp and Wachsmuth, 2010; Marshall and Hornecker, 2013; Schaffer and Reithinger, 2019). For example, in a spoken sentence "I used this for the sketch", the referent of the demonstrative "this" is unspecified. In conjunction with a gesture, e.g., pointing to a pencil, however, reference resolution and disambiguation are possible.

Less attention has been paid to the role of action in dialogue interpretation. Actions significantly contribute to the multimodal context within which linguistic utterances are made, and thus play a crucial role in understanding and interpreting dialogue. In the previous example, lifting the pencil can also direct attention to it, which is then linked to the demonstrative. Additionally, actions can also serve as antecedents to speech in VP ellipsis constructions, (e.g., "What did you do that for?" after someone slams a door), and as action-based bridging relations, where actions create links between concepts in a narrative (e.g., "I went to the store today", followed by taking fruit out of a grocery bag). Actions can even be referenced directly by participants, such as the case of a child relaying "My brother said 'thumbs up'!" when given permission to play with a favorite toy.

A major aspect of dialogue interpretation is the *common ground*— shared knowledge and beliefs that interlocutors possess about each other and the world (Clark and Brennan, 1991; Stalnaker, 2002; Tomasello and Carpenter, 2007). Conversations between agents introduce the problem of identifying and modifying the common ground (Tellex et al., 2020). Actions can update the common ground in ways that speech and gesture cannot, by adding, modifying, and deleting items within it.

We argue that, given the importance of actions to multimodal NLU and their direct influence on the common ground, it is essential to consider how they may be integrated with language and other communicative modalities in a shared annotation scheme.

In this paper, we review existing action annotation schemes, as well as Abstract Meaning Representation (AMR) (Banarescu et al., 2013). We then describe initial efforts to use AMR to anno-

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 45–51

tate actions in video data. We explain how action descriptions made with AMR can be translated to the VoxML interpretation language (Pustejovsky and Krishnaswamy, 2016), where they can be executed in a simulated environment, VoxSim (Krishnaswamy and Pustejovsky, 2016), and then close with a discussion of annotation challenges and future work.

2 Background

2.1 Action Annotation

Action recognition in videos is a prominent research area within computer vision, and numerous datasets have been developed providing lexical descriptions of video content, such as Kinetics (Kay et al., 2017) and MSR-VTT (Xu et al., 2016). To facilitate data-driven learning, many of these datasets consist of trimmed clips, categorized with a coarse-grained label describing the action being performed, such as "making pottery" or "bowling".

However, for the purpose of understanding the interplay between action and other communicative acts, we focus on videos that feature discourse between multiple people, and extend over a period of time, thereby allowing for the annotation of fine-grained actions. Although the Charades dataset (Sigurdsson et al., 2016) only involves single individuals, each clip captures a variety of actions through interval-timestamped captions, from which semantic roles can be inferred. The AVA (Gu et al., 2018) and AVA-Kinetics (Li et al., 2020) datasets provide the spatial information of each action associated with multiple people, though their annotations do not adequately assign semantic roles. VidSitu (Sadhu et al., 2021) excels in capturing actions alongside discourse by using movie datasets, introducing semantic role labeling in addition to coreference and event links.

2.2 Abstract Meaning Representation

AMR is a graph-based representation of the meaning of a sentence in terms of its predicate-argument structure (Banarescu et al., 2013). It was designed to be annotatable by humans, and easily parsed by computers. Several extensions have been put forth by the research community (described below), pointing to AMR's utility and expressiveness. For example, the English language sentence "Put that block there.", would be represented in PENMAN (Matthiessen and Bateman, 1991) notation as follows:

```
(p / put-01
  :ARG0 (y / you)
  :ARG1 (b / block
          :mod (t / that))
  :ARG2 (t2 / there)
  :mode imperative)
```

AMR was designed to represent the propositional content of individual written sentences in text. Various extensions to AMR have been proposed which make it more suitable for representing entire documents or dialogues, even using multiple modalities. First, Multi-sentence AMR (MS-AMR) allows AMR to represent meaning beyond the sentence level (O'Gorman et al., 2018). It augments sentence-level AMRs with implicit roles, and marks coreference and bridging relations between entities and events across AMRs.

AMR does not account for a spoken utterance's illocutionary force or effect on the broader dialogue context. Dialogue-AMR (Bonial et al., 2020) extends AMR to include this information in the form of speech act relations, as well as tense and aspect.

Gesture AMR is a further extension of AMR, that goes beyond the linguistic domain, to cover the semantics of gesture (Brutti et al., 2022). Content-bearing gestures are classified according to a taxonomy of gesture acts, and their meaning is annotated similarly to Dialogue-AMR.

Finally, Spatial AMR adds spatial information to AMR, in the form of spatial rolesets, concepts, and frames (Bonn et al., 2020). Of note, Bonn et al. use Spatial AMR to annotate a corpus of Minecraft dialogues, which include both utterances and textual descriptions of actions, such as *[Builder puts down/picks up a red block at X:0 Y:1 Z:0]*.

In addition to wide community adoption, there are several practical reasons for why we propose the annotation of actions with AMR. Every PropBank sense is associated with a single meaning, providing unambiguous interpretations for the labeled actions. PropBank also provides consistent and interpretable argument structures for semantic role labeling. For modeling multimodal dialogue, the efforts described above to capture natural speech and gesture with AMR extensions allow speech and gesture to be seamlessly linked with AMRs of actions using MS-AMR.

3 Approach

To explore the feasibility of applying AMR to actions, we examine two distinct datasets: the Fibonacci Weights Task dataset (Khebour et al., in

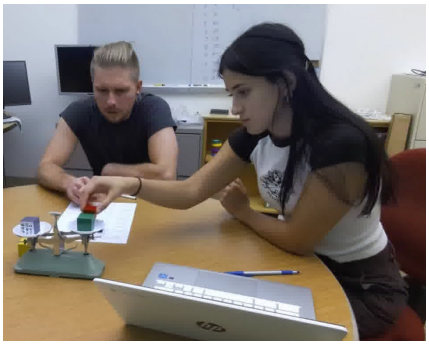


Figure 1: Participant putting a block on a scale.

review), as well as the egocentric Epic Kitchens dataset (Damen et al., 2022). In the examples below, we align observed actions with PropBank senses (Palmer et al., 2003).

3.1 Fibonacci Weights Task

The Weights Task data was designed to elicit teamwork as described in various collaboration frameworks (e.g., PISA (2015); Hesse et al. (2015); Sun et al. (2020)). The task is completed by 2-3 people, and includes blocks, a scale, a worksheet, and a computer with a survey, as seen in Figure 1.

Participants negotiate meaning (and update common ground) via multiple simultaneous modalities. They speak to discuss weights, they gesture to signal the blocks to weigh, and they learn by putting groups of blocks on the scale. The action of putting a block on a scale is annotated as:

```
(p / put-01
  :ARG0 (p1 / participant)
  :ARG1 (b / block)
  :ARG2 (s / scale))
```

Though the actions performed in this dataset are mostly limited to moving and grabbing blocks, they are often prompted by spoken utterances. For instance, an utterance of "let's try this" followed by the action described by the AMR above is an example of a cataphor, where the word *this* refers to the following action. This phenomenon and others like it can be captured by linking AMR arguments with MS-AMR.

3.2 Epic Kitchens

The Epic Kitchens dataset (Damen et al., 2022) consists of spontaneous first-person recordings of individual participants in kitchens, as in Figure 2. Contrasting with the Weights Task dataset, there is little speech in these videos, but a much wider variety of actions that constantly update the common ground for the viewer. Similar to the description of cooking (text) recipes in Tu et al. (2022), the states of the ingredients and tools are updated by each action. Applying AMR to actions in a scenario like this allows for tracking the progress of the recipe and its components.

An example action annotation for the image in Figure 2 is as follows:

```
(t / transfer-01
   :ARG0 (p / participant)
   :ARG1 (v / vegetables)
   :ARG2 (b / bowl)
   :ARG3 (p1 / pot)
   :instrument (c / chopsticks))
```

The AMR of the action registers the objects from the scene as arguments to the *transfer-01* PropBank predicate. As a direct result of actions like this, the vegetables undergo several transformations during the clip - they are combined, boiled, and eventually eaten. Tracking each entity and the changes they undergo is an interesting issue, motivating the following section.

4 Interpretation

4.1 VoxML as an Interpretation Language

The representation of action with AMR as outlined proves useful in modeling its interactions with speech: both the phenomena of VP ellipsis and anaphoric relations that often occur in spoken language can be resolved with MS-AMR cross-modality coreference chains.

However, AMR alone does not describe how actions affect objects in the common ground, such as their ability to update object locations and cause physical transformations. These changes stem from an associated subevent semantics that can be linked with PropBank predicates. For instance, a human executing PropBank *put-01* would involve a grasping and an ungrasping of a given object, with the end result being the object having moved to a new

Figure 2: Participant transferring vegetables from a pot to a bowl with chopsticks.

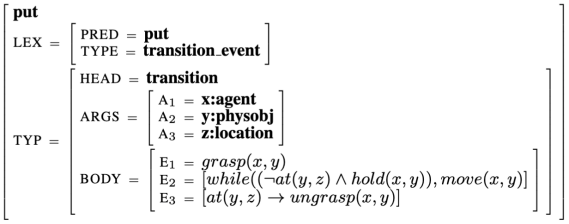


Figure 3: An example VoxML program corresponding to the PropBank predicate *put-01*.

location. These intermediate subevents are equally valid descriptions of a given action in video, and they can be individually referenced by speech, just as top-level actions can be.

We also note that AMR does not address the *lexical aspect* of its predicates - how they progress over time. To annotate the temporal component of an actions in long videos, we traditionally annotate the timestamps or frame numbers according to when the action begins and ends. However, while some actions suggest a continuous process (e.g., *move*), others are instantaneous results (e.g., *hit*), defined only for a single point in time. We can categorize actions by their lexical aspect in a taxonomy, as either states, atelic (without result) processes, or as telic (with result) achievements and accomplishments (Vendler, 1957).

To encode these semantics, we propose the use of a specification language to enrich these annotations with richer lexical semantics, as provided by Generative Lexicon (Pustejovsky, 2013) and VerbNet (Brown et al., 2022). Such information is encoded directly in VoxML (Pustejovsky and Krishnaswamy, 2016), originally designed as a markup language to describe the semantics of 3D simulations. VoxML consists of a library of concepts called the *voxicon*, where agents and objects are represented in entries called *voxemes*, and action predicates are represented in entries called *programs*. A program outlines a verb's lexical type along with its argument and subevent structure, as shown in Figure 3.

This program is classified as a transition event (telic) as opposed to a state or a process, aligning with the lexical aspect of *put-01*; it continues executing until a specific condition has been met, the result subevent. This characterization is reflected in the program's body, outlining a subevent structure involving grasping and moving the object until the object is finally at location *z*.

Voxemes, on the other hand, encode the affordances of objects given the habitats they reside in

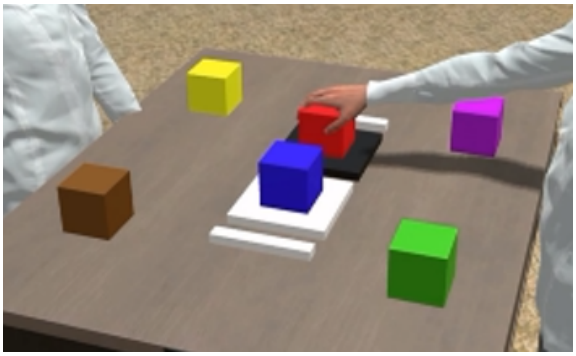


Figure 4: The VoxSim implementation of the Weights Task. At this point in time, two blocks rest on the central scale, one being grasped by a participant.

(e.g., a cup can only be rolled in a certain orientation), as well as geometric information for spatial reasoning. This specification provides insurance that programs are carried out logically, on the correct arguments in the correct situations.

4.2 AMR to Executable Annotation

The information encoded by VoxML allows it to be modelled in a simulated environment called VoxSim (Krishnaswamy and Pustejovsky, 2016), allowing us to capture and track persistent changes to the common ground. Not only can VoxSim simulate the progression of actions over time, it can also continually track the relations of objects to one another and maintain a history of all events. In our simulation of the Weights Task, displayed in Figure 4, VoxSim maintains the relative locations of each block.

To convert AMR to a format usable by VoxSim, we first require all arguments of AMR annotations to be grounded with specific entities labeled in the world. This can be done by linking every entity node to a string representing the object it refers to in the video. We then find the VoxML program entry that corresponds with the AMR's PropBank predicate, aligning its arguments semantically with that predicate's arguments. A concise executable annotation structure like the following example can then be constructed, where `GreenBlock` and `Table` are proper names assigned to entities in the video:

```
put(GreenBlock,on(Table))
```

Through VoxML, this string can be interpreted as an instruction to execute at a specific timestep defined in the annotation.

5 Discussion

We have described an initial exploration of action annotation within the context of communicative acts in dialogue. By investigating the application of AMR and VoxML, we aim for adequate representations to model the interactions between them, as well as define simulations that can track the evolving common ground. This analysis has highlighted certain challenges associated with annotation and possible directions for future work in designing representations.

5.1 Annotation Challenges

We have discussed how high-level actions can be further broken down into subevents, and how their lexical aspect must be respected. This poses multiple questions for annotation in practice.

The first issue is granularity. As illustrated in Figure 3, a *putting* action can be further broken down into its subevent structure, minimally involving a grabbing motion and a holding period. Other actions, like cutting vegetables, consist of a series of instantaneous slicing events. Other events can be easily annotated but may not considerably affect the state of the world, such as someone blinking.

There are multiple ways to describe a set of actions, and this introduces ambiguity to the annotation problem. To ensure consistency, an annotation environment with multiple annotators should agree on a restricted set of atomic predicates to use, with well-defined descriptions of what events constitute each action instance.

The second issue is temporal. As mentioned in our discussion of lexical aspect, different actions require different descriptions of how they progress through time. While processes and accomplishments are defined by an interval of time, achievements are only defined by a single point. Additionally, in contrast with speech, individuals often perform multiple actions simultaneously, such as when they multitask with both hands. This implies multiple overlapping intervals.

Annotation software like ELAN (Brugman and Russel, 2004) can handle simultaneity by placing intervals on multiple tracks. However, interval annotations alone cannot capture instantaneous events, which must either be omitted, or always placed in the context of an accomplishment event.

5.2 Automation of Action Annotation

Though action annotation is a straightforward process given a well-defined set of predicates, manual AMR annotation is more time-consuming. One approach to the automatic annotation of action AMRs involves first identifying actions in videos, then generating AMRs for those actions. Yang et al. (2022) used the VidSitu dataset (Sadhu et al., 2021) to train models to both identify the verbs in the video and fill in their semantic roles. Given a verb and its arguments, the conversion to AMR is straightforward.

Another possible approach is to generate natural language captions for events in the videos, then parse those captions into AMRs. For example, Xu et al. (2023) developed a modular multimodal model that represents the current state-of-the-art on video captioning on the MSR-VTT dataset (Xu et al., 2016). We can then leverage AMR parsers such as Structured mBART with Maximum Bayes Smatch Ensemble distillation (Lee et al., 2022) to convert those captions to the graph-based structure.

6 Conclusion

In this paper, we argue that representing actions is essential for the proper interpretation of situated dialogues. We describe how AMR can be used to annotate actions in different types of video interactions, and describe the challenges associated with this task. We also show how AMRs can be translated to the VoxML specification language to encode semantic information, allowing for the ability to track changes to the common ground in a simulation environment like VoxSim. In future work, we plan to further develop our annotation methodology, and apply it on a larger scale.

Acknowledgments

This research was supported by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805. The opinions expressed are those of the authors and do not represent views of the NSF.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic*

Annotation Workshop and Interoperability with Discourse, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

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.

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.

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

Hennie Brugman and Albert Russel. 2004. Annotating multi-media/multi-modal resources with ELAN. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract Meaning Representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583, Marseille, France. European Language Resources Association.

Justine Cassell, Joseph Sullivan, Elizabeth Churchill, and Scott Prevost. 2000. *Embodied conversational agents*. MIT Press.

Herbert H Clark and Susan E Brennan. 1991. Grounding in communication. *Perspectives on socially shared cognition*, 13(1991):127–149.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2022. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55.

Mary Ellen Foster. 2007. Enhancing human-computer interaction with embodied conversational agents. In *International Conference on Universal Access in Human-Computer Interaction*, pages 828–837. Springer.

Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056.

Friedrich Hesse, Esther Care, Juergen Buder, Kai Sassenberg, and Patrick Griffin. 2015. A framework for teachable collaborative problem solving skills. In *Assessment and teaching of 21st century skills*, pages 37–56. Springer.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Ibrahim Khebour, Richard Brutti, Indrani Dey, Rachel Dickler, Kelsey Sikes, Kenneth Lai, Mariah Bradford, Brittany Cates, Paige Hansen, Changsoo Jung, Brett Wisniewski, C Terpstra, Leanne Hirshfield, Sadhana Puntambekar, Nathaniel Blanchard, James Pustejovsky, and Nikhil Krishnaswamy. in review. When text and speech are not enough: Modeling meaning in situated collaborative tasks.

Stefan Kopp and Ipke Wachsmuth. 2010. *Gesture in embodied communication and human-computer interaction*, volume 5934. Springer.

Nikhil Krishnaswamy and James Pustejovsky. 2016. Voxsim: A visual platform for modeling motion language. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 54–58.

Young-Suk Lee, Ramón Astudillo, Hoang Thanh Lam, Tahira Naseem, Radu Florian, and Salim Roukos. 2022. Maximum Bayes Smatch ensemble distillation for AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5379–5392, Seattle, United States. Association for Computational Linguistics.

Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. 2020. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*.

Paul Marshall and Eva Hornecker. 2013. Theories of embodiment in HCI. *The SAGE Handbook of Digital Technology Research*, 1:144–158.

Christian Matthiessen and John A. Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Burns & Oates.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt.

OpenAI. 2023. GPT-4 technical report. arXiv.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. Amr beyond the sentence: the multi-sentence amr corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2003. The proposition bank: an annotated corpus of semantic roles. *Computational Linguistics*.

OECD PISA. 2015. Assessment and analytical framework: Science. *Reading, Mathematics and Financial Literacy, PISA*.

James Pustejovsky. 2013. Dynamic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 1–10.

James Pustejovsky and Nikhil Krishnaswamy. 2016. Voxml: A visualization modeling language. *arXiv preprint arXiv:1610.01508*.

James Pustejovsky and Nikhil Krishnaswamy. 2021. Embodied human computer interaction. *KI-Künstliche Intelligenz*, 35(3):307–327.

Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ram Nevatia, and Aniruddha Kembhavi. 2021. Visual semantic role labeling for video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5600.

Stefan Schaffer and Norbert Reithinger. 2019. Conversation is multimodal: thus conversational user interfaces should be as well. In *Proceedings of the 1st International Conference on Conversational User Interfaces*, pages 1–3.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer.

Robert Stalnaker. 2002. Common ground. *Linguistics and Philosophy*, 25(5-6):701–721.

Chen Sun, Valerie J Shute, Angela Stewart, Jade Yonehiro, Nicholas Duran, and Sidney D'Mello. 2020. Towards a generalized competency model of collaborative problem solving. *Computers & Education*, 143:103672.

Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. 2020. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:25–55.

Michael Tomasello and Malinda Carpenter. 2007. Shared intentionality. *Developmental Science*, 10(1):121–125.

Jingxuan Tu, Eben Holderness, Marco Maru, Simone Conia, Kyeongmin Rim, Kelley Lynch, Richard Brutti, Roberto Navigli, and James Pustejovsky. 2022. Semeval-2022 task 9: R2vq–competence-based multimodal question answering. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1244–1255.

Zeno Vendler. 1957. Verbs and times. *The Philosophical Review*, pages 143–160.

Wolfgang Wahlster. 2006. Dialogue systems go multimodal: The SmartKom experience. In *SmartKom: foundations of multimodal dialogue systems*, pages 3–27. Springer.

Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, Guohai Xu, Ji Zhang, Songfang Huang, Fei Huang, and Jingren Zhou. 2023. mPLUG-2: A modularized multi-modal foundation model across text, image and video. arXiv.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Guang Yang, Manling Li, Jiajie Zhang, Xudong Lin, Shih-Fu Chang, and Heng Ji. 2022. Video event extraction via tracking visual states of arguments.

From Sentence to Action: Splitting AMR Graphs for Recipe Instructions

Katharina Stein[1] **Lucia Donatelli**[2] **Alexander Koller**[1]
[1] Department of Language Science and Technology
Saarland Informatics Campus
Saarland University, Saarbrücken, Germany
[2] Vrije Universiteit Amsterdam
De Boelelaan 1105, 1081 HV Amsterdam, Netherlands
kstein@lst.uni-saarland.de

Abstract

Accurately interpreting the relationships between actions in a recipe text is essential to successful recipe completion. We explore using Abstract Meaning Representation (AMR) to represent recipe instructions, abstracting away from syntax and sentence structure that may order recipe actions in arbitrary ways. We present an algorithm to split sentence-level AMRs into action-level AMRs for individual cooking steps. Our approach provides an automatic way to derive fine-grained AMR representations of actions in cooking recipes and can be a useful tool for downstream, instructional tasks.

1 Introduction

Procedural texts are special kinds of text that serve the purpose of guiding humans through the steps required to accomplish a specific task. Recipe texts are an idiosyncratic kind of procedural texts whose successful execution depends on accurately interpreting which actions need to be carried out in which order and which ingredients and tools are involved in each step. For example, the instruction *"**Turn** the dough **out** onto a surface **dusted** with flour and **knead** briefly until smooth."* presents three actions: (i) dusting a surface; (ii) placing the dough on that surface; and (iii) kneading the dough until smooth. In recipe texts, actions often depend on other actions but there is often flexibility with respect to the overall order in which actions are instructed as some actions can take place in parallel or can be carried out at different stages of the cooking process. For example, dusting the surface could be instructed before preparing the dough.

Recipe texts frequently combine several actions in one sentence and often there are no uniform methods for putting specific actions into the same instruction; different versions of the same recipe may even differ in how equivalent or parallel actions are distributed across sentences.

Tasks such as adapting a recipe to a specific situation or presenting a recipe interactively to a user require the generation of a coherent recipe text that presents actions in a potentially different order and combination. To flexibly generate new versions of a recipe that present actions in an adapted order it is necessary to correctly decompose the recipe into the individual actions.

Previous work on recipe texts proposed identifying cooking actions and objects in recipes and representing the dependency relations between them in domain-specific graph representations with nodes for actions, ingredients and tools (Mori et al., 2014a; Yamakata et al., 2020). These representations are attractive for fine-grained analysis of recipe texts but lack the expressivity to represent details such as adverbs or relations such as conditions and alternatives. Yet, this kind of information is important for correctly reconstructing the original meaning when generating instructions.

We explore generating recipe instructions at the action level from Abstract Meaning Representation (AMR) graphs. AMR is able to represent fine-grained and rich semantic relations, and its focus on predicate-argument structure makes it attractive for representing cooking instructions. Yet, AMR is a sentence-level representation that represents individual sentences in individual graphs.

This paper addresses the challenge of splitting sentence-level AMR graphs into the individual action-level AMRs. We present a splitting algorithm that considers the semantic relationships between actions in recipe instructions (§3)[1]. We evaluate our approach in a direct manual evaluation of the action-level representations as well as in an automatic and human evaluation of recipes generated from the created representation (§4). Findings

[1]Code and documentation is available at https://github.com/interactive-cookbook/recipe-generation

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 52–67
June, 2023.

show that our algorithm accurately identifies action-level subgraphs in AMR recipe representations, suggesting its utility for AMR representations of procedural texts and for generating action-level instructions.

2 Related Work

At first glance, recipe texts may look quite simple. Yet recipe texts are semantically quite complex: subjects of actions are implicit, often as a result of being written in imperative mood; anaphoric expressions often refer to intermediate products that are outputs of actions and only partially corefer to input ingredients; and zero anaphora objects are frequent. To model recipe structure, most work on recipe text involves the identification and tagging of cooking actions, ingredients, intermediate substances and tools which get then used to create a structured representation (Mori et al., 2014a; Yamakata et al., 2020; Donatelli et al., 2021; Liu et al., 2022, *inter alia*). The most common representation approaches are graph and tree structures, which represent the flow and dependencies of actions and involved entities (Mori et al., 2014a; Jermsurawong and Habash, 2015; Kiddon et al., 2015; Yamakata et al., 2016, 2020; Donatelli et al., 2021).

The Abstract Meaning Representation (AMR) (Banarescu et al., 2013) framework represents the meaning of sentences with a focus on predicate-argument relationships, a key component of recipe structure. Figure 1a shows the AMR for the instruction *"Turn the dough out onto a surface dusted with flour and knead briefly until smooth."* on the left. As shown, AMRs are rooted, directed graphs in which nodes correspond to concepts and edges to the semantic relations between concepts. The framework makes use of a rich set of node and edge labels including frames from PropBank (Palmer et al., 2005), and within-sentence coreference is represented by re-entrancy.

Previous work on adapting AMR to the non-sentence level proposes approaches to create a multi-sentence AMR representation (O'Gorman et al., 2018; Naseem et al., 2022) or dialogue AMR graphs (Bai et al., 2021) but essentially keeps the sentence-level AMRs as part of the extended representations. AMR has also been used in the tasks of summarization (Liu et al., 2015; Lee et al., 2021) and text style transfer (Jangra et al., 2022).

3 Creating Action-Event AMRs

3.1 Actions and Action Events

Recipe texts should enable a cook to successfully prepare a dish by guiding them through the basic steps of the process. Yet, recipes rely on much commonsense knowledge for accurate interpretation, often combining several actions in one sentence or making only implicit reference to required actions. For example, *"**Turn** the dough **out** on a surface **dusted** with flour and **knead** until smooth."* conjoins the two steps of placing the dough on a surface and kneading it, mentioning the dusting of the surface only implicitly.

In previous work, the term *action* has been used in various ways. In some work, it refers to the action predicate together with its arguments (e.g. Kiddon et al., 2015; Liu et al., 2022). Often, only the action predicates themselves are referred to as an action (e.g. Mori et al., 2014b; Chang et al., 2018; Yamakata et al., 2020; Donatelli et al., 2021; Sakib et al., 2021). Trained taggers then identify corresponding spans of action predicate tokens.

To differentiate the two concepts, we use the term **action** to refer to an action predicate from here on and we introduce the concept of an **action event** to refer to an action predicate and all information belonging to it. In particular, we define an action event of a recipe as an individual action to be carried out by the cook together with all information (ingredients, time, result state, etc.) relevant to successfully complete the action. Importantly, not all actions in a sentence belong to different action events under this definition as we illustrate with examples from the action-tagged recipe corpus from Donatelli et al. (2021) shown in Table 1[2]. To distinguish actions and action events, we make use of the predicate-argument based structure of AMR.

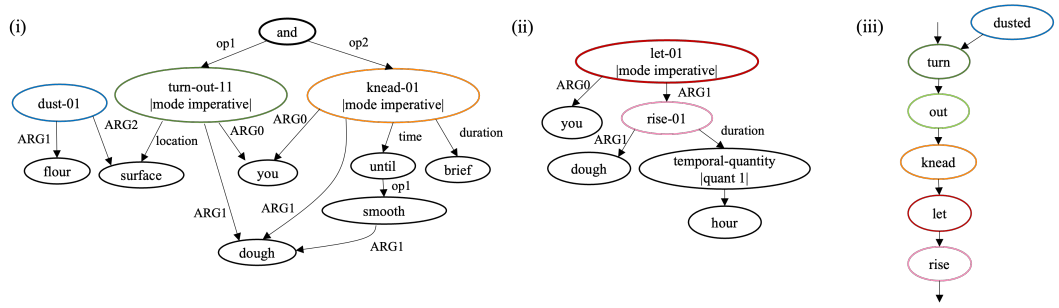
3.2 From Sentences to Action Events

AMR parsers typically predict one graph per sentence. Figure 1a presents two successive recipe instructions with the tagged actions shown in color; the corresponding **sentence-level AMRs** (**S-AMRs**) (i) and (ii); and a part of the action graph for the recipe (iii). The **action graph** consists of one node for each action and the edges represent their dependencies (Donatelli et al., 2021). Each of the two S-AMRs includes nodes corresponding to different actions. For both action and AMR graphs,

[2]Examples presented are shortened or slightly modified.

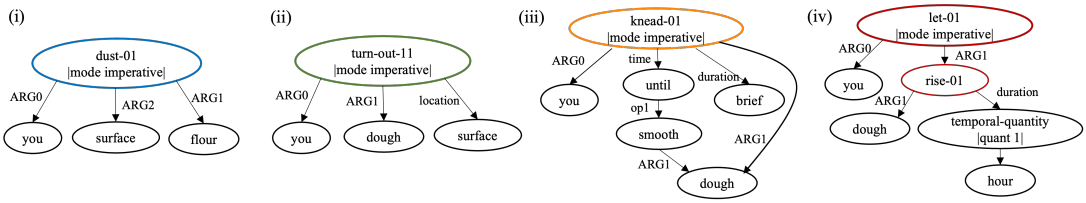
(1)	**Place** **cooked** chicken on paper towel to **drain** the oil.
(2)	**Stir** in the chocolate chips by hand **using** a wooden spoon.
(3)	**Bake** for 30 minutes, or until a toothpick **comes out** clean.
(4)	Gradually **add** the water, while **mixing**.
(5)	**Let** the loafs **cool** for 10 minutes before **turning** onto a wire rack.
(6)	**Divide** the batter evenly among the mini loaf pans or **pour** into large loaf pan.
(7)	If it is still a little bit lumpy, you can **add** a touch of heavy cream, and **blend** again.

Table 1: Examples of multi-action recipe instructions. Actions in the same color belong to the same action event.



Turn the dough **out** onto a surface **dusted** with flour and **knead** briefly until smooth. **Let** the dough **rise** for 1 hour.

(a) The S-AMRs for "Place the dough onto a surface dusted with flour and knead briefly until smooth." (i), "Let the dough rise for 1 hour." (ii) and a part of the action graph of the recipe to which the instructions belong (iii).



Dust a surface with flour. **Turn** dough out onto the surface. **Knead** dough briefly until smooth. **Let** the dough **rise** for 1 hour.

(b) The A-AMR graphs for the four action events "dust" (i), "turn out" (ii), "knead" (iii) and "let rise" (iv).

Figure 1: The different of graph representations we build upon in our work (1a) and the graphs we produce (1b).

we say that a node is **aligned** to the action (predicate) it corresponds to. Additionally, we say that an AMR graph is aligned to an action event if at least one node in the AMR is aligned to an action belonging to the action event.

Our goal is to split the S-AMRs into individual AMRs for action events (**A-AMRs**) such that each A-AMR includes exactly the actions from that event and all other nodes that belong to the action event, as shown in Figure 1b. Figure 1 illustrates several key aspects of this process. First, action aligned AMR nodes that belong to the same action event, such as `let-01` and `rise-01`, are always kept together. Second, each A-AMR contains only action nodes from a single action event. Different actions may share arguments as they operate on the same substances or tools, e.g. dough is the direct object of both "*turn out*" and "*knead*" and belongs to both action events. Our algorithm does not split an S-AMR into A-AMRs consisting of disjoint sets of nodes but selects the subgraph of the S-AMR that consists of nodes and edges belonging to the action event. Finally, the action graph allows us to properly order action events.

3.3 Datasets

As our main dataset, we use the ARA1.1 corpus[3] (Donatelli et al., 2021) which provides action graphs for 110 recipes spanning 10 dishes of the recipe corpus from Lin et al. (2020). We exclude three recipes and refer to the set of the remaining recipes as **ARA1**. We use an additional set of 110 recipes spanning 10 dishes as a secondary dataset (**ARA2**) to refine and validate our approach and use the tagger and parser from Donatelli et al. (2021)[4] to obtain the action tags and action graphs we use.

[3] https://github.com/interactive-cookbook/ara

[4] https://github.com/interactive-cookbook/tagger-parser

	Label node1	LPath	Label node2
1)	*action*	⟨ ARGX (opX)[1] ⟩	*action*
2)	*action*	⟨ direction (opX)[1] ⟩	*action*
3)	*action*	⟨ (*edge*)* *relation* (*edge*)* ⟩	*action*
	where *relation* is equal to purpose, manner, instrument, time or duration		
4)	*action*	⟨ opX, opX-of ⟩	*action*
5)	*action*	⟨ ARGX, ARGX-of ⟩	*action*

Table 2: Path patterns between two action-aligned AMR nodes that should be clustered together. *action* and *edge* can be any node or edge label, round brackets are used for optional labels on the **LPath**, ()[1] meaning zero or exactly one occurrence and ()* allowing any number of occurrences.

The full list of dishes and exclusion criteria can be found in Appendix A.

For our approach we rely on the availability of node-to-token alignments to determine to which action events each AMR is aligned. We obtain the S-AMRs for the recipes by parsing each sentence using the transition-based AMR parser of Drozdov et al. (2022)[5] (henceforth StructBART). The parser includes a neural aligner to predict node-to-token alignments needed for training and produces the alignments as a by-product during parsing.

3.4 Splitting Algorithm

Obtaining A-AMRs from S-AMRs consists of two main steps: (i) deciding which action-aligned AMR nodes correspond to the same action event and (ii) creating one A-AMR per action event from the S-AMR, i.e. extracting the appropriate subgraph. We focus on the overall process and the main decision rules in this section. The full set of clustering and splitting rules can be found in Appendix C.

Both steps of the process are based on the concepts of **labelled paths** (**LPath**) and **meeting nodes**. We define a path between two nodes u and v as a sequence of edges between two nodes where edges can be traversed in either direction and each node is visited only once. A labelled path is the sequence of the labels of edges of a path where edges that are traversed in reverse direction receive their reverse role label. For example, the LPath between dust-01 and turn-out-11 in the left AMR (i) in Figure 1a is ⟨ARG2, location-of⟩. We then define a meeting node as a node on a path at which two successive edges change their direction, i.e. where one edge is traversed in its original direction and the next edge in reverse direction or the other way round. On the path between dust-01 and and turn-out-11, there is one meeting node: surface.

The label of an edge between two action nodes represents the relation between them. LPaths allow us to capture relations between two action nodes that are further away, which we use to decide if two actions belong to the same action event. Meeting nodes are shared predecessor or successor nodes of two action nodes and intuitively correspond to nodes that belong to both action events such as shared arguments or conjunctions.

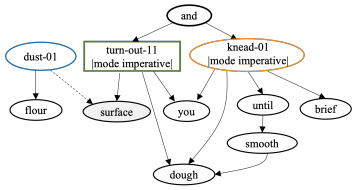
3.4.1 Clustering

Let $\mathbf{M_i}$ be the S-AMR for the i-th instruction in a recipe and let $\mathcal{A}$ be the set of all nodes of $\mathbf{M_i}$ aligned to different actions[6]. The clustering step groups all nodes $a \in \mathcal{A}$ into disjoint action clusters $\langle C_1, ... \rangle$ such that actions from the same action event are in the same cluster. It starts by creating all possible pairings of nodes from $\mathcal{A}$. Then for each pair (a_i, a_j) all possible paths and LPaths between the nodes get computed and checked against a set of rules. Table 2 lists the main patterns used for the clustering rules: if the two action nodes a_i, a_j and one of their LPaths match any of the patterns 1) - 3), a_i and a_j are clustered together. The patterns match the ways in which AMR represents the relations between actions of the same event. For example, 1) covers cases with discontinuous action spans and 3) can capture complex relations such as time specifications, even in nested structures.

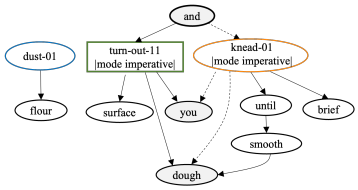
If pattern 4) or 5) matches, we check whether the meeting node is labelled or, slash or contrast-01 to make sure conjoined actions are not clustered together. Larger action clusters are built such that for each clustered pair (a_i, a_j) a_i and a_j end up in the same final cluster C_n. If an S-AMR has or, slash, possible-01 or have-condition-91 as the root node, $\mathcal{A}$ is treated as a single action cluster.

[5] amr3.0-structured-bart-large-neur-al-sampling5-seed42 from https://github.com/IBM/transition-amr-parser

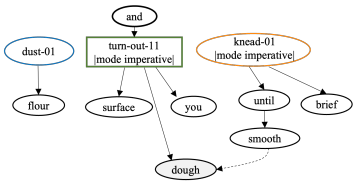
[6] If more than one AMR node is aligned to the same action we ignore the ones not labelled with a predicate frame.



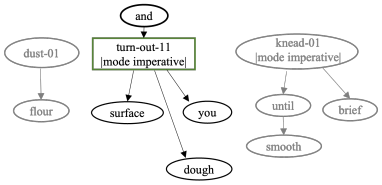
(a) Creating A-AMR for `turn-out` Step 1



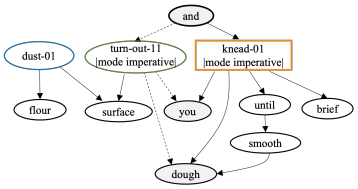
(b) Creating A-AMR for `turn-out` Step 2



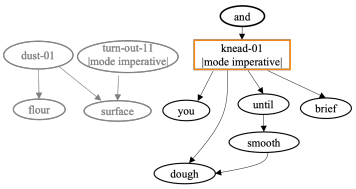
(c) Creating A-AMR for `turn-out` Step 3



(d) Creating A-AMR for `turn-out` Step 4



(e) Creating A-AMR for `knead` Step 1



(f) Creating A-AMR for `knead` Step 2

Figure 2: Step-by-step example of obtaining the A-AMRs from the S-AMR in Figure 1. Nodes from the current target cluster are shown as rectangles and edges that get removed as dotted edges. Creating the A-AMR for `dust` and the final postprocessing step are not shown. Edge labels are left out.

3.4.2 Splitting

If an S-AMR graph is aligned to more than one action cluster, the splitting algorithm is applied with the goal to obtain one A-AMR per cluster. Similar to the clustering approach, the splitting algorithm is based on the paths between action-aligned AMR nodes and meeting nodes.

The splitting algorithm always operates on one S-AMR and one target action cluster. The A-AMR gets created by iteratively removing nodes or edges until deriving one connected subgraph that contains all action nodes from the target cluster and no action nodes from any other clusters. Figure 3 presents the main structure of the algorithm: it starts by pairing each node from the target action cluster with all other action nodes, i.e. the pairs of all nodes that should not be connected anymore in the end. All paths from nodes of the target cluster to another cluster are considered for removing an edge or a node in order to separate the actions from each other. The shortest paths are considered first as they are usually the more meaningful paths that are captured by the removal conditions of the algorithm. The main rule checks for paths that consist of exactly one direction change, i.e. include one meeting node. If a path fulfills this condition then the edge "behind" the meeting node gets removed.

Figure 2 illustrates the removal steps applied to derive the three A-AMRs (i-iii) from Figure 1b from the left S-AMR (i) in Figure 1a. The S-AMR includes three nodes aligned to an action

1. create a set $\mathcal{Q}$ of node pairs by pairing each action node from C_j with each node from the other clusters

repeat

2. create a sequence $\mathcal{P}$ of all paths for all pairs in $\mathcal{Q}$, ordered by length in ascending order
3. **if** $\mathcal{P} = \emptyset$ then **break**
4. **for** p in $\mathcal{P}$ **do**
   (a) **if** CONDITION **do** REMOVAL and **continue** from 2.
5. **if** the graph does not change anymore **then return** original graph

end repeat

6. select the connected subgraph that includes the nodes from the current cluster C_j as the A-AMR for that action event

Figure 3: The main structure of the splitting algorithm given an S-AMR and a target action cluster C_j

node and the clustering results in three action clusters. Starting with the A-AMR for `turn-out`, one of the shortest paths connects `turn-out` and `dust` and consists of exactly one direction change at the shared child node `surface`. Therefore, the edge between (`surface`, `dust`) gets removed from the graph, as illustrated in Figure 2a. After removing an edge, the algorithm recomputes the set of all paths for the modified AMR.

There are still paths left in $\mathcal{P}$, so the splitting continues. Figure 2b and 2c summarize the next four

iterations, in which the connecting paths between `turn-out` and `knead` get broken up. When no path between `turn-out` and any other action node is left then the connected component of the graph including that action gets selected as the A-AMR (see 2d).

The algorithm continues with the action cluster for `knead`. Removing the edges from the three shortest paths between `knead` and `turn-out` gets rid of all connecting paths resulting in the subgraph for `knead` (see 2f). Lastly, the A-AMR for `dust` gets created in the same way. A postprocessing step transforms the subgraphs into the final A-AMRs as shown in Figure 1b by removing redundant nodes and representing actions that originally were participles such as `dust` as imperatives.

An important characteristic of the algorithm is that it ensures that no nodes from the original S-AMR get lost except if removed by the rules themselves. If the splitting results in A-AMRs that do not cover all nodes from the original S-AMR, the S-AMR gets treated as non-separable. The same holds if any of the action clusters cannot be separated from all other clusters.

4 Evaluation

4.1 Manual Evaluation

We apply the splitting approach to the S-AMR graphs of the ARA1 and the ARA2 recipes. Table 3 provides an overview over the original datasets as well as the output of the splitting algorithm. The dataset of A-AMRs created by the splitting approach consists of 1396 AMR graphs for the ARA1 recipes out of which 584 A-AMRs are equivalent to the original S-AMR. For the ARA2 dataset the splitting results in a total of 1471 A-AMRs out of which 648 get not modified. Few S-AMRs cannot be separated into individual A-AMRs by our algorithm, proving its effectiveness.

To evaluate the splitting algorithm, we manually compare all original S-AMRs to the generated A-AMRs. In the ARA1 dataset we identified 64 A-AMRs that were incorrect relative to the source S-AMRs: either they were split incorrectly or not split although they should have been.[7] For 46 out of the 64 incorrect A-AMRs, the initial mistake already happens before the splitting process, i.e. in the action tagging step or during AMR parsing. In the ARA2 dataset, there are 68 incorrect A-AMRs and for 58 of them the source mistake happens before the splitting step. We also identified cases for which the decision how to split the S-AMR is not straightforward. These cases will be discussed together with the limitations of the algorithm in Section 5.

	ARA1	ARA2
Recipes / action graphs	107	110
Action nodes	1583	1771
Sentences / S-AMRs	941	1001
Action clusters	1391	1473
A-AMRs	1396	1471
Non-separable S-AMRs	14	13
Incorrect A-AMRs	64	68

Table 3: Overview of the ARA1 and ARA2 datasets (upper part) and the results from applying the splitting algorithm (lower part).

4.2 NLG-based Evaluation

In addition to evaluating the splitting approach based on the output graphs themselves, we conduct a task-related evaluation. A potential use case for the fine-grained A-AMR graphs is the generation of recipe instructions at the action-event level in order to recombine them flexibly or present them incrementally to a user, e.g. to guide a user through the cooking process step by step in real-time. Therefore, we generate recipe instructions from the A-AMRs and evaluate them both automatically and manually with crowdsourced human evaluation.

To obtain gold instructions for the individual A-AMRs we use a rule-based heuristic. Another approach to obtain instructions corresponding to the A-AMRs would be to use an AMR-to-text model. However, as AMR parsers, AMR-to-text models are usually trained on the AMR3.0 corpus[8]. Therefore, the sentences generated by them for the A-AMRs might not resemble the style of recipe instructions (see §5). Splitting the instructions heuristically gives us a dataset on which we can fine-tune an AMR-to-text model for the recipe domain. Additionally, we can use the data to automatically evaluate and compare different models.

Our extraction heuristic is based on the node-to-token alignments produced by the parser and creates the gold instructions by selecting all tokens from the original instruction to which nodes in the specific A-AMR are aligned. Additionally, we use a set of rules based on POS tags to decide about the selection of unaligned tokens and to reorder

[7] We evaluate the "correctness" of the A-AMR given the S-AMR predicted by the StructBART parser.

[8] https://catalog.ldc.upenn.edu/LDC2020T02

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEURT
amrlib	44.00	74.9	45.96	68.88	72.38	38.28
GART5-0	56.15 ±1.2	84.63 ±0.5	62.88 ±1.3	80.28 ±0.7	82.09 ±0.7	59.42 ±1.8
GART5-1	**57.84** ±1.3	**85.75** ±0.4	**65.98** ±0.9	**81.91** ±0.7	**83.47** ±0.6	**62.64** ±1.5

Table 4: Results on the ARA1 test split averaged over 6 different seeds (± standard deviation).

	Grammar	Fluency	Verbosity	Structure	Success	Overall
Dependency	2.88	2.58	3.26	3.33	3.40	2.76
GART5-0	3.80	3.27	3.45	3.47	3.31	3.022
GART5-1	3.62	3.02	3.15	3.20	3.01	2.74
Original	5.25	5.13	5.30	5.28	5.14	5.06

Table 5: Mean evaluation scores from the human evaluation per rating criterion for the recipes generated with context (GART5-1), without context (GART5-0) and by the dependency baseline and for the original recipe.

the selected tokens to improve the grammar (see Appendix A.1 for an example). Participle actions get stemmed to their imperative form. Overall, 812 and 823 of the A-AMRs of the ARA1 and ARA2 recipes get a new instruction out of which around 85% (ARA1) and 80% (ARA2) are grammatical.

4.2.1 Generation Set-up

For the generation, we use the AMR-to-text model from the amrlib library[9] (amrlib model from here on), a pre-trained T5 model fine-tuned on the AMR3.0 corpus for AMR-to-text generation. We further fine-tune the amrlib model on the A-AMR dataset for the ARA1 recipes which we split into training (86), validation (11) and test (10) recipes.

Instead of passing a single linearized AMR graph we prepend the previous sentence as context information to the linearized AMR. For each recipe, we order the AMR-instruction pairs similarly to the instructions in the original recipe. A-AMRs obtained from the same S-AMR get ordered relative to each other based on the action graph such that e.g. *"Dust a surface with flour."* comes before *"Turn dough out onto the surface."*.

The amrlib model then gets fine-tuned to predict the sentence for the AMR-graph based on the AMR-graph and the context. Details about fine-tuning can be found in Appendix A. We call our generation model **GART5** (**G**enerating **A**ction-level **R**ecipes based on **T5**)[10].

4.2.2 Automatic Evaluation

For the automatic evaluation, each A-AMR gets paired with the previous sentence from the original recipe as context. We then fine-tune our model on the graph-context pairs from the train recipes (GART5-1) and compare the results on the test recipes to two baselines: the texts generated by the original amrlib model[11] and instructions generated by a model fine-tuned on the recipe dataset without context (GART5-0). Additional ablation experiments can be found in Appendix A.2.

Table 4 presents the results of the automatic evaluation. Our GART5-0 model without context performs considerably better than the amrlib model on the A-AMR ARA1 test split across all metrics, achieving an improvement of 12 points in BLEU score and even 21 BLEURT points. Adding context in the fine-tuning step results in an additional - but smaller - improvement across all metrics.

4.2.3 Crowd-sourcing Evaluation

In addition to the automatic evaluation, we conduct a human evaluation to get a more thorough and reliable assessment of the quality of the generated texts. 88 participants recruited via Prolific[12] judged various measures of coherence and acceptability for the generated instructions. Participants were paid £2.25 for their on average 15-minute participation.

We included each recipe from the test split in four versions. One version was generated with the GART5-0 and and one with the GART5-1 model, where the sentence generated at the previous time step was passed as context. As baseline recipes, we create action-level instructions from the original instructions by splitting them based on syntactic dependencies. Additionally, we include the original recipes as an upper bound. In the original condition, the instructions of each recipe were presented in

[9] https://github.com/bjascob/amrlib-models/releases/model_generate_t5wtense-v0_1_0

[10] Code for the training is available at https://github.com/interactive-cookbook/recipe-generation-model

[11] We remove the node-to-token alignments from the input to reproduce the format the amrlib model was trained on.

[12] https://www.prolific.co/

Dependency	GART5-0	GART5-1
Pour over the flour mixture. And very gently **stir**. Until about **combin**.	**Pour** in flour mixture. **Stir** very gently until about combined.	**Pour** in flour mixture. **Stir** very gently until about combined.
Melt butter. **Stir** in the butter. And **continue mixing** very gently until combined.	**Melt** butter. **Stir** in butter. **Continue mixing** very gently until combined.	**Melt** butter. **Stir** butter. **Continue to mix** very gently until combined.
Beat egg whites until stiff. **Preheat** iron. And slowly **fold** into batter.	**Beat** egg whites until stiff. **Preheat** waffle iron. Slowly **fold** in egg whites into batter.	**Beat** in egg whites until stiff. **Preheat** waffle iron. Slowly **fold** in egg whites into the batter.
Spoon the batter into waffle iron in batches. And **cook** according to its directions.	**Save** batter in batches on waffle iron. **Cook** batter according to directions.	**Scoop** the batter in batches onto waffle irons. **Cook** the batter according to its directions.

Original
Pour over the flour mixture and very gently **stir** until about **combined**. **Stir** in the **melted** butter and **continue mixing** very gently until combined. **Beat** egg whites until stiff and slowly **fold** into batter. **Spoon** the batter into preheated waffle iron in batches and **cook** according to its directions.

Table 6: An excerpt from a recipe for waffles in the four versions that were included in the crowd-sourcing evaluation.

their original order. In the other conditions, the order of the generated instructions was determined by traversing the corresponding action graph using a heuristic (see Appendix B).

Participants were presented two recipes per condition and they rated the textual quality of each recipe along six criteria on a six-point Likert Scale. Table 5 presents the results of the evaluation[13]. The original human written recipes were rated significantly better than the recipes from all other conditions for each rating criterion. Against our expectations and in contrast to the results of the automatic evaluation, we find that recipes generated with or without context were not rated significantly different with respect to their grammar, fluency and structure, but the recipes without context were rated significantly better with respect to their verbosity, success and overall quality. The grammar and fluency was rated worst in the dependency baseline.

5 Discussion

In this section we discuss the performance of our splitting algorithm and the results of the generation experiments in more detail.

Splitting algorithm. As described in §4.1, the splitting approach can successfully separate the S-AMRs of almost all instructions in the ARA1 and ARA2 recipes into A-AMRs. The iterative approach of removing edges at meeting nodes allows to split even deep and nested S-AMRs for long instructions successfully. For example, one of the instructions with the highest number of action events, *"**Remove** from oven and **let cool** on wire rack for about 10 minutes before **turning** bread **out** onto wire rack and **letting cool** completely before **slicing**, **toasting**, and **devouring**."* gets correctly separated into seven A-AMRs.

Many of the A-AMRs that are incorrectly split are based on a wrong S-AMR. We found that often the same tokens or specific types of tokens lead to parsing mistakes and that these tokens are mostly specific to the recipe domain (e.g. "grease", "Parmesan", "knead"). These observations are in line with the findings from Bai et al. (2021) that the main challenge for out-of-domain AMR parsing is the correct prediction of concepts. Additionally, when ignoring splitting mistakes resulting from parsing mistakes, almost all incorrect A-AMRs contain one of the following concepts: `mean-01`, `have-degree-91` and `have-quant-91`. These concepts are used to represent complex relations and they introduce path patterns into the AMR that are quite different and not covered by our algorithm.

Finally, during the manual evaluation of the A-AMRs we encountered a number graphs for which it is not straightforward to decide whether the specific splitting is adequate because of the specific semantic characteristics and especially temporal interactions of the actions. For example, *"**Bring** a*

[13]Statistical significance testing was performed using the software R (R Core Team, 2021) and the $lme4$ package (Bates et al., 2015). We used linear mixed effect models with condition as fixed effect, and by-subject and by-item intercepts and slopes as random effects, $p < 0.05$.

*pot of **salted** water to a boil."* gets split into *"**Salt** water."* and *"**Bring** a pot of water to a **boil**."*. However, none of the two potential orderings of the instructions is entirely adequate. On the one hand, the salt should be added before boiling the water. On the other hand, the water cannot be salted before it is filled into a pot but the "filling" action is only implicitly included in the original instruction.

Generated texts. In the automatic evaluation, fine-tuning on our recipe AMR dataset resulted in a considerable improvement compared to generating the instructions with the pretrained amrlib model. We found that the amrlib model struggles to produce the recipe specific writing style. For example, the amrlib model generates *"Stir for a commingling"* where the GART5 models generate *"Stir to combine"*.

In contrast to the automatic evaluation, we found that in the human evaluation the recipes generated with context were not judged significantly better. Table 6 shows an excerpt of a recipe for waffles in the four versions rated in human evaluation. Overall, the instructions generated by the two GART5 models are very similar. In our opinion, the most likely explanation for the different results is that the higher automatic evaluation scores are artifacts of the reference-based score computation and do not reflect real differences in quality.

The performance of the AMR parser also affected the quality of the generated texts as wrong concepts in the AMR lead to inadequate or nonsensical instructions. For example, representing "spoon" in the last instruction of the original version with `save-01` resulted in a wrong instruction generated by GART5-0.

General discussion. Our findings suggest that AMR representations are promising for representing and generating recipe instructions at the action level. The focus on predicate-argument structure makes them attractive for the representations of instructions as they center around actions and objects required to carry them out. Additionally, AMR graphs provide rich and fine-grained information about the semantic relations, the dependencies and also within-sentence coreference which makes it possible to identify the individual action events and to split even S-AMRs for long and nested instructions into their A-AMR components.

Furthermore, our approach produces again rich representations of the action events from which instructions for the individual action events can be generated. Heuristically splitting the textual instructions instead of the AMR representations would require a combination of different tools to predict all the relevant information such as dependencies and semantic roles. Additionally, splitting the instructions at the text level using our dependency baseline more often resulted in ungrammatical sentences as reflected by the significantly higher grammar and fluency ratings for the texts generated from the A-AMRs compared to the baseline.

6 Conclusion & Future Work

We have presented an approach to split sentence-level AMR representations for cooking recipe instructions into more fine-grained AMR representations of the individual action events. Our rule-based algorithm provides an automatic way to identify which cooking actions in a recipe instruction constitute separate action events to be carried out and to systematically breaking up the sentence-level AMRs into representations of the action events that provide more concise instructions. The predicate-argument oriented structure of AMR facilitates this process, and our approach achieves high performance on accurately breaking up the S-AMRs to more concise representations that can be used to generate instructions.

One bottleneck of our approach is the performance of the AMR parser in the domain of cooking recipes. Future work might investigate adapting AMR parsers to out-of-domain recipe vocabulary and processes. Regardless, representations of action events can support analysis and comparisons of actions in different cooking recipes as well as instruction generation in tasks that require more flexibility with respect to the exact order in which actions are instructed. As the presented approach makes use of the domain-independent structure of AMRs, we expect that it can generalize to other procedural texts, as well.

Acknowledgments

We thank Iris Ferrazzo, Xiulin Yang and the members of the Interactive Cookbook project for fruitful discussions. We also thank the reviewers for their helpful comments. This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – KO 2916/3-1.

References

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4430–4445, Online. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

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

Minsuk Chang, Leonore V. Guillain, Hyeungshik Jung, Vivian M. Hare, Juho Kim, and Maneesh Agrawala. 2018. Recipescape: An interactive tool for analyzing cooking instructions at scale. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA. Association for Computing Machinery.

Lucia Donatelli, Theresa Schmidt, Debanjali Biswas, Arne Köhn, Fangzhou Zhai, and Alexander Koller. 2021. Aligning actions across recipe graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6930–6942.

Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramón Astudillo. 2022. Inducing and using alignments for transition-based AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1086–1098, Seattle, United States. Association for Computational Linguistics.

Anubhav Jangra, Preksha Nema, and Aravindan Raghuveer. 2022. T-star: truthful style transfer using amr graph as intermediate representation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 8805–8825, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jermsak Jermsurawong and Nizar Habash. 2015. Predicting the structure of cooking recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 781–786, Lisbon, Portugal. Association for Computational Linguistics.

Chloé Kiddon, Ganesa Thandavam Ponnuraj, Luke Zettlemoyer, and Yejin Choi. 2015. Mise en place: Unsupervised interpretation of instructional recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 982–992, Lisbon, Portugal. Association for Computational Linguistics.

Fei-Tzin Lee, Chris Kedzie, Nakul Verma, and Kathleen McKeown. 2021. An analysis of document graph construction methods for amr summarization. *arXiv preprint arXiv:2111.13993*.

Angela Lin, Sudha Rao, Asli Celikyilmaz, Elnaz Nouri, Chris Brockett, Debadeepta Dey, and Bill Dolan. 2020. A recipe for creating multimodal aligned datasets for sequential tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4871–4884, Online. 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*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Xiao Liu, Yansong Feng, Jizhi Tang, Chengang Hu, and Dongyan Zhao. 2022. Counterfactual recipe generation: Exploring compositional generalization in a realistic scenario. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7354–7370, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852, Online. Association for Computational Linguistics.

Shinsuke Mori, Hirokuni Maeta, Tetsuro Sasada, Koichiro Yoshino, Atsushi Hashimoto, Takuya Funatomi, and Yoko Yamakata. 2014a. Flow-Graph2Text: Automatic sentence skeleton compilation for procedural text generation. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 118–122, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. 2014b. Flow graph corpus from recipe texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2370–2377, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tahira Naseem, Austin Blodgett, Sadhana Kumaravel, Tim O'Gorman, Young-Suk Lee, Jeffrey Flanigan,

Ramón Astudillo, Radu Florian, Salim Roukos, and Nathan Schneider. 2022. DocAMR: Multi-sentence AMR representation and evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3496–3505, Seattle, United States. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.

Md Sadman Sakib, Hailey Baez, David Paulius, and Yu Sun. 2021. Evaluating recipes generated from functional object-oriented network. *CoRR*, abs/2106.00728.

Yoko Yamakata, Shinji Imahori, Hirokuni Maeta, and Shinsuke Mori. 2016. A method for extracting major workflow composed of ingredients, tools, and actions from cooking procedural text. In *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6.

Yoko Yamakata, Shinsuke Mori, and John Carroll. 2020. English recipe flow graph corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5187–5194, Marseille, France. European Language Resources Association.

A Experiments

A.1 Model Training and Evaluation

Dataset. We restrict the recipes for our work to those recipes of the ARA1 corpus that describe the preparation of only one dish in a continuous text. Three of the 110 recipes do not meet this criterion and get excluded. The ten recipes for the test split were chosen manually based on the criteria that the original recipe text should not be shorter than 6 sentences and not include a lot of additional information or noise (e.g. nutrition lists). In order to avoid selecting recipes that are particularly easy for our approach, the recipes were selected by a student who was not familiar with the performance of the different parts of the pipeline on different kind of instructions and linguistic constructions. The test split consists of one recipe for each of the ten ARA1 dishes and comprises 151 A-AMR - sentence pairs in total. The remaining 97 recipes were randomly split into training and validation data.

ARA1
Baked Ziti, Blueberry Banana Bread, Cauliflower Mash, Chewy Chocolate Chip Cookies, Garam Masala, Homemade Pizza Dough, Orange Chicken, Pumpkin Chocolate Chip Bread, Slow Cooker Chicken Tortilla Soup, Waffles
ARA2
Bananas Foster, Chocolate Glaze, Cobb Salad, English Muffin Bread, Homemade Graham Crackers, How to Roast Garlic, Lavender Lemonade, Peanut Butter Bars, Sausage Grave, Southern Sweet Tea

Table 7: List of the Dishes from ARA1 and ARA2.

Gold instructions. Our extraction heuristic makes use of the node-to-token alignments produced by the AMR parser. Figure 4 shows the A-AMRs resulting from splitting the S-AMR for *"Top with shredded mozzarella cheese"* in PENMAN notation. We obtain the two corresponding gold instructions *"Shred mozzarella cheese"* and *"Top with mozzarella cheese"* by selecting all tokens from the original instruction to which nodes in the specific A-AMR are aligned. Tokens that have alignments to nodes in more than one A-AMR from the same S-AMR are included in the gold instruction for each of the A-AMRs (e.g. "mozzarella" and "cheese".) We use a set of rules based

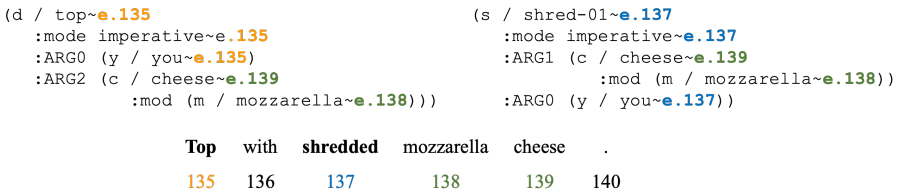


Figure 4: The A-AMRs obtained from the S-AMR for "Top with shredded mozzarella cheese." with the node-to-token alignments. The IDs of tokens that only have alignments to nodes in the A-AMR for "top" or for "shred" are marked in orange and blue respectively, and the IDs of tokens with alignments in both A-AMRs are green.

on patterns of the POS tags of successive tokens to decide about the selection of tokens that do not have an aligned node in any of the A-AMRs such as prepositions and determiners. Participle actions such as "shredded" get stemmed to ensure that the extraction heuristic creates grammatical imperative instructions. Additionally, the tokens selected for a gold instruction get partially reordered such that the action predicate is at the correct position in the new sentence if possible.

Training details. For training we build on the training scripts from the amrlib library[14] and adapt the required input format to our training set-up. All models are trained using the Adam optimizer and a linear learning rate scheduler with warm-up with $1e-4$ as the initial learning rate. The dropout rate is set to 0.1 for all reported experiments. The training and validation batch size is set to 24. We train all models using early stopping based on the train loss with a patience of 15 and a threshold of 0.00005 and select the final model based on the best BLEU score on the validation set.

Following previous work on using transformer LMs for AMR-to-text generation, we use a graph linearization based on the PENMAN format of the AMRs (Mager et al., 2020; Ribeiro et al., 2021, *inter alia*). We create the input to the model by concatenating the context sentence and the AMR in PENMAN format including the node-to-token alignments and introduce a special token to separate the context and the graph (see Table 8).

Generation. We set the token limitation for each generated sequence to 1024 and let the model output the best sequence using a beam size of 1.

Automatic evaluation. We compute BLEU, Rouge-1 (R-1), Rouge-2 (R-2), Rouge-L, Meteor (M) and Bleurt (BLRT) scores. For the computation of all automatic metrics we use the Huggingface Evaluate Metric package[15] which provides wrappers around the original metric implementations or the implementations from the SacreBLEU tool for comparable evaluation scores. We leave all parameters at their default values. For BLEU, we compute case-insensitive BLEU-4 at the corpus level. For BLEURT, we use the pre-trained bleurt-large-512 checkpoint and average over all predicted sentence-level scores to obtain a final BLEURT score.

A.2 Ablation Experiments

Unseen dishes. The test recipes are highly related to the training recipes as they are for the same dishes. In order to assess the performance of our models on new, unseen dishes, we evaluate the same GART5-0 and GART5-1 models also on the complete ARA2 recipes. The amrlib model performs worse than the GART5-0 model on all metrics on the ARA2 recipes, showing the general benefit of fine-tuning the generation model on a similar dataset from the same domain. However, the improvement is around 50% smaller than on the ARA1 recipes (see Table 9).

Effect of AMR type and alignments. We conducted some additional experiments to assess if and how the differences between the original S-AMRs and the split A-AMRs affect the performances of the models and to what extent the inclusion of the node-to-token alignments has an effect. Table 10 presents the results of training the GART5-0 and GART5-1 models on the A-AMR and the S-AMR datasets and testing on the same or the other dataset (approximately 40% of the amr-sentence pairs from the A-AMR dataset are also included in the S-AMR dataset). Overall, the results indicate that training and testing on the same kind of dataset yields the best results.

[14] https://github.com/bjascob/amrlib

[15] https://huggingface.co/evaluate-metric

GART5-0: <GRAPH> (t / top∼e.135 :ARG2 (c / cheese∼e.139 :mod (m / mozzarella∼e.138)))
GART5-1: Shred mozzarella cheese <GRAPH> (t / top∼e.135 :ARG2 (c / cheese∼e.139 :mod (m / mozzarella∼e.138)))
Output: Top with the mozzarella cheese.

Table 8: Example of the input to the GART5 generation model without context and with context and of a generated output sentence.

Model	BLEU	R-1	R-2	R-L	M	BLRT
amrlib	41.69	75.56	45.12	69.75	73.26	48.06
GART5-0	47.78 ±0.4	80.71 ±0.2	53.39 ±0.5	76.30 ±0.3	77.86 ±0.3	58.33 ±0.7
GART5-1	**51.08** ±0.5	**82.23** ±0.3	**57.57** ±0.6	**77.95** ±0.3	**79.63** ±0.4	**59.2** ±0.6

Table 9: Results on the full ARA2 dataset averaged over 6 different seeds (± standard deviation).

Regarding the effect of keeping the node-to-token alignments in the linearization we did not have any specific hypothesis. On the one hand, the amrlib model did not include alignments which could lead to a lower performance. On the other hand, the alignments indicate the original relative order of the words corresponding to the nodes and they might help the model to generate a well-ordered sentence. Table 11 presents the results from training and testing with and without the alignments in the graph linearization. The models trained and tested on the PENMAN linearization including the alignments perform best or second best across all metrics.

B Human Evaluation Set-up

Ordering heuristic. A correctly ordered sequence of the nodes $a \in \mathcal{N}_{\mathbf{A}}$ of an action graph needs to be a topological ordering of the action graph. However, not all potential orderings are good for structuring the steps in a recipe. For the generated recipes used in the human evaluation we defined a heuristic for ordering the actions that is based on the intuition that it is more convenient to keep working one subprocess for several steps instead of switching back and forth between different subprocesses. The traversal produces the ordered sequences $\mathcal{T}$ of action nodes in the following way:

1. Consider the set $\mathcal{B}$ of all nodes a without a parent node
2. Start the traversal with that node $a_i \in \mathcal{B}$ for which $\mathbf{Path}(a_i, end)$ is longest
3. Traverse the graph and add each visited node to $\mathcal{T}$ until reaching a node a_j that has parent nodes that are not yet in $\mathcal{T}$
4. Consider all nodes $a_k \in \mathcal{B}$ and $a_k \notin \mathcal{T}$ for which there is a $\mathbf{Path}(a_k, a_j)$ and select the node for which the path is longest to continue the traversal. If there are two candidate nodes chose the one which occurs earlier in the recipe text

Dependency baseline. The instructions for the baseline recipes are obtained based on the syntactic dependency tree of each instruction. The dependency splitting approach creates one instruction for each individual action predicate because the clustering approach to identify action events is based on semantic relations that are not available from the plain text. For each action, the baseline instruction is generated by selecting all tokens that can be reached by traversing the dependency tree starting from the action predicate without passing another action. We then use the same approach as for generating the gold instructions for re-ordering tokens and stemming participle actions.

Evaluation. For the generation of the recipe instructions presented in the human evaluation we used the specific checkpoints for which the automatic evaluation results are shown in Table 12. Table 13 presents the statements that were presented to the participants in the human evaluation study. Each participant saw one recipe at a time followed by the six statements. They were asked to rate for each of them to what extent they agree with the statement on a scale from 1 (disagree completely) to 6 (agree completely). In order to ensure that participants did pay attention to the recipe texts we included two filler recipes: one including multiple grammatical mistakes and one with randomly ordered instructions. The data from participants who rated the first one with a six or the latter one with 5 or higher was not included in the evaluation resulting in data from 88 participants.

C Clustering and Splitting

Table 5 lists rules that are used during the pairwise action clustering to decide whether two action nodes belong to the same event as well as the rules

Model	Train	Test	BLEU	R-1	R-2	R-L	M	BLRT
GART5-0	A-AMR	A-AMR	54.10	84.32	61.85	**79.98**	81.17	58.23
GART5-0	S-AMR	S-AMR	**54.86**	**84.64**	**64.58**	79.09	**82.58**	**59.81**
GART5-0	A-AMR	S-AMR	52.08	83.19	61.09	78.76	79.92	53.39
GART5-0	S-AMR	A-AMR	53.99	83.8	59.92	78.79	81.34	56.57
GART5-1	A-AMR	A-AMR	**59.26**	**86.34**	**67.78**	**83.19**	**84.25**	**65.14**
GART5-1	S-AMR	S-AMR	58.15	85.34	67.52	80.03	82.95	59.86
GART5-1	A-AMR	S-AMR	57.01	85.19	66.87	81.62	82.85	62.40
GART5-1	S-AMR	A-AMR	55.82	84.46	63.33	79.40	81.53	57.51

Table 10: Comparisons of the performance of the models when trained and tested on the A-AMR or S-AMR datasets.

Model	Train	Test	BLEU	R-1	R-2	R-L	M	BLRT
GART5-0	wA	wA	54.10	**84.32**	61.85	**79.98**	81.17	**58.23**
GART5-0	nA	nA	**54.66**	83.57	**61.87**	79.52	80.73	56.16
GART5-0	wA	nA	52.08	83.19	61.09	78.76	79.92	53.39
GART5-0	nA	wA	53.59	82.73	60.67	78.29	**81.23**	56.88
GART5-1	wA	wA	**59.26**	**86.34**	**67.78**	**83.19**	**84.25**	**65.14**
GART5-1	nA	nA	58.29	85.57	65.92	82.26	83.17	61.04
GART5-1	wA	nA	58.17	85.12	65.32	81.84	82.82	61.92
GART5-1	nA	wA	57.57	85.08	65.54	81.91	83.97	63.61

Table 11: Comparison of performances for different graph linearizations: with node-to-token alignments (wA) and without (nA).

for deciding already based on the root node that an S-AMR will not get separated. The full set of path patterns used by the rules are presented in Table 14.

In §3 we presented the main parts of the splitting algorithm. Table 6 presents the full algorithm with all rules and special cases. The following notation is used to describe the splitting conditions: When describing a path of actions between two nodes that includes and edge e_k we use the notation $e_k^{\rightarrow}$ and $e_k^{\leftarrow}$ to differentiate between edges that are traversed in their original direction ($e_k^{\rightarrow}$) and edges that are traversed in the reverse direction ($e_k^{\leftarrow}$). Therefore, if a path includes $\langle ..., e_k^{\rightarrow}, e_l^{\rightarrow}, ... \rangle$ with $e_k = (u, v) and e_l = (v, w)$ (i.e. the original edge in the graph is (w, v)) then v is a meeting node.

Model	Test context	BLEU	R-1	R-2	R-L	M	BLRT
GART5-0	0	54.10	84.32	61.85	79.98	81.17	58.23
GART5-1	1	**59.26**	**86.34**	**67.78**	**83.19**	**84.25**	**65.14**

Table 12: Results of the specific checkpoints used to generate the texts for the human evaluation on the ARA1 test split.

Criterion	Statement
Grammar	The recipe text is grammatically correct.
Fluency	The recipe text reads smoothly.
Verbosity	The recipe explains the steps concisely and does not repeat information unnecessarily.
Structure	The recipe explains the steps in a helpful order.
Success	In combination with a list of the required ingredients, the recipe would enable me to successfully prepare the dish.
Overall	Overall, the recipe is well written.

Table 13: The statements used in the human evaluation to assess the quality of the recipes along different criteria.

Root-based rules:
Let $\mathbf{M_i}$ be an S-AMR with root node $r_{\mathbf{M_i}}$. Do not split $\mathbf{M_i}$ if one of the following holds:

- the label of the root is `or`, `slash`, `possible-01` or `have-condition-91`
- the root node has an outgoing edge $(r_{\mathbf{Mi}}, u)$ to any node u with the label `condition`

Action-pair based rules:
Let $\mathbf{M_i}$ be a S-AMR and a_1, a_2 two action nodes of $\mathbf{M_i}$ aligned to different actions:
1. Pair a_1 and a_2 into one action cluster if there exists a path $\mathbf{Path}(a_1, a_2)$ which does not include any direction changes and if for the corresponding labelled path $\mathbf{LPath}$ one of the following conditions holds:

- The $\mathbf{LPath}$ corresponds to one of the path patterns from Pattern Set1 in Table 14, with a_1 and a_2 corresponding to **Node1** and **Node2**
- The $\mathbf{LPath}$ corresponds to one of the path patterns from Pattern Set2 in Table 14, with a_1 and a_2 corresponding to **Node1** and **Node2** and one of the following conditions holds
  - the path $\mathbf{Path}(a_1, a_2)$ between the two action nodes does not contain a node v that is labelled `before` or `after`
  - the path $\mathbf{Path}(a_1, a_2)$ between the two action nodes contains a node v that is labelled `before` or `after` and v has more than one child node.

2. Pair a_1 and a_2 into one action cluster if there exists a path $\mathbf{Path}(a_1, a_2)$ with exactly one direction change, i.e. with one meeting node v, and if one of the following conditions holds for the corresponding labelled path $\mathbf{LPath}$ and the meeting node:

- The $\mathbf{LPath}$ corresponds to the first pattern of Pattern Set3 in Table 14 and v is labelled `or` or `slash`
- The $\mathbf{LPath}$ corresponds to the second pattern of Pattern Set3 in Table 14 and v is labelled `contrast-01`

Figure 5: Rules for the pairwise clustering of action nodes of an S-AMR into action-event clusters.

	Label node1	LPath	Label node2
Pattern Set1	*action*	⟨ ARGX (opX)[1] ⟩	*action*
	action	⟨ direction (opX)[1] ⟩	*action*
	action	⟨*edge*⟩	off|up|down|out|in
	stir-01	⟨*edge*⟩	fry-01
Pattern Set2	*action*	⟨ (*edge*)* *relation* (*edge*)* ⟩	*action*
	where *relation* is equal to purpose, manner, instrument, time or duration		
Pattern Set3	*action*	⟨ opX, opX-of ⟩	*action*
	action	⟨ ARGX, ARGX-of ⟩	*action*

Table 14: Path patterns between two action-aligned AMR nodes that should be clustered together. *action* and *edge* can be any node or edge label, round brackets are used for optional labels on the **LPath**, ()[1] meaning zero or exactly one occurrence and ()* allowing any number of occurrences.

Input: a copy $\mathbf{N_i}$ of an S-AMR graph $\mathbf{M_i}$, the target action cluster C_j, and the set of all other action clusters $C_k, k \neq j$

Output: an A-AMR graph for C_j if successful, else the original S-AMR

1. create the set $\mathcal{Q}$ of all pairs $\{(a_1, a_2) | a_1 \in C_j$ and $a_2 \in \bigcup_{k \neq j} C_k\}$, i.e. all pairs of action AMR nodes that need to get separated from each other

repeat

2. compute all paths $p = \mathbf{Path}(a_1, a_2)$ for all pairs $(a_1, a_2) \in \mathcal{Q}$ in the graph $\mathbf{N_i}$ and create a sequence $\mathcal{P}$ of all paths ordered by length in ascending order
3. **if** $\mathcal{P} = \emptyset$ then **break** because then all nodes from C_j are successfully separated from all other action clusters
4. **for** p in $\mathcal{P}$ **do**
   - 4.1 **if** p does not include any node u labelled before or after
     - 4.1.1 **if** p has exactly one direction change ($\rightarrow$ to $\leftarrow$ or $\leftarrow$ to $\rightarrow$), and ($p = \langle ..., e_k^{\leftarrow}, e_l^{\rightarrow}, ... \rangle$ or $p = \langle ..., e_k^{\rightarrow}, e_l^{\leftarrow}, ... \rangle$) with $e_k = (v, w)$ and $e_l = (w, x)$, i.e. w is the meeting node, **then** remove e_l from $\mathbf{N_i}$ and **continue** from step 2.
   - 4.2 **else** p includes a node u labelled before or after
     - 4.2.1 **if** p has no direction changes **then** remove u from $\mathbf{N_i}$ and **continue** from step 2.
     - 4.2.2 **else if** p has exactly one direction change ($\leftarrow$ to $\rightarrow$), and $p = \langle ..., e_k^{\leftarrow}, e_l^{\rightarrow}, ... \rangle$ with $e_k = (v, w)$ and $e_l = (w, x)$, i.e. w is the meeting node, and $\mathcal{L}_{\mathbf{Mi}}(w) =$ and **then** remove u from $\mathbf{N_i}$ and **continue** from step 2.
     - 4.2.3 **else if** p has exactly one direction change ($\rightarrow$ to $\leftarrow$ or $\leftarrow$ to $\rightarrow$), and ($p = \langle ..., e_k^{\leftarrow}, e_l^{\rightarrow}, ... \rangle$ or $p = \langle ..., e_k^{\rightarrow}, e_l^{\leftarrow}, ... \rangle$) with $e_k = (v, w)$ and $e_l = (w, x)$, i.e. w is the meeting node, **then** remove e_l from $\mathbf{N_i}$ and **continue** from step 2.
5. **for** p in $\mathcal{P}$ **do** (fallback case if $\mathbf{N_i}$ did not change during step 4.)
   - 5.1 **if** p has more than one direction change, and ($p = \langle ..., e_k^{\leftarrow}, e_l^{\rightarrow}, ..., e_o^{\rightarrow}, e_p^{\leftarrow}, ... \rangle$ or $p = \langle ..., e_k^{\leftarrow}, e_l^{\rightarrow}, ..., e_o^{\leftarrow}, e_p^{\rightarrow}, ... \rangle$) with $e_k = (v, w)$, $e_l = (w, x)$ and w being the first meeting node and $\mathcal{L}_{\mathbf{Ni}}(w) =$ and with $e_o = (y, z)$, $e_p = (z, z_2)$ and z being the last meeting node **then** remove e_p from $\mathbf{N_i}$ and **continue** from step 2
   - 5.2 **else if** p has more than one direction change, and ($p = \langle ..., e_k^{\leftarrow}, e_l^{\rightarrow}, ... \rangle$ or $p = \langle ..., e_k^{\rightarrow}, e_l^{\leftarrow}, ... \rangle$) with $e_k = (v, w)$, $e_l = (w, x)$ and w being the first meeting node **then** remove e_l from $\mathbf{N_i}$ and **continue** from step 2
6. **if** $\mathbf{N_i}$ did not change during step 5. **then return** original graph $\mathbf{M_i}$

end repeat

7. select the connected subgraph that includes all nodes from the target cluster C_j as the new action-event-level AMR, apply the postprocessing and **return** the graph

Figure 6: The full splitting algorithm.

Meaning Representation of English Prepositional Phrase Roles: SNACS Supersenses vs. Tectogrammatical Functors

Wesley Scivetti **Nathan Schneider**
Georgetown University
nathan.schneider@georgetown.edu

Abstract

This work compares two ways of annotating semantic relations expressed in prepositional phrases: semantic classes in the Semantic Network of Adposition and Case Supersenses (SNACS), and tectogrammatical functors from the Prague English Dependency Treebank (PEDT). We compare the label definitions in the respective annotation guidelines to determine expected mappings, then check how well these work empirically using Wall Street Journal text. In the definitions we find substantial overlap in the distributions of the two schemata with respect to participants and circumstantials, but substantial divergence for configurational relationships between nominals. This is borne out by the empirical analysis. Examining the data more closely for participants and circumstantials reveals that there are some unexpected, yet systematic divergences between definitionally aligned groups.

1 Introduction

Broad coverage descriptive frameworks for annotating lexical semantics have proven useful for researchers in the field of computational semantics. Most of these frameworks have a primary focus on verbs and their participants (Baker et al., 1998; Bonial et al., 2014; Kipper et al., 2008; Palmer et al., 2017), though some frameworks extend annotation schema to cover the arguments of nominal phrases (Hajič et al., 2012; Meyers et al., 2004). Relatively few frameworks have focused on comprehensive accounts of prepositions, which can modify both verbal and nominal heads (Schneider et al., 2018; Litkowski and Hargraves, 2005), and can contribute crucial semantic information to sentences despite often being thought of as purely functional elements.

The most recent and comprehensive attempt to cover the semantics of prepositions is the Semantic Network of Adposition and Case Supersenses, or SNACS (Schneider et al., 2015, 2016, 2018), which is a hierarchy of semantic classifications of prepositional modifiers. SNACS contains 52 total preposition semantic classes, or SUPERSENSES, which are arranged into a hierarchy with different levels of granularity at each point in the hierarchy. In English, the SNACS framework has been applied to the reviews section of the English Web Treebank (EWT) corpus (Bies et al., 2012), resulting in the STREUSLE corpus with gold SNACS annotations (Schneider et al., 2018).

For researchers interested in the lexical semantics of prepositions, the STREUSLE corpus is a valuable resource, but is smaller in size compared to corpora that have been annotated for other lexical semantic projects. While some of these other resources do mark some semantic information conveyed by prepositional phrases, it is an open question to what extent these more general semantic frameworks overlap with the preposition-centric hierarchy of SNACS. If there is significant overlap between corresponding classes across different annotation schema, it may be possible to convert the classifications of prepositional phrases in these more general schemata into corresponding SNACS supersenses. This would make it possible to quickly augment the available data annotated within the SNACS hierarchy, and would provide useful comparisons between the coverage of different annotation schemata.

In particular, this research highlights the Prague English Dependency Treebank (PEDT, Hajič et al. 2012) as one resource with potential overlap with SNACS.[1,2] The PEDT contains multiple layers of

[1] PEDT is the English side of the Prague Czech-English Dependency Treebank. One reason to examine this framework and corpus is that if the correspondence proves reliable for English, it might be leveraged to obtain heuristic SNACS annotations of Czech data as well, since the tectogrammatical annotation scheme is also applied in the Czech translation of the Wall Street Journal corpus.

[2] In a comparison of an earlier version of SNACS to PropBank semantic roles, Schneider et al. (2016) found good correspondences between supersenses and PropBank modifiers,

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 68–73

Functor	Supersense	Functor	Supersense	Functor	Supersense	Functor	Supersense	Functor	Supersense
TSIN	StartTime	LOC	Locus	MEANS	Instrument,Means	ACT	Agent,Force	EXT	Cost
TTILL	EndTime	DIR1	Source	MANN	Manner	PAT	Theme,Topic	APP	Gestalt
TFHL	Duration	DIR2	Direction,Path	CAUS	Explanation	ORIG	Originator	COMPL	Identity
THL	Duration	DIR3	Goal	AIM	Purpose	ADDR	Recipient	MAT	QuantityItem
THO	Frequency	EXT	Extent			BEN	Beneficiary	RSTR	Characteristic
TPAR	Time					ACMP	Ancillary	CPR	ComparisonRef
TWHEN	Time								

Table 1: Heuristic mapping from PEDT functor to SNACS supersense based on the guidelines. Functors and supersenses without a clear correspondence are omitted. EXT is listed twice because it maps to both Spatial and Configurational supersenses.

syntactic/semantic annotation for the entire WSJ section of the Penn Treebank (Marcus et al., 1993). We focus on the tectogrammatical layer, or t-layer, which describes the deep syntax/semantics of the sentence, and labels nominals with a set of FUNCTORS. Many of these functors seem to show remarkable overlap with SNACS supersenses, though there are some significant divergences. This work investigates the overlap between the SNACS hierarchy and functor labels for prepositional phrases from PEDT, by first qualitatively outlining the similarities between the definitions of semantic classes in the two frameworks, then offering an empirical analysis of their overlapping distributions on a set of WSJ sentences.

2 Definitional Comparison

The SNACS hierarchy v2.6 (Schneider et al., 2022) contains 52 total supersenses organized into 3 main branches: the **CIRCUMSTANCE** branch, the **PARTICIPANT** branch, and the **CONFIGURATION** branch. Recent versions of the SNACS hierarchy assign supersenses to a preposition for both its *scene* role and its *function* role. The *scene* role represents the contextual semantic role of a preposition in combination with the predicate, while the *function* role is more faithful to the lexical semantics of the preposition (Schneider et al., 2018; Hwang et al., 2017). In many instances, the two roles are the same, but in cases where the *scene* and *function* roles differ, the two are represented using the SCENE⇝FUNCTION notation. We hypothesize that the *scene* role SNACS supersenses will more closely align with PEDT functors, and thus focus on *scene* role supersenses unless otherwise specified. Many SNACS supersenses correspond more or less directly to PEDT functors based upon the definitions set forth in their respective guidelines. Table 1 lists PEDT functors with clear corresponding SNACS supersenses. We exclude supersenses without clear corresponding functors, as well as functors which are not directly relevant to the SNACS hierarchy.

We see in Table 1 that most CIRCUMSTANCE supersenses, which add spatial, temporal, or other description to events, usually have corresponding PEDT functors. In Example (1) we see an example of the overlap between the THL ("how long?") functor, and the DURATION supersense. The directional functors DIR1 and DIR3 best correspond to SOURCE and GOAL respectively, and not (despite the terminology) DIRECTION. This is because the start point of movement (which answers the question "where from?") is labeled SOURCE, and the end point of movement (which answers the question "where to?") is labeled as GOAL. Examples of DIR1 and DIR3 are shown in Examples (2) and (3).

(1) Big mainframe computers for business had been around **for**_THL_DURATION years.

(2) All came **from**_DIR1_SOURCE Cray Research.

(3) Despite recent declines in yields, investors continue to pour cash **into**_DIR3_GOAL money funds.

On the other hand, SNACS DIRECTION is used to express the orientation of motion where the end result is not specified. We can observe the distinction in Examples (4) and (5), which are taken from the most recent version of the SNACS annotation guidelines (Schneider et al., 2022). If DIR1 and DIR3 do not generally correspond to DIRECTION, then DIRECTION is exceptional in that it does not have a directly corresponding PEDT functor. DIRECTION, which is a subtype of PATH, is probably most closely related to the more general DIR2.

(4) I headed **to**_GOAL work.

but less uniform correspondences for numbered arguments.

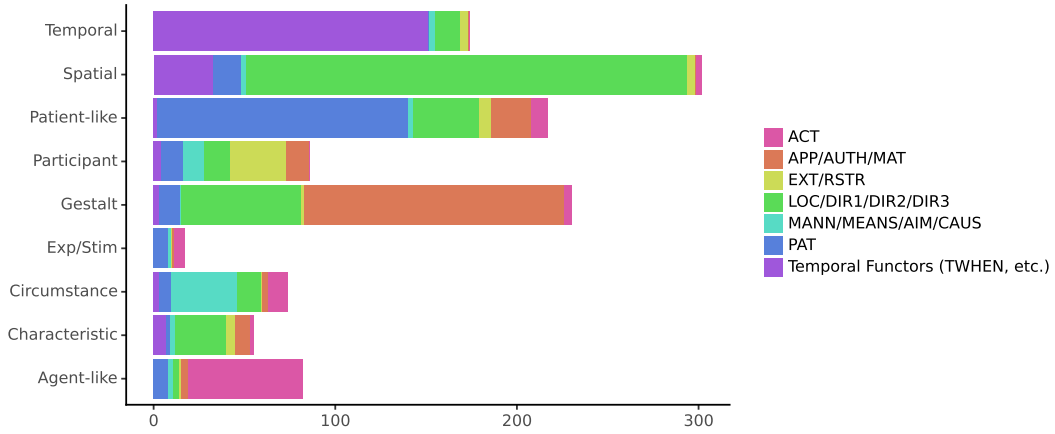


Figure 1: The Overlap of General Supersense Groupings with the PEDT Functors. Supersenses and Functors are combined into groups to show broad overlaps. Here "Circumstance" covers non-spatiotemporal circumstances (Manner, Means, Explanation, and Purpose). "Participant" covers Originator, Recipient, Beneficiary, Instrument, and Cost (excluding the agent-like, patient-like, and experiencer/stimulus participants).

(5) I headed **towards**_DIRECTION work, but never made it there.

PARTICIPANT supersenses, which introduce more canonical participants to events, also often correspond well to PEDT functors. The INSTRUMENT supersense lacks a directly corresponding functor, but is grouped with the MEANS supersense under the scope of the MEANS functor. The ACMP functor at least sometimes corresponds to ANCILLARY, as shown in Example (6). The ACT and PAT functors are potentially problematic, since they mark primarily syntactic roles of arguments, not semantic roles. This means that finer-grained supersenses, such as EXPERIENCER and STIMULUS, are not captured by PEDT functors. Furthermore, COST is perhaps the most problematic of the PARTICIPANT supersenses, with EXT being a marginal match at best.

(6) The U.S., **with**_ACMP_ANCILLARY its regional friends, must play a crucial role in designing its architecture.

CONFIGURATION supersenses, which describe state or property relationships between two nominals, are the least similar to PEDT functors, though there are some clear correspondences shown in Table 1, including the relationship between MAT and QUANTITYITEM as shown in Example (7). SNACS also includes more specific GESTALT subtypes, such as ORG and POSSESSOR, which are finer-grained than what is captured by the APP functor. Some more general configurations such as SPECIES, ENSEMBLE, and SOCIALREL lack

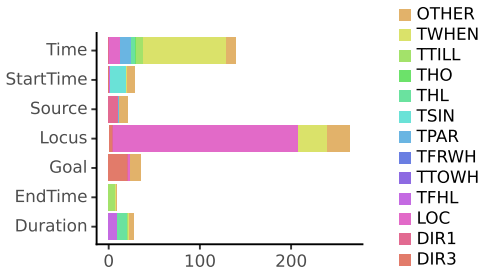


Figure 2: Spatiotemporal Supersense Overlap with PEDT functors

corresponding PEDT functors.

There are also some PEDT functors which are beyond the scope of the SNACS hierarchy: for instance, functors which mark paratactic relations (e.g. CONTRA), express primarily discourse functions (e.g. ATT), or mark types of syntactic information which is not conveyed in prepositional phrases (e.g. APPS). These functors are omitted from further analysis.

(7) About 20,000 sets **of**_MAT_QUANTITYITEM Learning Materials teachers' binders have also been sold in the past four years.

3 Empirical Comparison

3.1 Methodology

Now that we have outlined the overlap in descriptions between the SNACS hierarchy supersenses and various PEDT functors, we wish to quantify how these categories overlap in practice. In or-

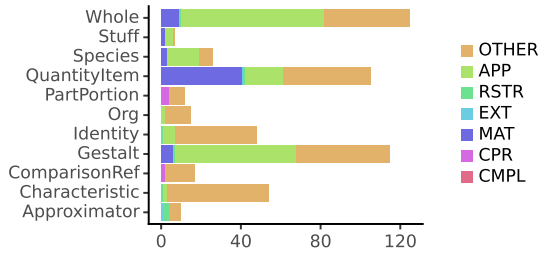


Figure 3: Configuration Supersense Overlap with PEDT functors

Class of Functors	# of Tokens	Percent Overlap
circumstantials	818	50.3
spatials	446	52.7
temporals	212	66.5
other	160	21.9
participants	500	47.0
ACT	113	55.8
PAT	238	58.0
other	149	22.8
configurations	386	36.0

Table 2: For groups of functors, percentage of tokens for which tagger-predicted supersense agrees with the heuristic mapping in Table 1. EXT is only considered in the configurations category.

der to compare the distribution of SNACS supersenses and PEDT functors, we first isolated all sentences containing relevant nominals from a small subsection of English PEDT using the tectogrammatical layer annotations. Specifically, we target nominals introduced by prepositional phrases, which have a formeme in the t-layer of the form "noun+preposition+X". Nominals with formemes of this type are found in a PP in the surface syntax. In total, we extract 838 sentences with 1837 total PPs with functor labels. These sentences were fed into a state-of-the-art SNACS supersense classifier (Arora, 2023), and a predicted supersense label was gathered for each of the target PPs. Since these tags were automatically generated, there is some expected noise in the resulting predictions, particularly for uncommon supersenses. We sampled 100 preposition tokens for manual tagging: focusing on the 71 that were not **of** tokens (as **of** is usually configurational), we found that the predicted classifier agreed with expert judgments roughly 60% of the time. We compare the automatically generated supersense labels with a rule-based heuristic based on our expectations outlined in §2. Generally, our heuristic aligns PEDT functors with the supersense that is most similar in definition. This heuristic was shown to be roughly 52% accurate on the manually tagged sample. After showing the overall distribution of supersenses across different functors, we then isolate examples of divergences between the automatic classifier and rule-based heuristic, finding that divergences come from both tagging errors and meaningful differences in the two frameworks.

3.2 Results

We compare the distribution of SNACS supersenses with PEDT functors in Figures 1 to 3. For all comparisons, supersenses that were predicted less than 5 times were excluded from analysis. Figure 1 shows the general overlap of different coarse groups SNACS supersenses with groupings of PEDT functors. We see here that supersenses grouped around broad semantic domains typically correspond to groups of PEDT functors with similar domains. The most clear correspondences are with the spatiotemporal, "Agent-like" and "Patient-like" supersenses, indicating that despite the syntactic definition of ACT and PAT in PEDT, they still pattern similarly to the semantic based categories in SNACS.

Figure 2 shows the overlap of spatiotemporal supersenses and functors with a higher degree of granularity than in Figure 1. We see that LOCUS and TIME are two of the most frequently predicted supersenses, and generally line up well with the LOC and TWHEN functors. This is in contrast with the overlap for **CONFIGURATION** supersenses, which is shown in Figure 3. We can see here that most of the supersenses seem to be spread over several competing PEDT functors. As expected, APP and MAT have substantial representation in these supersenses, but there is also considerable overlap with other unexpected PEDT functors.

We report the overlap of the predicted classifier supersenses with those predicted by a rule-based heuristic for different functor groupings in Table 2. We see that our expectations for functors and the predictions of the classifier diverge substantially, especially for configurations, though there is substantial divergence even in the spatiotemporal and participant classes.

Since the automatic SNACS classifier has substantial limitations in tagging WSJ data, it is worth considering whether the divergence reported in Table 2 is primarily due to tagging errors, or is due to real differences in annotation distributions for supersenses and functors. In Examples (8–12), we show the classifier-predicted supersense alongside the gold functor. For Examples (8, 9), the predicted

supersenses do not align with our expectations due to classification errors. We see in (8) that the classifier mistakenly predicts LOCUS instead of TIME. In this case, the heuristic which matches TWHEN to TIME would get this correct. In (9), the SNACS classifier predicts COST incorrectly, probably because it introduces a monetary amount as its dependent. Throughout the WSJ data, monetary values are often incorrectly classified as COST.

(8) The strong growth followed year-to-year increases of 21% **in**_TWHEN_LOCUS August and 12% in September.

(9) Imports were **at**_PAT_COST $50.38 billion, up 19%.

While classifier errors account for a substantial amount of misalignment between functors and supersenses, there are also systematic divergences. One reason for divergences is that some PEDT functors align more with SNACS *function* roles, rather than *scene* roles (as was expected). This is shown in Examples (10, 11), where both the predicted *scene* and *function* roles are shown. In Example (10), we see that the *scene* role of LOCUS does not align with DIR3, but the *function*, which is GOAL, does align with our expectations. This sentence is an example of *fictive motion*, where a preposition typically indicating motion is used in a static scene (Talmy, 1996; Hwang et al., 2017). In Example (11), we see that the *function* role of ANCILLARY is what we would expect to align with the ACMP functor, though the *scene* role AGENT does not. Problematic cases involving the ANCILLARY supersense have been a focus of prior SNACS research (Hwang et al., 2020), so it is perhaps unsurprising that some divergences arise in this case. Despite such examples, in most cases where *scene* and *function* differ, we observe that the *scene* role is closer to the PEDT functor. More investigation is needed to determine when PEDT functors map to *function* roles instead of *scene* roles in SNACS.

(10) The new plant, located in Chinchon about 60 miles **from**_DIR1_LOCUS⇝GOAL Seoul, will help meet increasing and diversifying demand for control products in South Korea, the company said.

(11) Moscow has settled pre-1917 debts **with**_ACMP_AGENT⇝ANCILLARY other countries in recent years at less than face value.

Beyond the discrepancies between PEDT functors and SNACS supersenses which arise from the *scene* and *function* distinction in SNACS, there are other unexpected divergences between PEDT functors and SNACS supersenses, two of which are shown in Examples (12, 13). In (12), the classifier's prediction of STARTTIME is obviously incorrect, but the expectation that CPR aligns with COMPARISONREF is also incorrect here. Instead, SOURCE is probably most appropriate, but is not predicted by the classifier or from our heuristic. This is one case where the usage of PEDT functors and SNACS supersenses do not overlap. Furthermore, in (13), the PEDT functor DIR3 would typically align with GOAL, but in this sentence the classifier prediction of PURPOSE is actually closer to the correct supersense. In general, it seems that DIR3 is not as clearly aligned with GOAL as anticipated, but also has some overlap with TOPIC, THEME, and PURPOSE. Despite the similar definitions of DIR3 and GOAL, in practice they are used in some non-overlapping situations.

(12) A seat on the Chicago Board of Trade was sold for $350,000, down $16,000 **from**_CPR_STARTTIME the previous sale last Friday.

(13) Then, in the guests' honor, the speedway hauled out four drivers, crews and even the official Indianapolis 500 announcer **for**_DIR3_PURPOSE a 10-lap exhibition race.

4 Conclusion

In this work, we compare SNACS supersenses with PEDT tectogrammatical functors in terms of how they account for English prepositions. We show that the substantial definitional overlap between SNACS supersenses and PEDT functors is reflected in the overlapping distributions of the various semantic classes, particularly for spatial, temporal, and participant related supersenses, with less overlap on the CONFIGURATION branch. However, we also find substantial divergences between the two schemata, due in part to limitations of the automatic SNACS classifier we employed. We observe that a simple heuristic mapping from PEDT functors to SNACS supersenses aligns somewhat with classifier predictions, but also has substantial limitations due to the differences between the two frameworks.

5 Acknowledgements

We thank Jan Hajič for helpful discussions regarding functors in PEDT, and anonymous reviewers for their constructive comments and suggestions. This research was funded in part by NSF award IIS-2144881.

References

Aryaman Arora. 2023. snacs: Models for parsing SNACS datasets. https://github.com/aryamanarora/snacs.

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.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. LDC2012T13.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of New Predicate Types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3013–3019, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3153–3160, Istanbul, Turkey. European Language Resources Association (ELRA).

Jena D. Hwang, Archna Bhatia, Na-Rae Han, Tim O'Gorman, Vivek Srikumar, and Nathan Schneider. 2017. Double Trouble: The Problem of Construal in Semantic Annotation of Adpositions. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, page 178–188, Vancouver, Canada. Association for Computational Linguistics.

Jena D. Hwang, Nathan Schneider, and Vivek Srikumar. 2020. Sprucing up Supersenses: Untangling the Semantic Clusters of Accompaniment and Purpose. In *Proceedings of the 14th Linguistic Annotation Workshop*, page 127–137, Barcelona, Spain. Association for Computational Linguistics.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A Large-scale Classification of English Verbs. *Language Resources and Evaluation*, 42(1):21–40.

Ken Litkowski and Orin Hargraves. 2005. The Preposition Project. In *Proc. of the Second ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, page 171–179, Colchester, Essex, UK.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The Nombank Project: An Interim Report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, page 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Martha Palmer, Claire Bonial, and Jena D. Hwang. 2017. VerbNet: Capturing English verb behavior, meaning, and usage. In Susan E. F. Chipman, editor, *The Oxford Handbook of Cognitive Science*, pages 315–336. Oxford University Press.

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Archna Bhatia, Na-Rae Han, Tim O'Gorman, Sarah R. Moeller, Omri Abend, Adi Shalev, Austin Blodgett, and Jakob Prange. 2022. Adposition and Case Supersenses v2.6: Guidelines for English. arXiv:1704.02134 [cs.CL].

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Meredith Green, Abhijit Suresh, Kathryn Conger, Tim O'Gorman, and Martha Palmer. 2016. A Corpus of Preposition Supersenses. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 99–109, Berlin, Germany. Association for Computational Linguistics.

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. Comprehensive supersense disambiguation of English prepositions and possessives. In *Proc. of ACL*, pages 185–196, Melbourne, Australia.

Nathan Schneider, Vivek Srikumar, Jena D. Hwang, and Martha Palmer. 2015. A hierarchy with, of, and for preposition supersenses. In *Proc. of The 9th Linguistic Annotation Workshop*, pages 112–123, Denver, Colorado, USA.

Leonard Talmy. 1996. Fictive motion in language and "ception". In Paul Bloom, Mary A. Peterson, Nadel Lynn, and Merrill F. Garrett, editors, *Language and Space*, pages 211–276. MIT Press, Cambridge, MA.

QA-Adj: Adding Adjectives to QA-based Semantics

Leon Pesahov[1] **Ayal Klein**[1] **Ido Dagan**[1]
[1]Computer Science Department, Bar Ilan University
{leonpes92,ayal.s.klein}@gmail.com dagan@cs.biu.ac.il

Abstract

Identifying all predicate-argument relations in a sentence has been a fundamental research target in NLP. While traditionally these relations were modeled via formal schemata, the recent QA-SRL paradigm (and its extensions) present appealing advantages of capturing such relations through intuitive natural language question-answer (QA) pairs. In this paper, we extend the QA-based semantics framework to cover adjectival predicates, which carry important information in many downstream settings yet have been scarcely addressed in NLP research. Firstly, based on some prior literature and empirical assessment, we propose capturing four types of core adjectival arguments, through corresponding question types. Notably, our coverage goes beyond prior annotations of adjectival arguments, while also explicating valuable implicit arguments. Next, we develop an extensive data annotation methodology, involving controlled crowdsourcing and targeted expert review. Following, we create a high-quality dataset, consisting of 9K adjective mentions with 12K predicate-argument instances (QAs). Finally, we present and analyze baseline models based on text-to-text language modeling, indicating challenges for future research, particularly regarding the scarce argument types. Overall, we suggest that our contributions can provide the basis for research on contemporary modeling of adjectival information.

1 Introduction

A main challenge addressed by Natural Language Processing research is designing useful semantic representations, capturing and explicating important aspects of the meaning of a text. Numerous recent works illustrate how even in the era of strong end-to-end neural models, leveraging explicit semantic representations facilitates downstream processing of challenging tasks (Huang and Kurohashi, 2021; Mohamed and Oussalah, 2019; Zhu et al., 2021; Chen and Durrett, 2021).

Numerous semantic representations have been proposed and pursued (Abend and Rappoport, 2017). Traditionally, semantic representations rely on pre-defined schemata of linguistic classes, e.g. semantic roles or relations. Thus, mapping natural language onto its representations becomes a complex annotation task that requires significant linguistic expertise, causing challenges in data collection and utility in new domains and languages.

Recently, many researchers and practitioners seek to benefit from an explicit representation of text meaning while alleviating the reliance on hard-to-scale structured formalisms. For instance, Open Information Extraction (Banko et al., 2007) has gained popularity as a light-weight, NL-based alternative to Semantic Role Labeling (SRL) formalisms like PropBank (Palmer et al., 2005) or FrameNet (Baker et al., 1998). More recently, several works proposed using question-answer pairs (QAs) as an intermediate structure, e.g. in order to assess information alignment between texts for evaluating summarization quality (Eyal et al., 2019; Gavenavicius, 2020; Deutsch et al., 2021) and faithfulness (Honovich et al., 2021; Durmus et al., 2020). While these latter works utilize "general-purpose" question-answering datasets and models for generating the QAs, Klein et al. (2022a) put forward a systematically targeted QA-based semantic framework dubbed *QASem*. Pioneered by addressing verbal predicate-argument relations in QA-SRL (He et al., 2015), this framework integrates three systematic QA-driven representations, jointly covering semantic role labeling for verbs (He et al., 2015; FitzGerald et al., 2018; Roit et al., 2020), nominalizations (Klein et al., 2020) and informational discourse relations (Pyatkin et al., 2020).

However, current QA-based approaches lack principled coverage of adjectival information. In natural language text, adjectives carry vital information about the properties of entities, essential for many downstream NLP applications. For example,

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 74–88
June, 2023.

*Galbraith attacked the consensus for monetarist economics and argued that Keynesian economics were far **more relevant** for tackling the emerging crises.*

Question type	Question	Answer
Object	What was more relevant for something?	Keynesian economics
Comparison	Compared to what was something more relevant?	monetarist economics
Domain	What was something more relevant for?	tackling the emerging crises
Extent	To what degree was something more relevant?	far

Table 1: An example of QA-Adj question-answer pair.

in benchmarks of the widely-used sentiment analysis task (Pontiki et al., 2014), adjectives comprise 75% of the annotated "sentiment triggers".

In this work, we extend the QASem paradigm by capturing and explicating the fundamental aspects of adjectival information using natural-language question-answer pairs. Our representation, termed *QA-Adj*, consists of four adjective-related roles — object, comparison, domain, and extent. As we will see later, these roles provide a fairly complete representation of the core arguments of adjectives. Roles are annotated using question templates while arguments are captured as answers, as illustrated in Table 1. In addition to syntactical arguments, which are commonly available in prior semantic or syntactic representations, QA-Adj is designed to capture and explicate implicit arguments not immediately discernible from syntax, for example, stating (in Table 1) *Compared to what is something **more relevant**?*.

The main contributions of this paper are as follows: (1) We formulate a QA-based representation for capturing adjectival arguments, grouping them into four semantic categories; (2) we present a method for collecting low-cost, high-quality QA-Adj data through controlled crowdsourcing; (3) we create a QA-Adj dataset, comprising over 5K sentences and 12K QA pairs, assess its quality, and compare it to PropBank annotations for adjectival predicates (Bonial et al., 2014); (4) we finetune a baseline QA-Adj parser and evaluate its performance, providing a foundation for future model development.

Overall, our work provides an intuitive QA-based representation for explicitly capturing the semantics of adjectives, as well as a dataset and a parser for future research.

2 Background

2.1 Semantic Representations of Adjectives

Traditional logical approaches denote adjectives, as well as verbs, as predicates over entities — e.g., $\text{RED}(x) \wedge \text{BALL}(x)$ would represent *a red ball*, and $\text{AFRAID}(x, y)$ may denote that x *is afraid of* y. NLP semantic formalisms, however — such as PropBank (Palmer et al., 2005), Minimal Recursion Semantics (Copestake et al., 2005), semantic dependencies (Oepen et al., 2015) and more (Banarescu et al., 2013; Abend and Rappoport, 2013, inter alia) — commonly adopted the Neo-Davidsonian approach (Parsons, 1995). This approach decomposes predicative meaning into a set of binary relations between entities and events, labeled by semantic roles, e.g. $\text{FEAR}(e) \wedge \text{AGENT}(e, x) \wedge \text{THEME}(e, y)$.

While the SRL task has gained substantial attention, research thereof focuses primarily on the semantics of verbs or eventive nouns. Nevertheless, several computational resources include adjectives under their scope. In FrameNet (Baker et al., 1998) — a well-known SRL formalism — adjectives are listed in frames with their participants, or Frame Elements, in the same way verbs and nominals do. For example, the adjective *hungry* is listed under the BIOLOGICAL-URGE frame. Similarly, with the goal of complementing PropBank with information about new predicate types, Bonial et al. (2014) annotated adjectives in the PropBank corpus using both pre-existing and newly introduced framesets, along with corresponding semantic roles (see annotation examples in Table 2). In contrast, the formulation presented in this work targets four broad, generic semantic dimensions pertaining to any adjective, coupled with corresponding question templates, and does not require mapping adjectival predicates to a pre-defined inventory of frames. In Section 5.5, we further compare our approach to prior representations.

2.2 QA-Based Semantics

Semantic Role Labeling (Palmer et al., 2010) is typically perceived as answering argument role questions, such as *who*, *what*, *to whom*, *when*, or *where*, regarding a target predicate. For instance, PropBank's ARG0 for the predicate *say* answers the

Sentence	QA-Adj	PropBank
(1) There's so much punch packed into this combination that it's almost **scary**.	Object: What is scary? — There's so much punch packed into this combination + it Extent: To what degree is something scary? — almost	ARG0: it ARGM-EXT: almost
(2) Although these new rockets are probably more **expensive**, they will be able to go at a much greater range than it's shuttle cousins.	Object: What is expensive? — these new rockets + they Extent: To what degree is something expensive? — more expensive Comparison: Compared to what is something expensive? — it's shuttle cousins	ARG1: these new rockets ARGM-EXT: more
(3) The 69 year old Dr. Lopez was found **guilty**.	Object: Who is guilty? — The 69 year old + Dr. Lopez	ARG1: The 69 year old Dr. Lopez
(4) Wise decision to go through the private sector – NASA's budget may be kinda **tight** to fund a project like this.	Object: What might be tight? — NASA's budget Extent: To what degree might something be tight? — kinda Domain: What might something be tight to do? — to fund a project like this Comparison: Compared to what might something be tight? — the private sector's budget	ARG0: NASA's budget ARGM-EXT: kinda ARGM-PRP: to fund a project like this
(5) If anyone is interested in listening to this song, and in offering their opinion whether it be **positive** or negative, I'd appreciate it.	Object: What might be positive? — their opinion + it Domain: What might something be positive about? — this song	ARG1: it
(6) If you have any questions please feel **free** to call me (after Sat. the 26th, when I will return from a trip).	Object: Who is free to do something? — you Domain: What is someone free to do? — call me	ARG3: call me (after Sat. the 26th, when I will return from a trip).
(7) Should the Arctic Ocean become ice **free** in summer, it is likely that polar bears would be driven toward extinction.	Object: What might be free? — the Arctic Ocean	ARG1: the Arctic Ocean ARG2: ice
(8) That is what is about to happen with Judge Samuel Alito, in my opinion, because he has one tragic flaw – a very serious blind spot in his thinking – which makes him completely **unacceptable** for the position of Supreme Court Justice.	Object: Who is unacceptable for something? — Judge Samuel Alito + he Extent: To what degree is someone unacceptable? — completely Domain: What is someone unacceptable for? — the position of Supreme Court Justice	ARG1: him ARGM-EXT: completely ARG3: for the position of Supreme Court Justice
(9) She doesn't have the funds to continue without the grant, and without these treatments, her prognosis is **grim**.	Object: What is grim? — her prognosis without these treatments	ARG1: her prognosis ARG-MNR: without these treatments

Table 2: A sample of QA-Adj annotations, along with corresponding PropBank annotations for adjectives (Bonial et al., 2014, see §5.5 for the comparison). The **+** sign denotes multiple answers for the same question. While most QA-Adj QAs are similar to PropBank predicate-argument relations, many introduce additional information, including implicit or inferred relations (Ex. 2, 4, 5) and within-sentence coreference (Ex. 1, 5, 8). Annotation mistakes are rare, but include incorrect splitting of arguments (Ex. 3), incomplete QA-Adj answers (Ex. 6) and recall misses (Ex. 7).

question "Who **said** something?". QA-SRL (He et al., 2015) suggests that answering role questions is an intuitive means to solicit predicate-argument structures from non-expert annotators. In QA-SRL, annotators are presented with a sentence in which a target predicate has been marked, and are asked to generate questions and highlight the corresponding answers from the sentence. A question captures the semantic role, whereas answers to the question — which are spans from the sentence — denote the set of arguments associated with that role. The QA-based approach allows for a transparent representation, as the questions and answers can be understood by non-experts while providing an explicit account of the underlying meaning of the sentence. This laymen-intuitive definition of roles covers traditional cases of syntactically linked arguments, but also additional semantic arguments clearly *implied* by the sentence meaning (Roit et al., 2020).

QA-SRL has been demonstrated to be beneficial for various downstream tasks. It was shown to subsume open information extraction (OIE) (Stanovsky and Dagan, 2016b), making it possible to construct large supervised OIE dataset (Stanovsky et al., 2018) to serve as an intermediate structure for end applications. Additionally, QA-SRL and related QA-based semantic annotations (Michael et al., 2018) were shown to provide beneficial semantic signal through indirect supervision, resulting in improved performance on downstream tasks for modern pre-trained-LM encoders (He et al., 2020). Recently, QA-SRL was explicitly utilized as an intermediate representation for aligning predicate-argument relations across texts (Brook Weiss et al., 2021) and for detecting analogies through structure mapping (Sultan and Shahaf, 2022).

To address a broader semantic scope, the QA-SRL formalism, well suited for scalable crowdsourcing (FitzGerald et al., 2018), has been incrementally extended to account for discourse relations using semi-templated questions and answers (Pyatkin et al., 2020) as well as for deverbal nominalizations (Klein et al., 2020). These tasks, jointly denoted *QASem*, have been recently bundled by a unifying modeling framework and parsing tool (Klein et al., 2022a). In the QASem framework, each propositional predication relation — in the spirit of the aforementioned Neo-Davidsonian approach — is captured through a corresponding Question-Answer pair. In this work, we further

Question Type	PREFIX	WH	AUX	SBJ	DET	TRG	PP	OBJ	?
Object		Who	are		the	most suitable	for	something	?
Domain		What	is	someone		active	in		?
Comparison	Compared to	what	is	something		prominent			?
Extent	To what degree		is	something		popular			?

Table 3: Example questions illustrating our question templates.

extend the QASem paradigm to account for adjectives.

3 Task Formulation

In order to keep the task simple — both for annotation and for modeling — we consider all adjectives occurring in the sentence under the same formulation. We thus refrain from distinguishing different classes of adjectives (e.g. subsective, intersective or privative (Partee, 2007; Pavlick and Callison-Burch, 2016); superlative and comparative; etc.) or different syntactic realizations of adjectives — i.e. attributive vs. predicative (*the red ball* vs. *the ball is red*).

While free-formed questions have been proposed as a natural representation of semantic relations (Michael et al., 2018), prior works show that they yield inferior coverage relative to annotation schemes that systematically design restricted question templates, such as QA-SRL and QADiscourse (Pyatkin et al., 2020; Klein et al., 2020). Consequently, we adopt the template-based approach and design question templates corresponding to four core argument types of adjectival semantics that have practical value for downstream applications. The coverage of these templates was validated through the examination of prior linguistic works on adjectives (Huddleston and Pullum, 2002; Baker et al., 1998). See Table 3 for an illustration of each question template.

The most basic argument role for an adjective is the entity described by it, which corresponds to the predicated entity variable in logical representations and is captured by all other representation schemes as well. Our annotation scheme captures this argument role through the first question type (*What/Who is [ADJ]*), termed here **Object**.

In addition to Object, we adopt the three semantic dimensions of adjectives as identified by Ikeya (1995), namely — the *Thematic* dimension, the *Comparative* dimension, and the *Degree* dimension.

The *Thematic* dimension is mapped to the **Domain** question type in our scheme. Answers to this question type give a semantic specification to the adjective — For example, *good at dancing*, *lactose intolerant* and *former president*. To illustrate, two-place predicates (in first-order logic) would mostly fit their arguments into our Object and Domain roles. While often syntactically attached to the adjective, such answers can also occur as implicit arguments (Ex. 5 in Table 2).

The **Comparison** question type is aimed to capture the group or entity referenced by the adjective to which the object is being compared. These arguments are frequently implicit (e.g. Ex. 2, 4 in Table 2) and are therefore mostly neglected in prior formalisms that rely on syntax, such as PropBank.

Lastly, the **Extent** question type corresponds to the *Degree* dimension, that is, to what extent does the adjectival assertion holds. Such arguments can be realized by adverbs (e.g. *almost complete*, *very good*) or by more complicated constructions (*too political for my liking*, *competent enough for this job*, etc).

We note that our questions are designed at capturing semantic complements of the adjective meaning. In preliminary investigations, incorporating adverbial modifiers into the task scope was found to introduce annotation noise. Our role set thus omits adverbial modifiers, such as time and location (e.g. *By June, you'll be **capable** of programming by yourself*), leaving their investigation for future work.

In this paper, we focus on "core" adjectival arguments as laid down by Ikeya. See Appendix A.1 for a more elaborated discussion.

QA Format In the spirit of He et al. (2015), we define a small grammar over possible questions. The questions are constrained by a template with eight fields, $q \in$ **PREFIX** $\times$ **WH** $\times$ **AUX** $\times$ **SBJ** $\times$ **DET** $\times$ **TRG** $\times$ **PP** $\times$ **OBJ**, each associated with a set of possible options (see Table 3). Full descriptions for each field are provided in Table 8 in the Appendix.

Answers are selected from the words in the sentence but can be manually modified in order to make the answer appropriate and natural-sounding.

	Sentences	Adjectives	Total Roles		Object		Domain		Comparison		Extent	
			QAs	Answers	QAs	Answers	QAs	Answers	QAs	Answers	QAs	Answers
Train	3377	7266	8198	9080	6802	7654	613	627	412	426	371	373
Dev	668	750	951	1099	733	872	90	93	80	85	48	49
Test	1281	1659	2093	2398	1622	1914	176	178	189	199	106	107
Total	5326	9695	11242	12577	9157	10440	879	898	681	710	525	479

Table 4: Annotation statistics of the QA-Adj dataset.

We instruct the annotators to rewrite answers manually only when copying words from the sentence is insufficient for constructing a meaningful or grammatical answer, such as in Ex. 4 in Table 2 (*the private sector**'s** budget*). In addition, questions may have multiple answers, in order to better account for coordinations or co-referring entity mentions (Ex. 1, 5, 8 in Table 2).

We further guide our annotators to include restrictive modifiers (Stanovsky and Dagan, 2016a) in their answers, as these are considered an integral part of the noun phrase, e.g., the underlined modifier in *"She wore the **shiny** necklace that her mother gave her"*. Non-restrictive modifiers, which provide parenthetical information about the entity — e.g., *"The speaker thanked **former** president Obama, who just walked into the room"* — are not included in the answer span.

4 Dataset Construction

Preprocessing and annotation interface In this section, we describe the dataset creation process and in section 5 analyze its quality. We annotated over 5K sentences with 9K adjective mentions, across two domains: Wikinews and Wikipedia. We select sentences that are also covered by previous annotated QASem datasets (Roit et al., 2020; Pyatkin et al., 2020; Klein et al., 2020). In each sentence, we identify the target adjectives using SpaCy's POS-tagger. If an adjective is preceded by one of the words 'more', 'less', 'most', or 'least', then it is considered part of the target adjective. Table 4 shows the full data statistics.

We developed a Graphical User Interface (GUI) (See Appendix, Figure 1) deployed at Amazon's Mechanical Turk crowdsourcing platform. The worker, presented with a sentence with a marked adjective as a target, should generate question-answer pairs pertaining to this adjective. Questions are generated by filling templated slots using drop-down lists, whereas answers are selected by highlighting spans from the sentence, and manually corrected if needs be. The GUI also includes a short overview of the task and instructions, along with 5 annotation examples.

Annotator selection and training We adapted the controlled crowdsourcing process used by Roit et al. (2020) for QA-SRL. After establishing the task formulation and interface, the first two authors jointly annotated 60 instances as a seed gold set, for evaluating and guiding worker qualification. We then release a preliminary crowd-wide annotation round and contact workers who exhibit reasonable performance. They are asked to review our short guidelines, which highlight a few subtle aspects, and then annotate four qualification rounds, of 15-30 target adjectives each. Each round is followed by extensive feedback via email, pointing at errors and missed arguments, which are identified by automatic comparison to expert annotation. In total, this worker training process lasted approximately 8 weeks, and cost 240$, and is orders of magnitude shorter and simpler than training annotators for traditional semantic formalisms.

Annotation process During data collection, we observed that outcomes of a single crowd annotator tend to be of insufficient quality, especially with respect to capturing the rather infrequent roles of Domain, Comparison and Extent. To enhance the coverage of the evaluation set (dev & test), we aggregated QAs from two independent QA-generation workers and forwarded them to *consolidation*. In the consolidation task, a third worker reviewed and judged the aggregated generated annotations, producing a non-redundant consolidated set.

While aggregating annotations from multiple generators coped well with the coverage challenge, data precision was still mediocre for the non-Object roles, as opposed to the Object role, where precision was satisfactory (in Section 5.4 (Table 5) we report evaluated quality for each phase of the data collection process.). We hence employed an additional *expert verification* step pertaining to instances in which one of the non-Object arguments is provided. In this step, one of the first two authors of this paper reviewed the annotations, filtering or

	Generation (avg. of 2 workers)			Consolidation			Expert Verification (avg. of 2 experts)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Object	83.7	78.5	81.0	87.7	93.4	**90.4**	-	-	-
Domain	46.1	64.4	53.7	43.4	82.6	56.9	93.3	84.9	**88.9**
Comparison	61.4	44.7	52.6	64.1	75.4	69.2	91.7	77.1	**83.7**
Extent	49.5	67.7	57.1	67.5	80.6	73.4	86.6	80.6	**83.4**
Total	72.3	72.2	72.2	75.8	89.3	82.0	88.0	89.5	**88.8**

Table 5: Evaluating the different annotation stages against an expertly annotated reference set of 300 instances (See §5.2 for evaluation metrics). **Bold** numbers represent the final stage in the annotation process of the dev & test sets.

fixing answers to non-Object questions as required. Verification was applied on top of consolidated annotations for the dev and test sets (1010 out of 2409 adjective instances), and over single-generator annotations for the training set (2182 out of 7266 adjective instances).

Annotation cost Our annotators were paid 20¢ per instance in both the generation and consolidation steps, and a single expert verification assignment takes around 30 seconds. The resulting cost per instance in the development and test sets is 60¢ (2 generators + consolidator), along with around 30 seconds of expert review time. In the training set, the cost is 20¢ and 30 expert review seconds per instance. In total, creating the development and test sets costs 1445$ and 9 hours of expert verification, while the training set costs 1456$ and 19 expert hours, totalling 2891$ and 28 hours of expert review time for the entire dataset. This approach allowed us to efficiently collect a high-quality dataset for our QA-based representation of adjectival semantics.

5 Dataset Analysis and Quality

In this section, we report several analyses to quantify and establish the quality and coverage of the QA-Adj dataset. Additional Information about the joint distribution of different roles is reported in Appendix A.4.

5.1 Implicit Arguments

A key benefit of our laymen-intuitive annotation task is its aptitude to capture implicit arguments, that is, arguments that are harder to automatically read off of syntax (See Ex. 2, 4, 5 in Table 2 for illustrations). To quantify this aspect, we utilize a syntactic dependency parser[1] for measuring the proportion of implicit arguments on the evaluation sets. Following similar prior analyses (Klein et al., 2020), an argument is considered implicit if none of its words is connected to the predicate on an undirected dependency tree in a path of length ≤ 2.

We find that 17%, 30%, 49%, and 13% of the arguments are implicit for the Object, Domain, Comparison, and Extent roles, respectively. This demonstrates that many of our semantic arguments are hardly accessible from syntactic representations, especially for the Comparison and Domain roles. Focusing on the Object role, we further inspect that 91% of the instances have at least one explicit argument, which entails that most of the implicit arguments provide a (commonly more informative) coreferring mention of a syntactically-connected argument (e.g. Ex. 8 in Table 2).[2] In the remaining 9%, the object entity is connected through more complex linguistic constructions such as control or raising verbs (*The audience is asked to remain* ***silent***), adverbial clauses (*Argentina dropped three places to be ranked* ***sixth***) and coordinations (*Switzerland and Italy each moved down one, ranked eighth and* ***ninth*** *respectively*). In sum, relying on intuitive laymen annotations naturally yields many informative arguments that fall out of scope of more linguistically oriented representations.

5.2 Evaluation metrics

We use the same evaluation protocol both for dataset analysis (this section) and for model evaluation (Section 6). Given predicted QAs for all adjectives, we report precision and recall against the ground truth for each question type separately, as well as for the total set of predicted QAs. Following previous work on annotating semantic relations with QA pairs (Roit et al., 2020; Pyatkin et al., 2020; Klein et al., 2020), answers of the same question type are considered a match if the intersection over union (IOU) between the sets of tokens in each answer is greater than 0.3.

[1] We apply the same SpaCy model used for POS-tagging in preprocessing.

[2] In general, while multiple answers are rare for other role questions, 16% of the Object questions are answered by more than one answer, most commonly due to coreferring mentions.

5.3 Inter-Annotator Agreement

To assess the consistency of the annotated data, we measure the inter-annotator agreement on the dev & test sets, as well as expert-vs-expert agreement on data used as part of the validation and test sets for parser evaluation. The **Object** QA type macro-averaged F1 inter-annotator agreement is **76.0**, while for QA types **Comparison, Domain, Extent** it is **29.8, 37.0, 41.3**, respectively.

The main issue in disagreement arises from sentences that do not contain apparent adjectival arguments, especially in question types Comparison and Domain, where workers are inclined to ask questions either way, resulting in sometimes unnatural or overly implicit questions. To measure the expert-vs-expert agreement, we randomly sample 227 instances that underwent the consolidation process and contain at least one of the Comparison, Domain, or Extent roles. We perform the expert review step on them by each of the first 2 authors of this paper and compare the outputs. The expert-vs-expert F1, excluding the Object question type which was not reviewed in the expert-review step, reaches a reasonable **77.9** F1. Notably, the consolidation and expert review steps boost consistency significantly.

Agreement on restrictive modifiers We conjecture that a decent proportion of annotator disagreements arise from the difficulty to designate the proper argument span, which requires keeping within the span restrictive modifiers of the argument while omitting non-restrictive modifiers (Stanovsky and Dagan, 2016a). Therefore, we estimate the agreement between annotators on modifiers' restrictiveness by sampling from the final dataset 50 answers of the Object role that contain a restrictive modifier, as judged by the first author, and examining whether both annotators captured it. 26 modifiers were captured by both annotators, mostly simple prepositional phrases (e.g. *routes for complex molecules*), while 16 were captured only by one of the annotators. 8 were missed by both, but captured by the consolidator. Examining the missed modifiers, we find that many involve non-continuous span selection (which is feasible through the manual modification our interface enables on top of a copied sentence span). For example, in the sentence *"The alveolar letters had longer left stems, while retroflexes had* ***longer*** *right stems"*, the correct argument is *right stems of retroflexes*, while the annotators only captured *right stems*, omitting this implicit restrictive modifier which is nonetheless essential for demarcating the precise argument.

5.4 Dataset Assessment by Gold Reference Set

To ensure the quality of our annotation, we created a gold reference set consisting of 300 instances from the development set. The reference set should represent QA-Adj annotations of optimal quality. For this purpose, we take generated annotations along with their consolidation decisions (as described in §4) and manually correct them by each of the two first authors independently. We then reconciliate to resolve any disagreement.

We compare the annotations attained from the initial generation step, consolidation step, and single expert verification step against the reference set (Table 5). Results indicate that consolidation significantly boosts coverage, and confirm the high quality of our full annotation protocol (in bold).

5.5 Comparison with Other Formalisms

In this section, we compare QA-Adj to two common representations covering adjectival semantics — PropBank (Palmer et al., 2005) and Abstract Meaning Representation (AMR; Banarescu et al., 2013).

PropBank for adjectives One of the most widely used resources of English predicate-argument structure is PropBank, which has also incorporated adjectival predicates (Bonial et al., 2014). It is thus illuminating to examine the overlap and discrepancies between QA-Adj and PropBank. For this purpose, we collect QA-Adj annotations for 150 adjective instances from PropBank using the same annotation protocol as for the evaluation set (§4), yielding 296 answers (260 QAs) compared against 232 PropBank arguments. We employ our evaluation protocol (§5.2) to measure argument agreement between the two annotation schemes, and manually examine disagreements. Examples throughout this section are referring to Table 2.

Notably, the scope of adjectival arguments targeted by the two annotation schemes is somewhat divergent. Designed to explicate the syntactic-semantic interface, PropBank captures some syntactic markers (e.g. discourse, relative clause, negation, and modality) that cannot naturally answer role questions. It is worth mentioning that QA-Adj annotations incorporate information about negation and modality within the questions (see Ex. 5),

Sentence	Test set	Parser output
(1) Any deviation from this family model is considered a "**nontraditional** family".	Object: What is nontraditional? — a family + any deviation from this family model Comparison: Relative to what is something nontraditional? — this family model	Object: What is nontraditional? — family
(2) Regarding the lack of women members in the cabinet, Mr. Abbott said he was "**disappointed**".	Object: Who was disappointed? — he + Mr. Abbott Domain: What was someone disappointed about? — the lack of women members in the cabinet	Object: Who was disappointed? — Mr. Abbott Domain: What was someone disappointed about? — the lack of women members in the cabinet

Table 6: Comparison between QAs in the test set and the parser's output. Example 1 demonstrates an implicit argument that the parser missed, while in Example 2, the parser captured such an argument.

following the QA-SRL approach. In addition, PropBank includes many types of adverbial modifiers that are out of QA-Adj scope (§3). We thus exclude PropBank roles that pertain to syntactic markers or adverbials from our henceforth quantitative analysis (details in Appendix A.5) and focus on core argument roles.

QA-Adj covers **93.1**% of PropBank arguments, demonstrating that our task formulation and annotation substantially capture traditional predicate-argument relations. One source of disagreements are pronouns (Ex. 8), which PropBank captures in the form they appear in the sentence (e.g. *him*), while our flexible rewriting mechanism allows to capture them in the more natural subject form (i.e. *he*). Out of 16 PropBank arguments not covered by QA-Adj, only 6 reflect actual QA-Adj annotation misses (Ex. 7). Another source of disagreement (4 out of 16) is QA-Adj arguments that are split into multiple roles in PropBank's finer-grained annotation (Ex. 9).

On the other hand, PropBank arguments cover only **72.9**% of QA-Adj annotated answers. Out of 80 QA-Adj arguments that don't match PropBank annotations, 70 are correct but fall out of PropBank's scope. These include co-referring mentions (14; Ex. 1, 2, 5), implicit arguments (22; Ex. 2, 4, 5), and cases where PropBank arguments are split by our scheme into two distinct, co-referring answers (11; See Ex. 3).

While this analysis elucidates the relationship between QA-Adj and a more traditional semantic formalism, it also reaffirms the coverage of our QA-Adj annotations, demonstrating that non-experts can capture a major portion of the information found in PropBank. At the same time, relying on intuitive NL-based QAs introduces new types of implicit information that seem useful downstream, in addition to making the annotations cheaper, faster, and easier to replicate compared to expertly annotated formalisms.

Abstract Meaning Representation AMR is a comprehensive semantic representation designed to capture semantic aspects of complete sentences, including adjectival semantics, in an abstract, cross-language manner. It employs various mechanisms to account for adjectival semantics. For instance, the phrase "attractive spy" is represented with the corresponding verbal roleset, SPY :ARG0-OF ATTRACT-01, while for other adjectives, AMR defines specific framesets (e.g. SAD-02). The specification for semantic roles is predicate-specific, where, to cite AMR guidelines, "ARG0 often refers to the thing being described by the adjective, while ARG1 names the next most natural argument.".[3] These correspond to QA-Adj Object and Domain roles in most cases.

A later study (Bonial et al., 2018) expands the AMR lexicon with various constructions, including a HAVE-DEGREE-91 roleset, which handles degree adjectives and related constructions. Upon a close inspection, we find that QA-Adj Comparison and Degree roles capture most of the information within the HAVE-DEGREE-91 roles, though in a more coarse manner. See Appendix A.6 for an elaboration on the comparison to AMR.

6 Baseline Models

We devise an initial QA-Adj parser to serve as a baseline for future work on this task. We first apply the same preprocessing steps for identifying target adjectives as in our data collection procedure (§4). Then, following a prior QA-driven semantic parser (Klein et al., 2022a), we fine-tune the Text-to-Text Transformer model (T5; Raffel et al., 2020), which unifies multiple text modeling tasks, and achieves state-of-the-art results in various NLP benchmarks. We use Huggingface (Wolf et al., 2020) for fine-tuning the T5 model. A special token is marking the target adjective within the input sentence, while

[3] https://github.com/amrisi/amr-guidelines

Model Evaluation	Single Model Automatic			Role-specific Models Automatic			Role-specific Models Manual		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Object	82.2	75.1	78.5	78.2	75.4	76.8	86.3	84.4	85.3
Comparison	36.8	43.7	40.0	36.2	47.7	41.2	60.0	45.2	51.5
Domain	50.5	51.1	50.8	51.8	55.0	53.4	62.5	60.9	61.6
Extent	41.7	57.0	48.2	80.0	52.6	63.2	85.0	57.5	68.5
Total	72.5	69.9	71.2	71.5	70.6	71.0	73.7	62.5	67.6

Table 7: Baseline models evaluation. Automatic evaluation results are on the full test set, while manual evaluation is on a sample.

the output is formatted as *question: <**Q**> answer: <**A**>*. In case the semantic role is empty, the parser is to generate the special token [NO-QA].

In preliminary experiments, training a single model to generate all four QA pairs in one go has yielded poor results. We hypothesize this is due to the sparsity of the Domain, Comparison and Extent question types, which appear in 8%, 5%, and 5% of the training examples, respectively.

Therefore, to set up baseline results, we fine-tune an independent T5 model on each question type separately. The train set per question type consists of all instances which have the specific question type answered, along with random negative samples, i.e. empty QA instances. The ratio of negative samples is treated as a hyper-parameter of the model and is optimized on the development set.

Since holding a separate fine-tuned T5 model for every QA type is memory-consuming, we also fine-tune a single T5 model using the union of the training sets of each question type, using a different prefix for each QA type.

6.1 Results

Previous work on QA-based semantics has demonstrated that automatic argument-matching criteria can be too strict (Roit et al., 2020). Hence, to better estimate precision, we randomly select 40 generated QAs for each question type and assess their validity manually. Similarly, to estimate recall, we sample 40 annotated QAs of each question type and manually compare them to the parser's output.

Table 7 presents the automatic evaluation measures for a single parser trained on all roles, as well as automatic and manual evaluation of the role-specific models. Results indicate there is ample room for improvement, particularly on the more subtle roles of Comparison and Domain.

One factor contributing to the challenges in capturing these roles is the high prevalence of implicit arguments within them (Ex. 1 in Table 6), as demonstrated in our analysis (Section 5.1). As implicit arguments often rely on commonsense reasoning rather than syntactic structure, they may be more difficult for a model to identify. In future work, we aim to investigate methods for better capturing implicit arguments and explore the use of external knowledge sources to aid in this task.

7 Conclusion

In this work, we propose and realize a new approach to representing the semantics of adjectives using natural language question-answer pairs, focusing on four generic, core semantic dimensions. This intuitive representation enables high-quality yet scalable annotation through controlled crowdsourcing along with minimal expert verification. Our annotations explicate the fundamental aspects of an adjective's meaning in context, substantially overlapping with an expertly annotated SRL resource while adding previously uncovered implicit arguments.

We advocate utilizing QA-Adj downstream as an alternative for syntactical or semantic representations. As an example, recent works on aspect-based sentiment analysis use syntactic or semantic dependencies as scaffolds for enhancing domain transfer (Wang and Pan, 2019; Pereg et al., 2020; Klein et al., 2022b). Explicating relations between adjectives (sentiment/opinion terms) and their semantic objects (aspect terms) directly, QA-Adj is a worthwhile alternative to dependency representation.

Future works should explore methods for improving the baseline models presented in this work, such as prompt tuning (Lester et al., 2021) or multi-task learning with related QA-semantic tasks (Klein et al., 2022a). In addition, since the annotations are based on natural language and layman workers, it is appealing to transfer the scheme into various languages, possibly utilizing both machine translation and/or crowd annotations.

8 Limitations and Ethics

Unlike prior QASem annotation tasks, we empirically find that adding an expert verification step on a selective portion of the data — where more subtle roles are handled — is important for maintaining good precision. Indeed, despite putting efforts in making the task guidelines simple and intuitive, margins of the semantic space often introduce complexity that is hard to account for in a consistent manner without linguistic background. Although still significantly faster than full-fledged expert annotation, requiring an expert in the loop may pose a bottleneck to scaling annotations to large datasets and new domains and languages, which is a shortcoming of the current proposal.

Annotations were conducted on Amazon Mechanical Turk (MTurk) with an average pay of $12 per hour for all crowdsourcing data collection tasks. To maintain the anonymity of our workers, we do not collect personal information and do not keep any deanonymizing information such as MTurk IDs.

License The data collected in this work is licensed under the Creative Commons license.

Acknowledgements

This research was funded in part by grants from Intel Labs, the Planning and Budgeting Committee (PBC) of the Israeli Council for Higher Education under the National Data Science Competitive Program, and the Israel Science Foundation grant 2827/21.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–89, Vancouver, Canada. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Claire Bonial, Bianca Badarau, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Tim O'Gorman, Martha Palmer, and Nathan Schneider. 2018. Abstract Meaning Representation of constructions: The more we include, the better the representation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of new predicate types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3013–3019, Reykjavik, Iceland. European Language Resources Association (ELRA).

Daniela Brook Weiss, Paul Roit, Ayal Klein, Ori Ernst, and Ido Dagan. 2021. QA-align: Representing cross-text content overlap by aligning question-answer propositions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9879–9894, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jifan Chen and Greg Durrett. 2021. Robust question answering through sub-part alignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1251–1263, Online. Association for Computational Linguistics.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. 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 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale QA-SRL parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060, Melbourne, Australia. Association for Computational Linguistics.

Mantas Gavenavicius. 2020. Evaluating and comparing textual summaries using question answering models and reading comprehension datasets. B.S. thesis, University of Twente.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. QuASE: Question-answer driven sentence encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8743–8758, Online. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yin Jou Huang and Sadao Kurohashi. 2021. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052, Online. Association for Computational Linguistics.

Rodney D. Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press.

Akira Ikeya. 1995. Predicate-argument structure of English adjectives. In *Proceedings of the 10th Pacific Asia Conference on Language, Information and Computation*, City University of Hong Kong, Hong Kong. City University of Hong Kong.

Ayal Klein, Eran Hirsch, Ron Eliav, Valentina Pyatkin, Avi Caciularu, and Ido Dagan. 2022a. Qasem parsing: Text-to-text modeling of qa-based semantics. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7742—7756, Online and Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ayal Klein, Jonathan Mamou, Valentina Pyatkin, Daniela Stepanov, Hangfeng He, Dan Roth, Luke Zettlemoyer, and Ido Dagan. 2020. QANom: Question-answer driven SRL for nominalizations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3069–3083, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ayal Klein, Oren Pereg, Daniel Korat, Vasudev Lal, Moshe Wasserblat, and Ido Dagan. 2022b. Opinion-based relational pivoting for cross-domain aspect term extraction. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 104–112, Dublin, Ireland. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing question-answer meaning representations. 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 560–568, New Orleans, Louisiana. Association for Computational Linguistics.

Muhidin Mohamed and Mourad Oussalah. 2019. Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing Management*, 56(4):1356–1372.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic Role Labeling*, 1st edition. Morgan and Claypool Publishers.

Terence Parsons. 1995. Thematic relations and arguments. *Linguistic Inquiry*, pages 635–662.

Barbara H Partee. 2007. Compositionality and coercion in semantics: The dynamics of adjective meaning. *Cognitive foundations of interpretation*, pages 145–161.

Ellie Pavlick and Chris Callison-Burch. 2016. So-called non-subsective adjectives. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 114–119, Berlin, Germany. Association for Computational Linguistics.

Oren Pereg, Daniel Korat, and Moshe Wasserblat. 2020. Syntactically aware cross-domain aspect and opinion terms extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1772–1777, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online. Association for Computational Linguistics.

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

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013, Online. Association for Computational Linguistics.

Gabriel Stanovsky and Ido Dagan. 2016a. Annotating and predicting non-restrictive noun phrase modifications. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1256–1265, Berlin, Germany. Association for Computational Linguistics.

Gabriel Stanovsky and Ido Dagan. 2016b. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. 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 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Oren Sultan and Dafna Shahaf. 2022. Life is a circus and we are the clowns: Automatically finding analogies between situations and processes. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Online and Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenya Wang and Sinno Jialin Pan. 2019. Syntactically meaningful and transferable recursive neural networks for aspect and opinion extraction. *Computational Linguistics*, 45(4):705–736.

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.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

A Appendices

A.1 Omitting Adverbial Modifiers

As mentioned in the paper body (§3), our role questions are designed toward semantic aspects that complement the meaning of the adjective. Consequently, in the spirit of the famous linguistic *argument//modifier* distinction (or *complement//adjunct*), we choose not to incorporate questions targeting generic ("adjunctive") adverbial information, such as temporal, causal, or locative modifiers of the copular phrase.

Field	Description	Values
PREFIX	Specific question type prefixes	Compared to, Relative to, To what degree
WH*	Question words	who, what
AUX*	Auxiliary verbs	is, was not, could be, ...
SBJ	Place-holder for subject position	someone, something
DET	Determiner	the
TRG*	The target adjective	tall, active, most accurate, ...
PP	Frequent prepositions	by, for, in, ...
OBJ	Placeholder for object position	someone, something

Table 8: The fields of question templates. WH, AUX and TRG are required; all other fields may be left empty.

This design choice arises from practical considerations. In preliminary investigations and crowdsourcing experiments, we have found the distinction between modifiers and the **Domain** role to be rather intricate, especially for non-linguist annotators. For example, locative or temporal descriptions that are commonly adverbial modifiers (*He was **hungry** this morning*) can in certain cases be semantic complements (***earlier** this morning*). Further, when supporting modifier questions like "When is something [ADJ]?" in the interface, non-expert annotators are often inclined to embrace loosely related and erroneous phrases as arguments. To illustrate, the instance *"If you have any questions please feel **free** to call me (after Saturday the 26th)"* (from Ex. 6 at Table 2) might be annotated with the inaccurate QA *"When should someone be free? — after Saturday the 26th"*.

A.2 Question Templates Description

Table 8 shows a full description of the 8 question fields comprising the four question templates (one per role), together with possible values that can fill each question field. Each field's exact set of optional values defines the role-dependent question template. In the table, we use three dots (...) to denote partial lists of values (the full lists would be released as supplementary material upon acceptance).

A.3 Annotation Task User Interface

Our Graphical User Interface (Figure 1) allows to create multiple answers per QA type (+ button). Upon answering the Object question, its answer is embedded in the other questions, making them more natural.

A.4 Arguments Joint Distribution

The sparsity of arguments corresponding to the question types Domain, Comparison, and Extent is a major challenge in our task and data (See Table 4). Indeed, as demonstrated in Section 6.1, this sparsity makes it difficult for a parser to accurately identify these roles. In our development set of 750 adjective instances, most (547) have only the Object question answered. There are 157 instances with two answered questions, 26 instances with three, and only three instances with all four questions answered.

A small minority of instances has no argument roles at all (17 out of 750 on dev). This is primarily due to POS-tagger erroneous adjective identification — for example, *Khufu's pyramid **complex** consists....* Annotators were instructed to leave empty such erroneous target adjectives, where our roles questions are not sound.

A.5 PropBank Roles Excluded from Comparison

Following our discussion in Appendix A.1, we need to account for the scope discrepancy between QA-Adj and PropBank prior to measuring their argument agreement. We thus exclude PropBank arguments capturing adverb, causation, temporal, location and relative clause roles, as well as markers of discourse, modality, and negation. The full list of PropBank's excluded roles, along with examples, can be seen in Table 9.

A.6 More details about AMR Comparison

Predicative vs. Attributive Adjectives AMR maintains a directionality distinction between predicative adjectives (*The marble is white*) and attributive adjectives (*The white marble*). Predicative adjectives would be the "root" of the sentence graph

Role Name	Argument	Sentence
ARGM-ADV	your career	The one department of life that may not quite be as hopeful as you'd like could be your career, where advancement may be slow and satisfaction **rare**.
R-ARGM-ADV	where	
ARGM-LOC	areas	There have been large numbers of population extinctions in Mexico and southern California in areas where the habitat is still **acceptable**.
R-ARGM-LOC	where	
ARGM-CAU	reasons	Also, from their own webpage, reasons why NASA is **important**, in a 5th-grade format.
R-ARGM-CAU	why	
R-ARG1	who	80 - Percentage of the Iraqi workforce who are **unemployed** a year after the war.
ARGM-MOD ARGM-NEG	may not	They may not be **familiar**, but they will be fascinating.
ARGM-TMP	when we love another person	We become most fully **human** when we love another person.
ARGM-DIS	Please	Please feel **free** to call me.

Table 9: Examples of PropBank roles omitted from comparison to QA-Adj.

(or clause subgraph), and the subject entity would be their :DOMAIN argument, e.g. WHITE :DOMAIN MARBLE. Attributive adjectives, on the other hand, are denoted as :MOD arguments of their target entity, e.g. MARBLE :MOD WHITE. This distinction is necessary for maintaining a fine-grained account of sentence meaning, as it captures the sentence focus, which may have pragmatical implications. In QA-Adj, and QASem in general, we take a more "informational" perspective on semantics (rooted in more traditional logical representations), thus wishing to abstract out surface realization details that do not modify the conveyed information.

Degree constructions Bonial et al. (2018) expands the AMR lexicon with various constructions. These include a HAVE-DEGREE-91 roleset, which handles constructions related to degree adjectives, such as comparatives, superlatives, or more idiosyncratic constructions, e.g. what they term 'Degree Consequence' (see Table for example annotations). The HAVE-DEGREE-91 roleset comprises the following semantic roles:

- ARG1: domain, entity characterized by attribute
- ARG2: attribute (e.g. tall)
- ARG3: degree itself (e.g. more/most, less/least)
- ARG4: compared-to
- ARG5: superlative: reference to superset
- ARG6: consequence, result of degree.

Compared to our scheme, ARG1 directly corresponds to the **Object** role, while ARG3 and ARG6 correspond to the **Extent** role. ARG4 and ARG5 align with the **Comparison** role. Examples illustrating this mapping are presented in Table 10. This comparison illustrates that the roles defined by our task are less fine-grained than those that can be found, at least in some contexts, in other semantic frameworks like AMR. Our choice of granularity is informed by our objective, aiming to facilitate streamlined non-expert annotation. Nevertheless, the comparison also demonstrates that our four roles adequately cover the most essential semantic roles of adjectival semantics.

Sentence	AMR	QA-Adj
(1) The watch is too **wide** for my wrist.	Arg1: watch Arg2: wide Arg3: too Arg6: my wrist	Object: What is wide? — The watch Extent: To what extent is something wide? too wide for my wrist
(2) The girl is **taller** than the boy.	Arg1: girl Arg2: tall Arg3: more Arg4: boy	Object: Who is taller? — The girl Comparison: Compared to whom is someone taller? — the boy

Table 10: Examples of AMR annotations for adjectives, using the specialized HAVE-DEGREE-91 roleset, along with corresponding QA-Adj annotations.

Figure 1: User interface for the Question-Answer Generation task.

The long and the short of it: DRASTIC, a semantically annotated dataset containing sentences of more natural length

Dag T. T. Haug
University of Oslo, Norway
d.t.t.haug@ifikk.uio.no

Jamie Y. Findlay
University of Oslo, Norway
jamie.findlay@iln.uio.no

Ahmet Yıldırım
University of Oslo, Norway
ahmet.yildirim@iln.uio.no

Abstract

This paper presents a new dataset with Discourse Representation Structures (DRSs) annotated over naturally-occurring sentences. Importantly, these sentences are more varied in length and on average longer than those in the existing gold-standard DRS dataset, the Parallel Meaning Bank, and we show that they are therefore much harder for parsers. We argue, though, that this provides a more realistic assessment of the difficulties of DRS parsing.

1 Motivation

Corpora with deep, logic-based semantic annotations are quite rare because they are so hard to annotate. The arrival of the Groningen Meaning Bank (Bos et al., 2017) and the Parallel Meaning Bank (PMB; Abzianidze et al., 2017) changed this situation by offering full Discourse Representation Structures (DRSs; Kamp, 1981b) for substantial amounts of text in Dutch, English, German, and Italian. The current release, 4.0.0, contains more than 10,000 sentences in English and between 1,400 and 2,800 sentences in other languages. However, the dataset contains both bronze (automatic), silver (partial manual disambiguation), and gold (full manual disambiguation) data, and the gold sentences are consistently very short (mostly $<$10 words). Since the dev, test, and eval sets contain only gold data, this means that DRS parsers are tested only on very short sentences, yielding an overly optimistic assessment of results in this area.

In this paper, we improve on the situation by offering a gold standard dataset containing DRSs with a more realistic sentence length distribution. We call this dataset DRASTIC, for 'Discourse Representation Annotation with Sentence Texts of Increased Complexity'.[1] An additional strength of DRASTIC is that the texts it contains – three contiguous documents plus a selection of medium-length sentences – are from the GUM corpus (Zeldes, 2017), allowing users to explore connections between the DRS annotation and the rich annotation available in GUM: beside morphosyntactic annotation following the Universal Dependencies (UD) scheme (de Marneffe et al., 2021), this also includes entity recognition, coreference, discourse structure and more.[2] The current size of our dataset is small, at 157 sentences with full manual disambiguation, but around 1,000 more sentences have received a first manual annotation by student annotators and will subsequently be integrated into the dataset.

DRS parsing gets harder as sentences grow longer (cf. van Noord et al., 2020, 4594f.). This is natural, but some peculiarities of the PMB annotation are especially hard to capture, and contribute only little extra information. Cases in point are recursive presuppositions, strict separation of different presuppositions of a single sentence, and the use of discourse relations with relatively bland content such as CONTINUATION. As the sentence grows in length, these result in a complex network of embedded DRSs. In such cases, parser output that is (more or less) logically equivalent to the gold can still get a low score. To avoid this, we simplify the annotation of such structures (see Section 2.2). Since our corpus is small and does not include training data, we provide a script that flattens PMB-style annotations to our format. This can be used to flatten PMB data before training a parser, or alternatively to flatten the output of a parser trained on the PMB.

The structure of the paper is as follows. In Section 2, we introduce Discourse Representation Theory (DRT), as well as the PMB annotation and our simplifications of it. In Section 3, we describe our corpus, and Section 4 studies the effects of sen-

[1] The dataset and accompanying scripts are available here: https://github.com/Universal-NLU/DRASTIC.

[2] The list at https://gucorpling.org/gum/annotations.html (accessed 31 May 2023) provides the full set of annotation layers.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 89–98
June, 2023.

tence length on DRS parsing and offers baseline modelling results on our data.

2 The format: Discourse Representation Structures

In Discourse Representation Theory (Kamp, 1981b; Kamp and Reyle, 1993; Kamp et al., 2011; Kamp and Reyle, 2019) the meaning of a sentence is analysed as its contribution to the existing semantic representation of the discourse context, called a Discourse Representation Structure (DRS). This means that DRT belongs to the family of theories called *dynamic semantics*, although DRT treats only the process of interpretation as dynamic, not the notion of meaning itself.[3]

DRSs are traditionally represented as boxes divided into two: a universe of discourse at the top, containing a number of *discourse referents*, which can then be referred to by the set of *conditions* in the lower part of the box. DRS conditions are by and large simply formulae of some predicate logic, but can also contain complex conditions relating multiple DRSs via logical operators like negation, implication, and disjunction, or modal operators like possibility and necessity. By way of illustration, Figure 1 gives a DRS for the sentence *Jadzia thought that Miles or Julian had been hurt*. DRT is compatible with many different specific theoretical approaches to semantics; in Figure 1, as in our corpus, we use a Neo-Davidsonian event semantics where events and states (collectively called *eventualities*) are treated as first-class entities in the ontology, and semantic dependents are related to their heads via thematic role predicates such as Agent, Patient, etc. (on event semantics see e.g. Davidson, 1967; Parsons, 1990). A basic representation of tense is also given, by including the relation Time between an eventuality and its time, and relating that time to the constant 'now' (referring to the time of utterance) or to other times.

Aside from the rich body of theoretical work in DRT exploring various knotty semantic phenomena such as anaphora (Kamp, 1981b; Haug, 2014), tense (Kamp, 1981a), rhetorical structure (Lascarides and Asher, 1993; Asher and Lascarides, 2003), propositional attitudes (Asher, 1986; Kamp, 1990), and others, one other good reason for using DRSs as our semantic representations is the existence of the Parallel Meaning Bank (PMB; Abzianidze et al., 2017), a multilingual corpus of DRS-annotated texts in English, Dutch, Italian, and German, to which we aim to contribute.

2.1 DRT in the PMB

The PMB makes a number of specific choices with regard to its DRS representations, which we endeavour to follow. Firstly, it represents DRSs not as graphical boxes, but as machine-readable text files, in a clausal format (van Noord et al., 2018a). An example PMB-style DRS and its corresponding translation into the clausal format is shown in Figure 2. Each clause begins with the label of a DRS (a 'box', hence the `b`), indicating where the condition is introduced. It then contains one of three types of condition: (1) a unary or binary predicate name, followed by its argument(s), as in `b1 scowl.v.01 e1`; (2) the explicit introduction of a discourse referent, as in `b1 REF e1`; or (3) a relation between DRSs, as in `b2 PRESUPPOSITION b1`, which states that the contents of DRS `b2` is a presupposition of `b1`.[4] Finally, the clause contains information about which word it originates from, and gives the character offsets of that word in square brackets.

As indicated in Figure 2, the PMB also represents presupposition, following the approach of Projective DRT (Venhuizen, 2015; Venhuizen et al., 2018). In the graphical representation, we indicate presupposed material with a prefixed asterisk, '`*`', since we flatten any embedded presupposition structure so that we just have a single box containing all presupposed material for the sentence (see Section 2.2 for more on our simplifications of the clausal format). On the clausal side, `b2 PRESUPPOSITION b1` means that DRS `b2` is a presupposition of DRS `b1`. A full list of PMB relations, including temporal relations (such as `TPR`, temporal precedence, used in Figure 2) is available on the PMB website.[5].

The PMB representations do not include any indication of number (singular vs. plural, etc.), nor of aspect, but they do contain detailed lexical semantic information, because each lexical concept, i.e. unary predicate, is identified with a WordNet synset (Fellbaum, 1998) indicating which

[3] Muskens (1994, 1996) provides a compositional interpretation of DRT using the lambda calculus, which also treats meaning itself as dynamic, thus uniting two divergent approaches within the dynamic semantics family.

[4] Other relations between DRSs used in the PMB follow the rhetorical relations of Segmented DRT (SDRT; see e.g. Asher and Lascarides 2003), but we do not use these in the DRASTIC corpus – see Section 2.2.

[5] https://pmb.let.rug.nl/drs.php (accessed 31 May 2023).

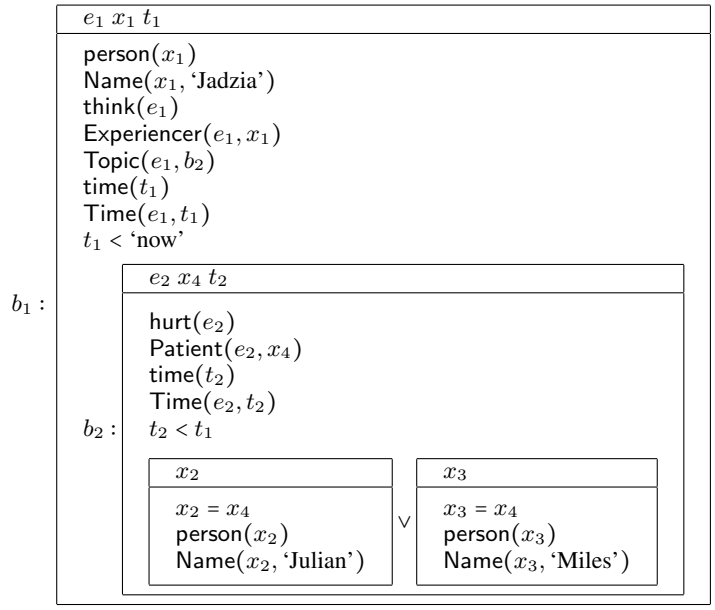
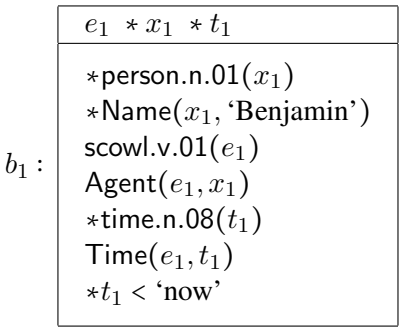


Figure 1: DRS for *Jadzia thought that Miles or Julian had been hurt*



(a) Graphical representation

```
b2 PRESUPPOSITION b1
b2 REF x1               % Benjamin [0...8]
b2 person.n.01 x1       % Benjamin [0...8]
b2 Name x1 "Benjamin"   % Benjamin [0...8]
b1 REF e1               % scowled [9...16]
b1 scowl.v.01 e1        % scowled [9...16]
b1 Agent e1 x1          % scowled [9...16]
b2 REF t1               % scowled [9...16]
b2 time.n.08 t1         % scowled [9...16]
b1 Time e1 t1           % scowled [9...16]
b2 TPR t1 "now"         % scowled [9...16]
```

(b) Clausal notation

Figure 2: Graphical vs. clause-based representation of a PMB-style DRS for the sentence *Benjamin scowled*

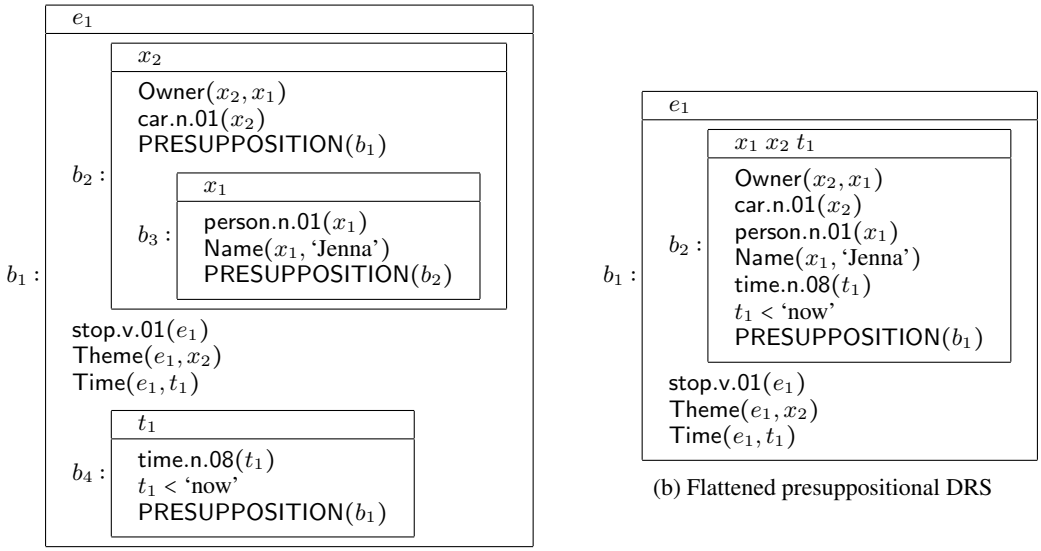
particular word sense is implicated. That is, the clause `b2 person.n.01 x1` indicates that the discourse referent `x1` falls under the first nominal sense of the lexeme PERSON listed in WordNet, i.e. a human being, as opposed to a body or the grammatical category (senses 2 and 3).

2.2 Simplifications

In general, the DRASTIC corpus follows the PMB annotation style, to allow the transfer of tools and techniques developed for the PMB, and in particular to provide test data involving longer sentences for the evaluation of parsers trained on the PMB. However, there are two areas in which we have chosen to simplify the PMB scheme in DRASTIC.

Firstly, we flatten DRSs by removing extraneous presuppositional sub-DRSs. To see what this means, consider the sentence *Jenna's car stopped*. Here we have (at least) three distinct existential presuppositions: the possessive construction presupposes the existence of Jenna's car; the proper noun *Jenna* itself introduces a presupposition that someone called 'Jenna' exists; and the past tense presupposes the existence of some time before the present. In PMB, this would result in three separate presuppositional DRSs, with two related directly to the main, outer DRS, and one related indirectly, via another of the presuppositional DRSs. This is shown in Figure 3a. In our own representations, all presuppositional material that originates in a given DRS is collapsed into a single sub-DRS, as shown in Figure 3b.

Since presuppositional material is ultimately not interpreted where it originates, but at the level to which it projects (on presupposition projection in DRT, see Venhuizen et al. 2018), this move is harmless with respect to the content of the DRSs in question. We lose track of which presuppositions



(a) Unflattened presuppositional DRS

(b) Flattened presuppositional DRS

Figure 3: Unflattened vs. flattened DRS for *Jenna's car stopped*

originated together, but this is not essential for interpretation. Moreover, when it comes to evaluating DRS parsing, we avoid many cases where logically equivalent DRSs are identified as distinct, owing to inconsequential differences in presupposition structure, which will then inappropriately suppress performance scores for DRS parsers.

This move also has the major advantage of making the representations easier for contemporary general-purpose neural networks to learn in the first place. As van Noord et al. (2018b, 619) observe, "DRSs are recursive structures and thus form a challenge for sequence-to-sequence models because they need to generate a well-formed structure and not something that looks like one but is not interpretable". By collapsing largely extraneous structure, we reduce one major source of difficulty for sequence-to-sequence models in producing DRSs.

The second simplification that we make is to eliminate rhetorical/discourse relations from our representations. This is more destructive than our first change since some such relations are genuinely informative (e.g. EXPLANATION). However, by far the most common relation in the PMB is CONTINUATION, the semantics of which reduces to conjunction, meaning that nothing is lost by eliminating it. Annotation of such rhetorical relations is also rather more subjective than other aspects of semantic annotation, which can inevitably lead to inconsistencies within or between annotators. Finally, removing these relations once again results in flatter DRSs, and so also serves to aid machine learning of DRS parsing.

However, since our corpus is too small to train a parser on our simplified format, model training must still rely on the PMB training set. Since most sentences there are very short, the structures that we simplify are unlikely to arise in large numbers; nevertheless, to make sure that the annotations are compatible, we provide a script that flattens PMB-style annotations as described above. This can be used to flatten the PMB data before training (to train a parser directly on this simplified format) or to flatten the output of a parser trained on the PMB directly. In Section 4, we report the results of some experiments using this second approach.

3 The corpus

3.1 The texts

The DRASTIC corpus consists of four sub-corpora: three entire documents from the biographical section of GUM, and one selection of shorter sentences drawn from different sub-parts of the GUM corpus.

The three biographical texts are Wikipedia articles relating to Czech composer Antonín Dvořák (`GUM_bio_dvorak`), YouTuber Jenna Marbles (`GUM_bio_marbles`), and translation theorist Eugene Nida (`GUM_bio_nida`), while the short texts corpus contains sentences 6–19 words long from 6 academic articles included in the 216 texts of the GUM corpus (the specific texts are shown in Table 1). Table 2 gives details about the size of the sub-corpora. 'Tokens' in this table refers to orthographic words separated by whitespace or hyphens,

GUM_academic_art
GUM_academic_census
GUM_academic_eegimaa
GUM_academic_enjambment
GUM_academic_epistemic
GUM_academic_games

Table 1: GUM texts from which the short-texts corpus draws

Sub-corpus	Sentences	Tokens	UD tokens
dvorak	28	668	678
marbles	43	842	926
nida	46	878	917
short-texts	40	512	539
TOTAL	157	2900	3060

Table 2: Size breakdown of the DRASTIC corpus

and to some punctuation characters (., ,, !, ?, ;). The UD tokenisation used in the GUM CoNLL-U files is more morphosyntactically motivated (e.g. possessive *'s* is separated from its host), and as such gives a larger number.

The major contribution of our corpus is that the sentence length distribution is more evenly spread and has a far wider range than that of the PMB data (especially the test set). For instance, the median sentence length in our corpus is 17, compared to 8 in the PMB data overall, and 6 in the PMB test set. The full distributions are shown in Figure 4, while Table 3 gives some further descriptive statistics about sentence length across the (sub-)corpora.

(Sub-)corpus	Median	Mean	St.dev.
dvorak	23	23.9	9.68
marbles	17	19.6	12.4
nida	18	19.1	11.1
short-texts	13	12.8	4.29
DRASTIC (all)	17	18.5	10.6
PMB (all)	8	10.0	9.53
PMB (test only)	6	6.60	2.08

Table 3: Sentence length across (sub-)corpora

Although it only has a modest number of sentences, the DRASTIC corpus nevertheless also manages to exemplify a range of complex linguistic phenomena, including negation, modal expressions, meta-linguistic usage, appositions, relative clauses, complement clauses, and a variety of other kinds of multi-clausal structure.

3.2 The annotation procedure

In the first instance, our annotation procedure follows that of the PMB as described in Abzianidze et al. (2017).[6] Our texts were uploaded to the PMB, where they were automatically analysed on several layers: tokenisation, CCG parsing, semantic tagging and WordNet sense selection. With this information, the Boxer system (Bos, 2008, 2015) then automatically produces a DRS representation for the sentence. All layers were subject to manual correction by trained annotators, and annotations were harmonised through weekly meetings and subsequent retagging of texts. This was done for around 1,000 sentences. For the 157 sentences released in the current version of the corpus, all sentences were also checked by the authors of this paper, and this process will continue.

The PMB interface imposes compositionality, in the sense that the final representation cannot be edited; only the representation of the tokens can be changed, and Boxer will then assemble a new representation of the sentence. While this is theoretically desirable, it can be practically limiting. Consider the sentence *She paid $800 rent by working various jobs, like bartending, working at a tanning salon, blogging, and go-go dancing at nightclubs.* Because this is a sequence of coordinated gerund VPs, Boxer produces disjoint DRSs connected by the discourse relation CONTINUATION. By manual intervention, we can instead make sure that the gerunds are coordinated to form one complex event, which bears an Instrument role to the matrix event.

To deal with such complicated cases, we therefore exported our data from the PMB and manually corrected remaining errors. Because we wanted to factor out anaphora resolution as a separate task, we exported the data with no anaphora resolved. However, all cases of sentence-internal anaphoric reference were noted, and we distribute the data in two versions: with and without anaphoric resolution. The former is the standard of the PMB and was used in our subsequent experiments. Unfortunately, it is not easy to represent cross-sentential anaphoric references when each DRS represents one sentence only; for that the DRSs must be merged or connected by discourse relations. This is a task for full-fledged discourse parsing, which we do not

[6] We are grateful to the PMB team, in particular Johan Bos and Rik van Noord, for helping us with both technical and linguistic issues in using the PMB interface.

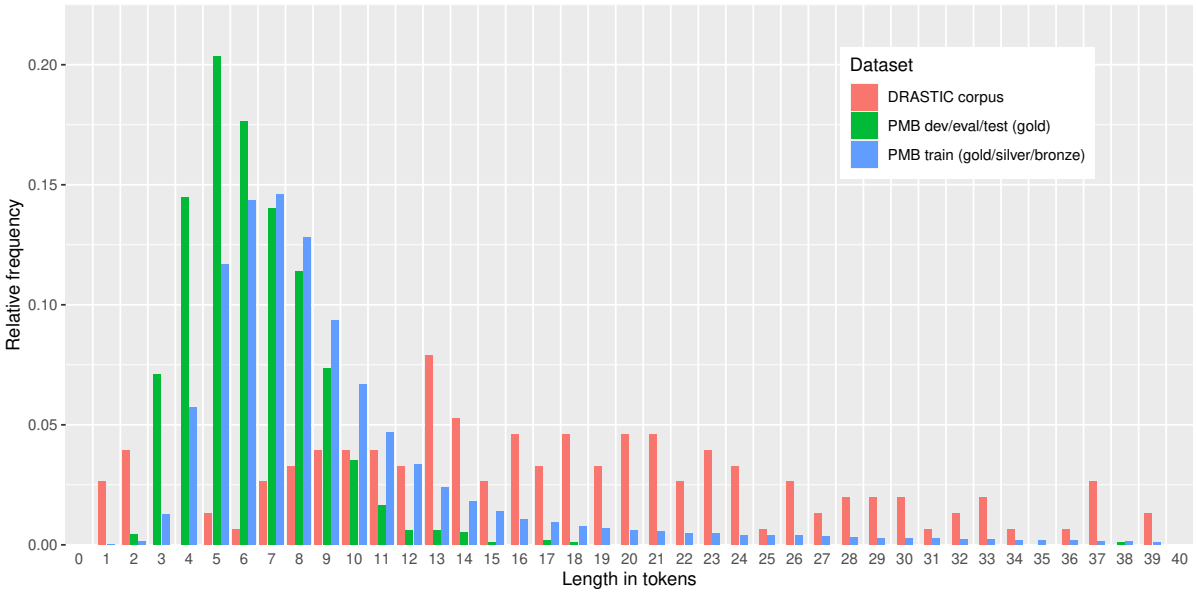


Figure 4: Length distribution in PMB 4.0.0 datasets compared to our corpus

attempt here. However, it is worth noting that this is an area where the additional annotation layers of the GUM corpus will be particularly useful. In this instance, the discourse annotation layer, based on Rhetorical Structure Theory (Mann and Thompson, 1988; Taboada and Mann, 2006), may aid in, for example, reconstructing the SDRS rhetorical/discourse relations which DRASTIC omits.

3.3 The format

Each of the two versions of the corpus, with and without anaphoric resolution, is provided as a set of files, one for each sentence, named after their GUM sent_id. Each file contains the raw text of the sentence and its clausal DRS annotation. We make the connection from the DRS annotation to both the original text and the GUM UD format explicit by decorating clauses in our data not only with character offsets, as shown in Figure 2, but also with UD token offsets, taken from the CoNLL-U files. This indicates which word(s) the clause in question originates from.

4 Modelling results

4.1 State of the art DRS parsing

Work on DRS parsing has recently involved applying deep neural networks. The majority of the work in this area (van Noord et al., 2018b; van Noord, 2019; Evang, 2019) has used sequence-to-sequence (seq2seq) LSTMs (Hochreiter and Schmidhuber, 1997). Table 4 presents recently reported performances of DRS parsing on the PMB datasets, along with the best results from our seq2seq experiments (Yıldırım and Haug, 2023), which, unlike previous work, also reports results on PMB 4.0.0.[7] We trained this state-of-the-art parser following the design principles used by van Noord et al. (2020), but instead of an LSTM we used transformer-based encoders and decoders. Here, we report the results obtained by using bert_base_cased as a frozen encoder along with a non-pretrained (randomly initialized) transformer as the decoder (12 layers, 12 attention heads per layer, using the Wordpiece tokenizer (Wu et al., 2016) used by the input (encoder) for the output as well).

The results in Table 4, with F1 scores in the high 80s/low 90s, clearly leave room for improvement, but do suggest that DRS parsing is a relatively straightforward task for current systems. The results are better, for example, than state-of-the-art parsing for Abstract Meaning Representation (AMR; Langkilde and Knight, 1998), which is in the low to mid 80s (Bai et al., 2022). This is surprising, because the expressive power of AMR is strictly less than that of DRT (Bos, 2016), and because the PMB DRSs capture many phenomena that AMR ignores, particularly involving scope.

However, there is reason to believe that DRS parsing as evaluated on the PMB test set understates the difficulty of the task. One issue that was

[7] Poelman et al. (2022) report performances of parsing Discourse Representation Graphs (DRG), a simpler form of DRSs, using PMB 4.0.0.

	PMB 2.2.0		PMB 3.0.0		PMB 4.0.0			
	dev	test	dev	test	dev	test	eval	DRASTIC
van Noord et al. (2020)	86.1	88.3	88.4	89.3	–	–	–	–
Liu et al. (2021)	–	88.7	–	–	–	–	–	–
Yıldırım and Haug (2023)	87.5	89.2	89.8	90.3	88.1	89.0	86.9	36.2

Table 4: Recently reported F1 scores for PMB 2.2.0, 3.0.0, and 4.0.0 datasets, and our result for DRASTIC

noticed by van Noord et al. (2020, 4594f.) is that, unsurprisingly, all models in their experiments performed worse as sentences got longer. In this context, the short length of the sentences in the PMB test set becomes especially noteworthy. The distribution of sentence lengths in the PMB was already shown in Figure 4. We see that it is very different between the training set and the dev/eval/test sets. As noted above, this is because the latter only include data that have been fully corrected manually – generally very short sentences – whereas the training set also contains data with no or only partial manual disambiguation, and those sentences are much longer. This mismatch is in itself a potential problem and may be the reason why several teams have fine-tuned their models on only the gold data of the training set, which has a similar length distribution to that of the test set.

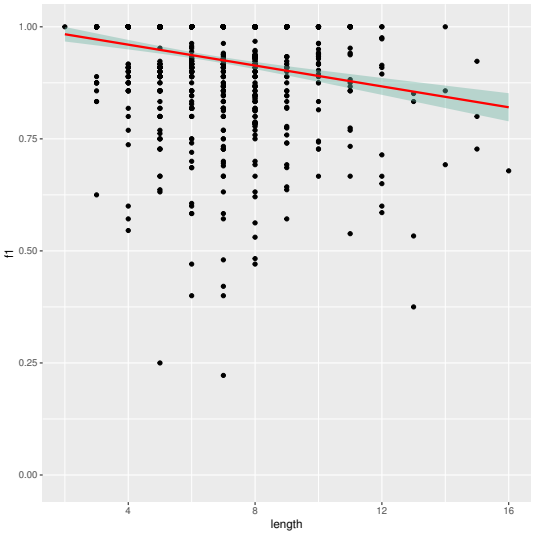


Figure 5: Performance vs. length in the PMB test set obtained by using the model reported under PMB 4.0.0 in Table 4 by Yıldırım and Haug (2023)

4.2 DRS parsing and sentence length

More worrying than the mismatch between training and evaluation is the overall short length of the sentences in the PMB dataset. We observe that sentences longer than 10 tokens are very rare. This is quite different from what one encounters in most genres of running text. Owing to the small range of sentence lengths in the PMB test set, the deleterious effect of increased length noted by van Noord et al. (2020) is only weakly felt there. The correlation between sentence length and F1 score in the PMB test set has a Pearson's r value of -0.21 ($p < 5 \times 10^{-10}$), a trend shown in Figure 5 (with regression line and 95% confidence intervals). In the DRASTIC data, with its more varied sentence lengths, the correlation with F1 scores is slightly more pronounced, as shown in Figure 6 (Pearson's $r = -0.29$, $p < 4 \times 10^{-4}$). Nevertheless, it is still fairly weak. Although longer sentences may confuse the transformer architecture by virtue of their length alone (because there was little or no data with the same positional encodings in the training phase), linguistic complexity (e.g. the presence of negation or other scopal operators, along with embedded structures) is another, semi-orthogonal source of difficulty, which will affect performance independently of length. Of course, the two are not entirely unrelated, since longer sentences also tend to be linguistically more complex (especially in terms of sentential embedding), exhibiting more structures that are rarely seen in the training data.

4.3 Performance on our dataset

Since our data structures are simplified ('flattened') compared to the PMB annotations, as described in Section 2.2, we transform the output of our parsers, which are trained on the original PMB data. This is done automatically in three steps:[8]

1. Removing discourse relations: Each clause of the form `x REL y`, where `REL` is one of `CONTINUATION`, `CONTRAST`, `ELABORATION` or `EXPLANATION`, is eliminated. All occurrences of the box variable `x` are replaced by `y` in all clauses.

[8]The script to perform this transformation has been made available along with the data at https://github.com/Universal-NLU/DRASTIC.

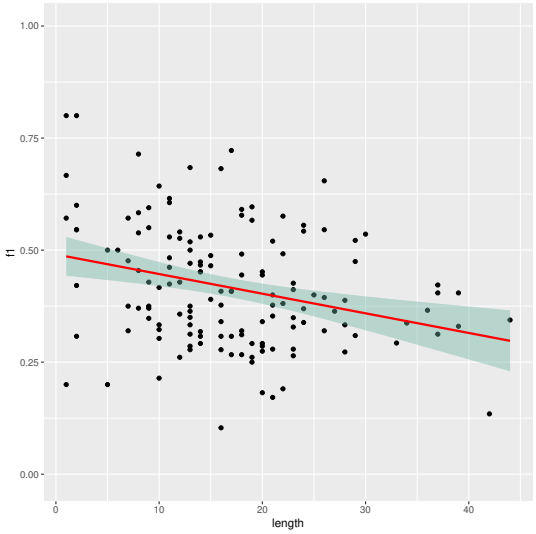


Figure 6: Performance vs. length in the DRASTIC corpus obtained by using the model reported under PMB 4.0.0 in Table 4 by Yıldırım and Haug (2023)

2. Flattening recursive presuppositions: for all occurrences of pairs of clauses of the form `x PRESUPPOSITION y`, `y PRESUPPOSITION z`, we remove the first clause and replace all occurrences of the box variable `x` by `y`.

3. Grouping presuppositions: for all occurrences of clause pairs `x PRESUPPOSITION y`, `z PRESUPPOSITION y`, we remove the first clause and replace all occurrences of the box variable `x` by `z`.

For the purposes of this paper, we perform these transformations on the output of the DRS parser before measuring performance on our dataset. This allows us to use the same model both on PMB data (with unflattened output) and on our data (with flattened output). As an alternative, it would be possible to train the model on flattened PMB output, so that the model will have seen the simplified structures directly during training; we leave this for future research.

We saw in Figure 6 the performance of our best model across sentences of different lengths in the DRASTIC corpus. Often for longer sentences the output of the model contains far fewer clauses than the gold data, suggesting an effect of length alone. But the model also performs much worse on DRASTIC than on the PMB in general, as witnessed by the low F1 score of 36.2 shown in Table 4.[9] Partly, this is because our dataset is more linguistically complex than the PMB. Sentences involving negation, for example, cause particular problems, and the negative meaning is often absent from the model output. Interaction between scopal elements such as negation and modality is also difficult: for the sentence *While the impact of a translation may be close to the original, there can be no identity in detail*, the model incorrectly stacks the possibility operators and flips the scope of negation and possibility, so that the meaning of the second clause becomes "it is possible that it is possible that there is no identity in detail", while in *This is, perhaps, not the best example of the technique* . . . , the negation disappears altogether.

Linguistic complexity cannot be the whole story, however. There are also unusual errors such as names that occur in our data but not in the PMB being incorrectly rendered in the parser output: e.g. the name "Marbles" becomes 'georgia strawberry', 'margau', 'margis', and 'name'. It is surprising to see such behaviour in a parser that performs so well on the PMB test set. This might indicate that the models overfit on peculiarities of the PMB.[10] A deeper investigation into what causes this drop in performance is clearly required – for example, one could replace names in the DRASTIC corpus with frequently-occuring names in the PMB to see if this improves performance. Whatever the exact origins of these deficiencies turn out to be, we believe our more varied data can contribute to more robust DRS parsers, especially as DRASTIC grows in size.

5 Summary

We have presented a new dataset, the DRASTIC corpus, which contains PMB-style DRSs annotated over sentences with more realistic lengths than the original PMB dataset, and which is, accordingly, much more of a challenge for state-of-the-art parsers. We hope that this will lead both to a more realistic assessment of the difficulty of DRS parsing and, in the longer term, to the development of more robust models.

[9] And this is true even when length is held constant: for example, in the PMB, the majority of sentences of length 8 are parsed to an F1 score of 0.75 or higher, whereas in our data, only one "sentence", of length 1, gets a score at this level.

[10] As an anecdotal example, we can mention that that 15-20% of the sentences across the PMB subsets contain the proper name *Tom*.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Nicholas Asher. 1986. Belief in Discourse Representation Theory. *Journal of Philosophical Logic*, 15:127–189.

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 277–286. College Publications.

Johan Bos. 2015. Open-domain semantic parsing with boxer. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 301–304, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Johan Bos. 2016. Squib: Expressive power of Abstract Meaning Representations. *Computational Linguistics*, 42(3):527–535.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The groningen meaning bank. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2, pages 463–496. Springer.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The logic of decision and action*, pages 81–120. University of Pittsburgh Press, Pittsburgh, PA.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: an electronic lexical database*. MIT Press, Cambridge, MA.

Dag T. T. Haug. 2014. Partial dynamic semantics for anaphora: compositionality without syntactic coindexation. *Journal of Semantics*, 31(4):457–511.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hans Kamp. 1981a. Événements, représentation discursive et référence temporelle. *Langages*, 64:39–64. [English version published in 2017 in *Semantics & Pragmatics* 10(2), available online here: https://doi.org/10.3765/sp.10.2].

Hans Kamp. 1981b. A theory of truth and semantic representation. In Jeroen Groenendijk, Theo M. B. Janssen, and Martin Stokhof, editors, *Formal methods in the study of language*, pages 277–322. Mathematical Centre Tracts, Amsterdam.

Hans Kamp. 1990. Prolegomena to a structural account of belief and other attitudes. In C. Anthony Anderson and Joseph Owens, editors, *Propositional attitudes – the role of content in logic, language, and mind*, pages 27–90. CSLI Publications, Stanford, CA.

Hans Kamp, Josef van Genabith, and Uwe Reyle. 2011. Discourse Representation Theory. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of philosophical logic*, 2nd edition, volume 15, pages 125–394. Springer, Berlin.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic*. Kluwer, Dordrecht.

Hans Kamp and Uwe Reyle. 2019. Discourse Representation Theory. In Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors, *Semantics: theories*, pages 321–384. De Gruyter, Berlin.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 704–710, Montreal, Quebec, Canada. Association for Computational Linguistics.

Alex Lascarides and Nicholas Asher. 1993. Temporal interpretation, discourse relations, and commonsense entailment. *Linguistics and Philosophy*, 16:437–449.

Jiangming Liu, Shay B. Cohen, Mirella Lapata, and Johan Bos. 2021. Universal discourse representation structure parsing. *Computational Linguistics*, 47(2):445–476.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Reinhard Muskens. 1994. A compositional Discourse Representation Theory. In Paul Dekker and Martin Stokhof, editors, *Proceedings of the 9th Amsterdam Colloquium*, pages 467–486. ILLC, Amsterdam.

Reinhard Muskens. 1996. Combining Montague semantics and discourse representations. *Linguistics and Philosophy*, 19:143–186.

Rik van Noord. 2019. Neural boxer at the IWCS shared task on DRS parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018a. Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018b. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4587–4603, Online. Association for Computational Linguistics.

Terence Parsons. 1990. *Events in the semantics of English: a study in subatomic semantics*. MIT Press, Cambridge, MA.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022. Transparent semantic parsing with Universal Dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Maite Taboada and William C. Mann. 2006. Rhetorical Structure Theory: looking back and moving ahead. *Discourse Studies*, 8(3):423–459.

Noortje J. Venhuizen. 2015. *Projection in discourse: a data-driven formal semantic analysis*. Ph.D. thesis, University of Groningen.

Noortje J. Venhuizen, Johan Bos, Petra Hendriks, and Harm Brouwer. 2018. Discourse semantics with information structure. *Journal of Semantics*, 35(1):127–169.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint: 1609.08144.

Ahmet Yıldırım and Dag Trygve Truslew Haug. 2023. Experiments in training transformer sequence-to-sequence DRS parsers. In *Proceedings of the 15th International Conference on Computational Semantics (IWCS 2023)*, Nancy, France. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

UMR Annotation of Multiword Expressions

Julia Bonn[1], **Andrew Cowell**[1], **Jan Hajič**[3]
Alexis Palmer[1], **Martha Palmer**[1], **James Pustejovsky**[2], **Haibo Sun**[2]
Zdenka Urešová[3], **Shira Wein**[4], **Nianwen Xue**[2], **Jin Zhao**[2]
[1]University of Colorado, Boulder, [2]
Brandeis University [3]Charles University, Prague, [4]Georgetown University
Corresponding author: julia.bonn@colorado.edu

Abstract

Rooted in AMR, Uniform Meaning Representation (UMR) is a graph-based formalism with nodes as concepts and edges as relations between them. When used to represent natural language semantics, UMR maps words in a sentence to concepts in the UMR graph. Multiword expressions (MWEs) pose a particular challenge to UMR annotation because they deviate from the default one-to-one mapping between words and concepts. There are different types of MWEs which require different kinds of annotation that must be specified in guidelines. This paper discusses the specific treatment for each type of MWE in UMR.

1 Introduction

Uniform Meaning Representation (UMR) (Gysel et al., 2021) is a graph-based formalism designed to represent natural language semantics. It is based on Abstract Meaning Representation (AMR) (Banarescu et al., 2013), but is enriched and extended in accordance with typological principles to account for linguistic uniformity and variation across a wide range of languages of the world, from languages like Arabic, Chinese, and English that have a large population of speakers, to languages like Arapaho, Kukama, Navajo, and Sanapana with a relatively small number of speakers. Expanding on AMR, UMR also includes a document-level representation that represents linguistic relations that go beyond sentence-boundaries, such as coreferential relations and temporal and modal dependencies.

Like AMR, the basic building blocks of a UMR graph are concepts and relations, with concepts typically mapping to words in a sentence and relations representing how those words are related semantically. UMR concepts are typically lemmas or sense-disambiguated lemmas, but they can also be abstract concepts that do not map to specific word tokens and are instead inferred from the context of the sentence. Common in a UMR graph are subgraphs that represent predicate-argument structures in which the predicate is the parent and its arguments are its children. The relations between a predicate and its arguments are typically the semantic roles that each argument plays with respect to the predicate, but they can also be other types of semantic relations. A UMR example containing a multiword expression (MWE) is provided in (1):

(1) They are willing to throw America under the bus.

```
(w / will-02
    :aspect State
    :modstr FullAff
    :Arg0 (p / person
        :ref-person 3rd
        :ref-number Plural)
    :Arg1 (t2 / throw-under-bus-08
        :Arg0 p
        :Arg1 (c / country
            :name (n / name
                :op1 "America"))))
```

The mapping between UMR concepts and words in a sentence is complex. While most UMR concepts map to single words, there are also UMR concepts as in (1) that map to multiple words, in this case four words. The opposite is true as well where one word can map to multiple UMR concepts, which is often the case in polysynthetic languages like Arapaho (Gysel et al., 2021).

In this paper, we draw from a broad range of

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 99–109

annotated examples from different languages to discuss the properties of different types of MWEs and how they can be annotated in UMR. Because UMR annotation occurs in two stages– one called stage 0 for languages without rolesets, and one called stage 1 for languages with rolesets– we discuss differences in the strategies used for MWE annotation at both stages. Here we adopt an operational definition of MWE in the context of UMR annotation. When multiple words map to a single concept in UMR, these words form an MWE. Since they are so common, formulating a consistent approach for representing MWEs in UMR is critical to the success of UMR as a representation. This requires, first of all, that we have a good understanding of what types of MWEs exist in languages of the world, and then design a consistent set of guidelines to direct their annotation in UMR.

The MWEs that we consider in this paper include light verb constructions (LVCs) (Section 2), MWEs without a strong figurative interpretation (Section 3), non-consecutive multiword expressions that occur in certain constructions (Section 4), idioms (Section 5), proverbs (Section 6), and two-part allegorical sayings (Section 6.1). These categories distinguish MWEs by how they are handled in UMR, both in terms of how tokens are incorporated into UMR graphs, and whether/how figurative meaning is conveyed through the UMR schema. We expect that this work will be useful for UMR annotation in the future, and is broadly relevant to studies of abstract meaning in formal semantic representations.

2 Light Verb Constructions

Light verb constructions take the form of a semantically-light verb with a nominal predicate as its object. The arguments are selected by the nominal predicate rather than the light verb, although the light verb can contribute proto-roles as well as aspectual interpretations of the construction as a whole. Following AMR, the UMR concept used to represent the LVC in a graph is derived from the nominal predicate, with the light verb contributing to the aspectual annotation for the predicate. In the English example in (2), the light verb "make" pairs with the nominal predicate "break", which has arguments for an entity in motion (literal or abstract) and a destination. In the UMR graph, the light verb is glossed by the appropriate PropBank sense *break-20*, along with its arguments[1].

(2) The children made a break for the playground.

```
(b / break-20
   :Arg0 (c / child
      :refer-number Plural)
   :Arg2 (p / playground)
   :aspect Performance
   :modstr FullAff)
```

LVCs are also common in Chinese In (3), for example, the light verb 获得 ("get") takes a nominal predicate 认可 ("acceptance") as its object and together they form an LVC in which the nominal predicate selects the arguments and the light verb contributes to a *Performance* aspectual value, meaning the acceptance event has been successfully completed.

(3) 这 一 方法 获得认可 。
this one method get acceptance .

"This method got accepted."

```
(x1 / 认可-01
   :aspect Performance
   :modstr FullAff
   :Arg1 (x2 / 方法
      :mod (x3 / 这)))
```

With the Spanish LVC "dar miedo" (scare, lit. give someone fear), UMR annotation omits the light verb and substitutes the whole construction with the roleset for the verb "asustar," which also means *to scare*[2].

(4) Le di miedo.
him I.gave fear

"I scared him."

```
(a / asustar-01
   :Arg0 (p / person
      :refer-person 1st
      :refer-number Singular)
   :Arg1 (p / person
```

[1] From the English PropBank Lexicon: https://github.com/propbank/

[2] In accordance with Spanish roleset conventions (Wein et al., 2022)

:refer-person 3rd
:refer-number Singular)
:aspect Performance
:modstr FullAff)

In Czech, the same phenomenon exists[3]. Example (5) shows the LVC "vznést připomínku" ("to comment" or "to remind," lit. "raise a reminder"), which is similar to the previous Spanish example in that the LVC can be represented in the UMR graph with a roleset for a synonymous verb "připomenout" (lit. "to remind")[4].

(5) Vznesl poté připomínku.
Raised after-that a-comment.

"He then made a comment."

(p / připomenout-01
:Arg0 (p2 / person
:ref-person 3rd
:ref-number Singular)
:temporal (p3 / poté)
:aspect Performance
:modstr FullAff)

There are also cases in Czech, however, where the LVC is complex and cannot be replaced by a single related verbal roleset without some of the meaning being lost. An example is "projít zkouškou ohněm" (lit. "go_through test [by]fire", experience ordeal by fire). Here, the nominal portion is itself idiomatic. The preferred UMR approach in such cases is to use an MWE predicate reflecting the whole construction.

(6) Prošel zkouškou ohněm.
go_through exam by_fire

"He passed the ordeal by fire."

(z / zkoušet-ohněm-01
:Arg1 (i / individual-person
:ref-person 3rd
:ref-number Singular)
:aspect Performance
:modstr FullAff)

[3]The usual Czech linguistic terminology uses the term "compound [verb] phrases."

[4]In Examples 5 and 6 that since Czech is a pro-drop language, subject personal pronouns are usually dropped because all person and number information is provided on the verb thanks to grammatical agreement rules. However, UMR graphs still provide a node for this argument to enable co-reference.

3 MWEs Without Strong Figurative Interpretation

Some MWEs do not have a strong figurative interpretation, i.e., any figurative interpretation can still be derived from the parts. This category includes everything from fully-fixed MWEs like complex function words (Constant et al., 2017) to semi-fixed MWEs (Sag et al., 2002) like Verb Particle Constructions (VPCs) and 'decomposable' idioms (Sag et al., 2002), as well as everything in between. Fixed MWEs are consecutive and do not vary at all, while semi-fixed MWEs can allow a wide array of variations. Where lexical variation is allowed, it ranges from inflection to token addition, alternation, or elision. Some semi-fixed MWEs include a core set of tokens that provide the key semantics and are never altered beyond inflection. These can be combined with additional tokens that can be replaced or modified to contextualize the MWE's semantics. Some semi-fixed MWEs have a fixed word order, and others, such as many VPCs, allow variable word order.

UMR has a similar array of treatments for annotating these MWEs. At any annotation stage, core tokens can be concatenated to form the concept used in the UMR graph. If the MWE is clausal, requires sense-disambiguation, or has a distinct argument structure, a new roleset can be created for it.

3.1 Fixed MWEs

A fixed MWE (Sag et al., 2002) is an unmodifiable, consecutive sequence of word tokens that maps to a single UMR graph concept. Many fixed MWEs are complex function words like *by-and-large* in English (Constant et al., 2017). These do not have argument structures and do not need anything beyond a single concatenated node in the graph (e.g., (b / by-and-large)).

Many predicating prepositional phrases are also fixed (*in_love*, *in_arrears*). Since these are clausal and take at least one argument, they are treated as a predicate in UMR, using the UMR participant roles during stage 0 annotation (7) and being assigned a roleset in stage 1 annotation (8).

(7) "The bank was in arrears."

```
(i / in-arrears
    :theme (b / bank)
    :aspect State
    :modstr FullAff)
```

(8) "John was in love with Mary."

```
(l / love-01
    :Arg0 (p / person
        :name (n / name :op1 "John"))
    :Arg1 (p2 / person
        :name (n2 / name :op1 "Mary")
    :aspect State
    :modstr FullAff)
```

During roleset creation, predicating PPs can either be assigned a unique roleset or included with one that already exists and is semantically/etymologically related. For example, *in-arrears* would be assigned its own roleset since it has no corresponding verbal or nominal roleset (i.e., in-arrears-01, :Arg1-entity owing money, :Arg2-amount, :Arg1-money owed), but, *in-love* can be included as part of love-01, which already exists for verbal/nominal *love*.

3.2 Verb Particle Constructions

Verb Particle Constructions are semi-fixed MWEs that include a specific verb and one or more specific particles. The verb may be inflected, and many VPCs allow the verb and particle to be split up. In UMR, VPCs are represented as a concatenated predicate. In English, they are included as their own rolesets, separate from the base verb.

(9) The sheep ate the flowers up.

```
(e / eat-up-02
    :Arg0 (s / sheep)
    :Arg1 (f / flower
        :refer-number Plural)
    :aspect Performance
    :modstr FullAff)
```

Fixed and semi-fixed MWEs like these are highly language-specific, and different languages may express similar concepts with different types of MWEs. For example, VPCs in English often correspond to verb compounds in Chinese (Sun et al., 2023) as the particle is generally considered to be a verb. However, the UMR annotation is similar, with the verb compound as a whole is treated as a UMR concept.

(10) 小 羊 把 花 吃掉 了 。
little sheep BA flower eat up ASP .

"The little sheep ate up the flower."

```
(x5 / 吃掉-01 ["eat up']
    :aspect Performance
    :modstr FullAff
    :Arg0 (x2 / 羊 ["sheep"]
        :mod (x1 / 小 ["little"]))
    :Arg1 (x4 / 花 ["flower"]))
```

Interestingly, Czech has no VPCs in the proper sense. Instead, some verbs (in one or more of their senses) can require a particular preposition as the only acceptable form of expression of one of its arguments. However, the preposition is not considered to be part of the predicate, even if neither the meaning of the verb and the preposition, nor the preposition and the noun phrase which form a PP, is compositional. An example is "zmínit se o něčem" ("to mention sth", lit. "to mention about sth(.locative-case)"), where the preposition "o" ("about"), requiring locative case, loses its meaning of "aboutness". Such constructions are thus not considered MWEs from the predicate point of view, and they would get the following UMR annotation:

(11) zmínit se o něčem
mention [refl.] about something

"to mention something"

```
(z / zmínit-se-01
    :Arg1 (n / něco)
    :aspect Perfective
    :modstr FullAff)
```

The example also shows that verbs with reflexive particles (such as "se" in this case), having a "frozen" meaning which is required to be used in the sentence simply as a [mostly discontinuous] part of the predicate, are always considered an MWE.

3.3 Semi-fixed MWEs

Many MWEs fall into the semi-fixed category, being semi-compositional, modifiable, and figuratively transparent, while not being entirely literal. Such MWEs are also handled in UMR

by concatenating core tokens in a graph predicate and using either participant roles or numbered arguments associated with a unique (or related) roleset. In some cases, there are a cluster of closely related MWEs that can be grouped together into a single roleset. In (12), a roleset "keep-eye-out-02" is used for instances of *"keep an eye out for", "keep an eye open for",* and *"keep your eyes peeled for"*:

(12) "He was keeping [an eye out]/[an eye open]/[his eyes peeled] for potholes."

```
(k / keep-eye-out-02
    :Arg0 (p / person
        :refer-person 3rd
        :refer-number Singular)
    :Arg1 (p2 / pothole
        :refer-number Plural)
    :aspect Activity
    :modstr FullAff)
```

Inside the roleset file, a slot/token structure shows that the first slot is always *keep*, the third slot is always *eye* (or its plural), and the third slot can take variants *out, open*, and *peeled*. (See Figure (1) for more on slots.)

4 Non-consecutive Constructions

Another challenge in UMR annotation lies in representing constructions that are cued by a non-consecutive (and sometimes inter-clausal) sequence of words. They are also MWEs in the sense that they consist of multiple words, but the words may be predominantly function words, and the meaning may not be derived from any one particular word in the sequence. Following AMR, UMR uses abstract rolesets to represent the established semantics of such constructions. For example, "the more ... the more ..." (*the Xer, the Yer*) is annotated with an abstract roleset called *correlate-91*, which take as arguments the two predicates that are correlated:

(13) The more I studied, the less I understood.

```
(c / correlate-91
    :Arg1 (m / more
        :Arg3-of (h / have-quant-91
            :Arg1 (s / study-01
                :Arg0 (i / i)
                :aspect Activity
                :modstr FullAff)))
    :Arg2 (l / less
        :Arg3-of (h2 / have-quant-91
            :Arg1 (u / understand-01
                :Arg0 i
                :aspect State
                :modstr FullAff))))
```

Chinese has a similar construction that also maps to the abstract concept *correlate-91*:

(14) 时间 越 临近 , 我就 越 感到
time more get close , I then more feel
幸福 。
happy

"The closer the time comes, the happier I will be"

```
(c / correlate-91
    :Arg1 (x3 / 临近-01
        :aspect Performance
        :modstr FullAff
        :Arg0 (x1 / 时间))
    :Arg2 (x8 / 感到-01
        :aspect Performance
        :modstr FullAff
        :Arg1 (x9 / 幸福)
        :Arg0 (i / individual-person
            :ref-person 1st
            :ref-number Singular)))
```

The same example could be given for Czech, with the correlation expressed via a pair of pronouns "čím ..., tím ...", for example "Čím důležitější schůzka, tím jsem nervóznější.", lit. "The more important the meeting [is], the more nervous I am."

5 Idioms - MWEs that Have a Figurative Interpretation

Idioms are MWEs that are ambiguous between a literal meaning and a figurative interpretation, where ambiguity can be resolved in context. Depending on how MWEs are interpreted, they are mapped to UMR concepts in different ways. If the literal meaning is intended given the context, such an expression can be represented compositionally in UMR. However, if the figurative meaning is intended, UMR concepts are created by concatenating the word tokens in the MWE, similar to how fixed or semi-fixed expressions are handled. We illustrate this with the expression "jump

on the bandwagon" in English, with a graph for the literal interpretation in (15) and one for the figurative interpretation (16). With the idiomatic interpretation, an English PropBank roleset jump-on-bandwagon-09 is used, shown in Figure (1).

(15) "He jumped on the bandwagon." *(lit.)*

```
(j / jump-03
    :Arg0 (p / person
        :ref-person 3rd
        :ref-number Singular)
    :Arg1 (b / bandwagon)
    :aspect Performance
    :modstr FullAff)
```

(16) "He jumped on the bitcoin bandwagon." (*'he joined the bitcoin boom'*)

```
(j / jump-on-bandwagon-09
    :Arg0 (p / person
        :ref-person 3rd
        :ref-number Singular)
    :Arg1 (b / bitcoin)
    :aspect Performance
    :modstr FullAff)
```

Rolesets for idiomatic MWEs are able to be quite descriptive about the relationships between the MWE's tokens and between the elements in the literal and figurative frames, as illustrated in Figure (1).

First, numbered roles are provided for participants in the idiomatic frame as well as any modifiers the expression can take. Here, the Arg0 of jump-on-bandwagon-09 corresponds to the same agent that appears in (15). Additionally, the expression *'jump on the bandwagon'* is modifiable, so the roleset provides an Arg1 for any phrase that might be used to modify 'bandwagon'. The idiomatic meaning of the expression is *'to join in with others who are following a certain fad'*, and it conveys this by evoking historical imagery of political parade-goers jumping onto the wagon that carried the band at the front of the parade. In current use, speakers identify the 'bandwagon' in the expression with a fad, and tell us what the fad is by modifying that token syntactically.

Next in the roleset, the tokens are identified and labeled with slot position, part of speech, and syntactic head. Then, two parallel graphs are given that use the slot labels (A-D), the numbered arguments (N-ARG0 and N-ARG1), and token values to map between the literal frame and the metaphorical frame. The token 'jump' in slot A is equated with the 'jump-03' roleset (physical jumping) in the literal interpretation and with the 'join-in-05' roleset (joining a group) in the idiomatic interpretation. The token 'bandwagon' in slot D appears as the destination argument of jump-03 in the literal frame and is equated with 'people following a fad' in the figurative frame. N-ARG0 is equated with the literal jumper and the figurative fad-joiner. Lastly, N-ARG1 tells us what kind of fad is being discussed in the figurative frame (as in *He joined the bitcoin boom*).

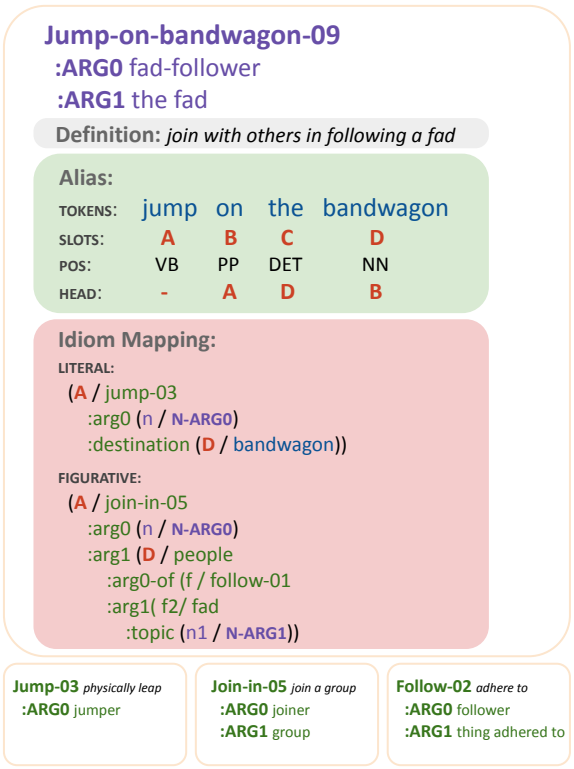


Figure 1: Roleset for jump-on-bandwagon-09 with token breakdowns and mappings between literal and figurative frames.

5.1 Chinese Idioms

Chinese idioms, known in Chinese as *xiyu*, can also have literal or figurative interpretations and are annotated in a similar manner to English. For example, the Chinese expression 炒鱿鱼 ('stir fry squid') can have a literal or figurative meaning depending on the context. In (17), 炒鱿鱼 should be interpreted literally and

compositionally, mapping to two UMR concepts, one for 炒 (stir-fry) and one for 鱿鱼 (squid):

(17) 他 在 厨房 里 炒 鱿鱼 。
he at kitchen inside stir-fry squid .

"He was stir-frying squid in the kitchen."

```
(x5 / 炒-01
    :Arg0 (i / individual-person
        :ref-person 1st
        :ref-number Singular)
    :Arg1 (x6 / 鱿鱼)
    :place (x4 / 厨房)
    :aspect Activity
    :modstr FullAff)
```

More often, however, the expression has a figurative interpretation of 'fire from a job', as in (18). In this case, it is treated as an MWE that maps to a single UMR concept.

(18) 他 被 那 个 公司 炒 了
he BEI that CL company stir-fry ASP
鱿鱼 .
squid .

"He was fired from that company."

```
(x6 / 炒鱿鱼-00 ["fire"]
    :Arg1 (i / individual-person
        :ref-person 3rd
        :ref-number Singular)
    :Arg0 (x5 / 公司 ["company"]
        :mod (x3 / 那 ["that"]))
    :aspect Performance
    :modstr FullAff)
```

Chinese Chengyu Chengyu (成语) are idioms that obey the grammar of ancient Chinese but have become fixed expressions in modern Chinese. The literal meaning of such expressions often describes a scenario in the past that no longer applies today. This is illustrated in (19), where the underlined expression is an idiom meaning *step by step*:

(19) 他 因 初次 创业 ,
he because first time start business ,
所以 凡事 都 步步为营 。
therefore everything all step by step .

"Because he started his own business for the first time, he took everything step by step."

```
(x9 / 步步为营-00
    :Arg0 (i / individual-person
        :ref-person 3rd
        :ref-number Singular)
    :mod (x7 / 凡事 ["everything"])
    :mod (x8 / 都 ["all"]))
    :cause (x4 / 创业-01
        :Arg0 i
        :mod (x3 / 初次 ["first time"])
        :aspect Performance
        :modstr FullAFF)
    :aspect Activity
    :modstr FullAFF)
```

5.2 Spanish Idioms

Idiomatic phrases in Spanish can also be taken literally or figuratively. For example, the Spanish phrase "con las manos en la masa" (meaning caught red-handed or in the act, lit. "with hands in the dough") could refer to an actual dough thief caught because of their messy fingers, or some other crime a person is caught committing.

(20) él fue atrapado con las manos en la
he was trapped with the hands in the
masa.
dough.

"he was caught red-handed."

```
(a / atrapar-01
    :Arg0 (p / person
        :ref-person 3rd
        :ref-number Singular)
    :manner (c / con-manos-en-masa-01
        :Arg0 p))
```

The UMR graph captures the idiomatic meaning by modifying a 'caught' verb with a roleset for the 'caught in the act' sense of 'with hands in the dough'.

The Spanish idiom "el que corta el bacalao" which denotes the person in charge (literally *he who cuts the cod*) is handled similarly, with a roleset for the idiomatic interpretation of 'cut cod':

(21) el que corta el bacalao
he who cuts the cod

"person in charge."

```
(c / cortar-bacalao-01
    :Arg0 (p / person
```

:ref-person 3rd
:ref-number Singular))

5.3 Czech Idioms

In Czech, the same literal vs. figurative interpretations exist for many expressions, with most being interpreted idiosyncratically or figuratively (i.e., non-compositionally). In such cases, using a single synonymous predicate is generally preferred, although the choice of predicate can be context-dependent. (22) shows two different solutions used for the idiom "jít z kopce" (lit. "go downhill"): the first one uses a predicate corresponding to the literal meaning of the idiom (but which could also be interpreted figuratively), whereas the second solution (with the predicate "chudnout-01") assumes that the context implies that it means "he was getting poorer."

(22) šlo to s ním z kopce.
went it with him down [the]hill.

He deteriorated/became poorer/became more sick/became asocial/was getting worse results at work/...

(j / jít-z-kopce-01
:Arg0 (p / person
:refer-person 3rd
:refer-number Singular))

(c / chudnout-01
:Arg0 (p / person
:refer-person 3rd
:refer-number Singular))

5.4 Arapaho Idioms

Arapaho is an agglutinating polysynthetic language and as such has very few constructions that might qualify as multiword expressions. Still, Arapaho has idiomatic constructions that need to be treated in a way that allows literal interpretations to be separated from figurative ones. In (23), *'nih3iikoncebeit'* is a word/phrase (lit. 'a ghost shot him [with an arrow]') that means that someone gave the person in question a disease. In the morphological breakdown, */3iikon-/* is a noun-incorporating preverb that refers to the ghost. While Arapaho might not normally include such preverbs as part of the predicate in a UMR graph (instead, using just */ceb/*, 'shoot', as the predicate), in the case of idiomatic expressions like this, the graph predicate includes it. A roleset for this phrase would include numbered arguments for the shooter (disease-giver) and victim as well as the disease. A source/target mapping in the roleset file would link */3iikon-/* (ghost) to the numbered argument for disease. This roleset is separate from the roleset for literal shooting (ceb-01), but is included in the same file.

(23) nih- 3iikon- ceb -eit
PAST- ghost- shoot -4/3

"Someone gave him a disease."

(x / 3iikonceb-01
:Arg0 (p / person
:refer-person 3rd
:refer-number Singular)
:Arg1 (p2 / person
:refer-person 3rd
:refer-number Singular)
:Arg2 [implicit-for-coref]
:aspect Performance
:modstr FullAff)

6 Proverbs

Like idioms, proverbs also have a literal and figurative interpretation. Unlike idioms, however, proverbs are often self-contained sentences with all participants of the predicates filled, and it is hard to construct alternative contexts in which the proverbs can be interpreted literally. Since they tend to be longer than idioms and their literal meaning can be constructed compositionally in UMR, we annotate proverbs with an abstract roleset called *proverb-91*, which takes two arguments. The first argument is required and is annotated compositionally; the second argument will be described in the next section. We illustrate a standard proverb that uses the Arg1 with a Chinese proverb in (24).

(24) 山 高 皇帝 远
mountain high emperor far away

"The mountains are high and the emperor is far away."

(p / proverb-91
:Arg1 (a / and
:op1 (x2 / 高-01 ["high"]
:Arg0 (x1 / 山 ["mountain"])

:aspect State
:modstr FullAFF)
:op2 (x4 / 远-01
:Arg0 (x3 / 皇帝 [“emperor”])
:aspect State
:modstr FullAFF)))

6.1 Xiehouyu, or two-part allegorical sayings

Xiehouyu, also known as a two-part allegorical saying (Lai, 2008), is common in Chinese and other Asian languages; similar forms can be found in other languages as well. Xiehouyu consists of two parts–an antecedent that is a highly allegorical and figurative expression, and a consequent that provides an explanation for the antecedent. We represent such sayings with *proverb-91* as well, using Arg1 for the antecedent and Arg2 for the consequent:

(25) 你 这 是 大炮 打 蚊子 ——
you this be cannon shoot mosquito -
小题大做
solving small problem with big action

“(By doing this), you are shooting cannon at mosquitoes - making too much out of something small.”

(c / proverb-91
:Arg1 (x5 / 打-02 [“shoot”]
:Arg0 (i / individual-person
:ref-person 2nd
:ref-number Singular)
:Arg1 (x6 / 蚊子 [“mosquito”])
:instrument (x4 / 大炮 [“cannon”])
:aspect Habitual
:modstr FullAff)
:Arg2 (x8 / 小题大做-01 [”make too much out of something small”]
:aspect Habitual
:modstr FullAff))

The English saying *“Life is like a box of chocolates– you never know what you’re going to get.”* follows the same format, and two-part proverbs (where the second part is optional) are also present in Russian (Dahl, 2000).

7 Related & Future Work

MWEs have always been a thorny issue for computational linguistics (Sag et al., 2002) and have been studied from various perspectives, from linguistic modeling (annotation) to automatic identification. The existence of a series of workshops focusing on MWEs (Markantonatou et al., 2020; Cook et al., 2021; Bhatia et al., 2022) attests to the interest of a large community of researchers. For many European languages, multiword expressions including verbal ones have been tackled in the PARSEME project[5] (Rosén et al., 2015; Savary et al., 2017). For Czech, previous work on identification and extraction of verbal (and other) MWEs from treebanks is described in (Uresova et al., 2013; Urešová et al., 2016; Bejček et al., 2017).

This work addresses the representation of MWEs in UMR, resolving many of the issues surrounnding many-to-one word-to-concept mappings. It builds on previous work respecting light verb constructions in multilingual PropBanks (Hwang et al., 2010) and AMR annotation of certain English constructions (Bonial et al., 2018), and expands it to include additional MWEs in multiple languages. We have discussed different types of MWEs that present challenges to UMR annotation and presented solutions for their treatment. Users with an existing valency lexicon or PropBank may wish to undertake creating new MWE rolesets based on recommendations we have outlined. Forthcoming work will build on the strategies outlined here as we tackle one-to-many word-to-concept mappings.

Acknowledgments

This work is supported by grants from the CNS Division of National Science Foundation (Awards no: NSF_2213805, NSF_2213804, NSF_IIS 1764048, NSF_1763926 RI) entitled “Building a Broad Infrastructure for Uniform Meaning Representations” and ”Developing a Uniform Meaning Representation for Natural Language Processing”, respectively. The work on Czech has been supported by the UMR project No. LUAUS23283 supported by the Czech Ministry of Education, Youth and Sports (MSMT CR). It has used data provided by the LRI LINDAT/CLARIAH-CZ, Projects No. LM2018101 and LM2023062, supported by the MSMT CR. The work on Spanish is supported by a Clare Boothe Luce Scholarship.

[5] https://typo.uni-konstanz.de/parseme

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Eduard Bejček, Jan Hajič, Pavel Straňák, and Zdeňka Urešová. 2017. Extracting verbal multiword data from rich treebank annotation. In *Proceedings of the 15th International Workshop on Treebanks and Linguistic Theories (TLT 15)*, pages 13–24, Bloomington, IN, USA. Indiana University, Bloomington, Indiana University, Bloomington.

Archna Bhatia, Paul Cook, Shiva Taslimipoor, Marcos Garcia, and Carlos Ramisch. 2022. Proceedings of the 18th workshop on multiword expressions@ lrec2022. In *Proceedings of the 18th Workshop on Multiword Expressions@ LREC2022*.

Claire Bonial, Bianca Badarau, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Tim O' Gorman, Martha Palmer, and Nathan Schneider. 2018. Abstract meaning representation of constructions: The more we include, the better the representation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Paul Cook, Jelena Mitrović, Carla Parra Escartín, Ashwini Vaidya, Petya Osenova, Shiva Taslimipoor, and Carlos Ramisch. 2021. Proceedings of the 17th workshop on multiword expressions (mwe 2021).

VI Dahl. 2000. Proverbs of the russian people. moscow: Russian language media.

Jens Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Timothy J. O'Gorman, Andrew Cowell, W. Bruce Croft, Chu-Ren Huang, Jan Hajic, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a uniform meaning representation for natural language processing. *Künstliche Intell.*, 35:343–360.

Jena D Hwang, Archna Bhatia, Claire Bonial, Aous Mansouri, Ashwini Vaidya, Nianwen Xue, and Martha Palmer. 2010. Propbank annotation of multilingual light verb constructions. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 82–90.

Huei-ling Lai. 2008. Understanding and classifying two-part allegorical sayings: Metonymy, metaphor, and cultural constraints. *Journal of Pragmatics*, 40(3):454–474.

Stella Markantonatou, John Philip McCrae, Jelena Mitrović, Carole Tiberius, Carlos Ramisch, Ashwini Vaidya, Petya Osenova, and Agata Savary. 2020. Proceedings of the joint workshop on multiword expressions and electronic lexicons. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*.

Victoria Rosén, Gyri Losnegaard, Koenraad De Smedt, Eduard Bejček, Agata Savary, Adam Przepiórkowski, Petya Osenova, and Verginica Mititelu. 2015. A survey of multiword expressions in treebanks. In *14th International Workshop on Treebanks and Linguistic Theories (TLT 2015)*, pages 179–193, Warszawa, Poland. IPI-PAN, IPIPAN.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings 3*, pages 1–15. Springer.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, Antoine Doucet, Kübra Adalı, Verginica Mititelu, Eduard Bejček, Ismail El Maarouf, Gülşen Cebiroğlu Eryiğit, Luke Galea, Yaakov Kerner, Chaya Liebeskind, Johanna Monti, Carla Escartín, Jolanta Kovalevskaitė, Simon Krek, Lonneke Plas, Cristina Aceta, Itziar Aduriz, Jean-Yves Antoine, Greta Attard, Kirsty Azzopardi, Loic Boizou, Janice Bonnici, Mert Boz, Ieva Bumbulienė, Jael Busuttil, Valeria Caruso, Manuela Cherchi, Matthieu Constant, Monika Czerepowicka, Anna Santis, Tsvetana Dimitrova, Tutkum Dinç, Hevi Elyovich, Ray Fabri, Alison Farrugia, Jamie Findlay, Aggeliki Fotopoulou, Vassiliki Foufi, Sara Galea, Polona Gantar, Albert Gatt, Anabelle Gatt, Carlos Herrero, Uxoa Inurrieta, Glorianna Jagfeld, Milena Hnátková, Mihaela Ionescu, Natalia Klyueva, Svetla Koeva, Viktória Kovács, Taja Kuzman, Svetlozara Leseva, Sevi Louisou, Teresa Lynn, Ruth Malka, Héctor Martínez Alonso, John McCrae, Helena Caseli, Ayşenur Miral, Amanda Muscat, Joakim Nivre, Michael Oakes, Mihaela Onofrei, Yannick Parmentier, Caroline Pasquer, Maria Buono, Belem Sanchez, Annalisa Raffone, Renata Ramisch, Erika Rimkutė, Monica-Mihaela Rizea, Katalin Simkó, Michael Spagnol, Valentina Stefanova, Sara Stymne, Umut Sulubacak, Nicole Tabone, Marc Tanti, Maria

Todorova, Zdeňka Urešová, Aline Villavicencio, and Leonardo Zilio. 2017. Annotated corpora and tools of the PARSEME shared task on automatic identification of verbal multiword expressions (edition 1.0).

Haibo Sun, Yifan Zhu, Jin Zhao, and Nianwen Xue. 2023. UMR annotation of chinese verb compounds and related constructions. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+ NLP, GURT/SyntaxFest 2023)*, pages 75–84.

Zdeňka Urešová, Eduard Bejček, and Jan Hajič. 2016. Inherently pronominal verbs in czech: Description and conversion based on treebank annotation. In *Proceedings of the 12th Workshop on Multiword Expressions (ACL 2016)*, pages 78–83, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL), Association for Computational Linguistics (ACL).

Zdenka Uresova, Jan Hajic, Eva Fucikova, and Jana Sindlerova. 2013. An analysis of annotation of verb-noun idiomatic combinations in a parallel dependency corpus. In *Proceedings of the 9th Workshop on Multiword Expressions*, pages 58–63, Atlanta, Georgia, USA. Association for Computational Linguistics.

Shira Wein, Lucia Donatelli, Ethan Ricker, Calvin Engstrom, Alex Nelson, Leonie Harter, and Nathan Schneider. 2022. Spanish Abstract Meaning Representation: Annotation of a general corpus. In *Northern European Journal of Language Technology, Volume 8*, Copenhagen, Denmark. Northern European Association of Language Technology.

MR4AP: Meaning Representation for Application Purposes

Bastien Giordano and **Cédric Lopez**
Emvista
Immeuble Le 610
10 rue Louis Breguet
34830 Jacou, France
{firstname.lastname}@emvista.com

Abstract

Despite the significant progress made in Natural Language Processing (NLP) thanks to deep learning techniques, efforts are still needed to model explicit, factual, and accurate meaning representation formalisms. In this article, we present a comparative table of ten formalisms that have been proposed over the last thirty years, and we describe and put forth our own, Meaning Representation for Application Purposes (MR4AP), developed in an industrial context with a definitive applicative aim.

1 Introduction

An efficient human-machine interface is one of the dreams of Artificial Intelligence (AI). In NLP, despite the dazzling progress of the last few years with the emergence of large language models, it is necessary to resort to formal representations of textual statements, so that the machine can structure its result and reason while providing the explanation.

The last thirty years have witnessed numerous formalism proposals, the most recent of which are Universal Conceptual Cognitive Annotation (UCCA, Abend and Rappoport, 2013), Abstract Meaning Representation (AMR, Banarescu et al., 2013), Uniform Meaning Representation (UMR, Van Gysel et al., 2021) and BabelNet Meaning Representation (BMR, Navigli et al., 2022). The adoption of a meaning representation formalism is not a trivial choice, especially in an industrial context, as is the case of the authors of this paper. In this context, it is required that a formalism be explicit and factual while maximizing the richness and accuracy of the most semantically salient linguistic phenomena (Abzianidze and Bos, 2019).

The contribution of this article is twofold. On the one hand, we facilitate the comparison of ten formalisms *via* a table (section 2). To the best of our knowledge, although it is one of the shortcomings expressed by the community (Abend and Rappoport, 2017), only Koller et al. (2019) and Žabokrtský et al. (2020) have established such a comparison[1][2], but their studies and ours do not overlap much. Moreover, we include the most recent formalisms. It is on this basis that we present our own, Meaning Representation for Application Purposes (MR4AP, section 3), which we are already exploiting in an industrial context with an applicative focus. In section 4, we put forward three examples of our representation choices, while section 5 describes the first version of an annotated corpus following our formalism as well as a first small-scale manual annotation effort, accompanied by the annotation guidelines. Before concluding, we discuss some limitations and prospects for future work (section 6).

2 Meaning Representations comparison

In this section, we compare ten meaning representation formalisms, with which we compare our own (see Table 1). Each of the formalisms occupies a column (from oldest to newest), while the rows represent some of the linguistic features and phenomena that are fully covered (✓), partially covered (#), or not covered at all (empty space). The rows are grouped into five clusters, respectively related to genericity, structure, explicitness, various intra- and inter-sentence relations, and diversity of annotated attributes. For this last characteristic, we symbolize it from the least rich (+) to the richest (+++).

Partial coverage (#) has several meanings. It can mean that a feature is covered, but only in one of the formalism's extensions. This is the

[1] Other works address and compare in a more or less extensive way a number of formalisms (Bonn et al., 2023; Hershcovich et al., 2020; Pavlova et al., 2022; *inter alia*).

[2] Flanigan et al. (2022) have also prepared a tutorial in which they present and compare several meaning representation formalisms, but this tutorial was unknown to the authors at the time of writing and was not yet available.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 110–121
June, 2023.

	DRT	UNL	MRS	PDT	GMB	UCCA	AMR	UDS	UMR	BMR	MR4AP
Multilingual	✓	✓	✓	#	#	✓	#	✓	✓	✓	✓
Invariance	#	#	#		#	✓	✓		✓	✓	✓
Multi-sentence	✓			✓	✓	✓	#		✓	#	✓
P-A structure	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Named rel.		✓	#	#	✓	#	#		#	✓	✓
Sem. typing	✓	✓	#	#	#		✓	✓	✓	✓	✓
Anaph. & coref.	✓			✓	#	#	#		✓	✓	✓
Event coref.				✓		#	#		✓	✓	✓
Temporal rel.	✓	#	#	✓	✓	✓	#	#	✓	#	✓
Discourse rel.	✓	#	#	✓	#	✓	#		#	#	✓
Modal rel.									✓		
Attr. richness	++	+++	++	+++	+	+	+	+	+++	++	++

Table 1: Comparative table of meaning representation formalisms

case, for example, for the multilingual nature of the Prague Dependency Treebank's Tectogrammatical Layer (PDT-TL, whose original version was only for Czech (Mikulová et al., 2006) before covering English as well (Hajič et al., 2012)) and Groningen Meaning Bank (GMB, which was only for English (Basile et al., 2012) before being extended to German, Dutch, and Italian with Parallel Meaning Bank (PMB, Abzianidze et al., 2017)). The need to use an extension also holds true for AMR, a formalism for which various works have aimed at making it multi-sentence (O'Gorman et al., 2018) or enriching the annotated attributes (such as tense and aspect in Donatelli et al., 2018). This is also the case for UCCA with respect to coreference resolution (entities and events), which is dependent on a layer over the foundational one (Prange et al., 2019b)[3].

Partial coverage can also mean that a feature is covered but only in a limited way. This is the case for relations that are numbered rather than named (`:ARG0`, `:ARG1`, etc., in AMR), for labels that are insufficiently fine-grained (UCCA), or when nodes carry the label that is traditionally assigned to the arcs (PDT-TL). This partial coverage mainly concerns the group of intra- and inter-sentence relations: when one of these relations is taken into account by the formalism, however this coverage is only realized at one of the two levels (for instance, UMR's discourse relations), we consider that it is incomplete.

The different formalisms proposed over the years diverge on many points and converge on others. One of the most salient points of divergence is the distance between the meaning representation and the surface syntactic form. The Czech school with the PDT-TL formalism is among those that remain closest to syntax. Several layers of annotation are superimposed, the highest of which being the so-called tectogrammatical layer (`t-layer`), which combines syntax and semantics. Many linguistic phenomena are encoded (grammatical tense, coreference and anaphora, semantic types), some of which are largely discarded by other formalisms (ellipsis, focus/topicalization). However, its obvious proximity to syntax means that complex sentences that are semantically similar, but whose main and subordinate clauses would have been inverted produce drastically different results (Abend and Rappoport, 2017).

The invariance of representations for semantically close segments, regardless of their syntactic configuration (active/passive voice, paraphrasing, cleft sentences), is a consensus feature. AMR, UMR and BMR are among the formalisms that adhere to it. All three belong to the non-anchored semantic graphs (*i.e.*, of type 2 according to the typology of Kuhlmann and Oepen, 2016), that is to say that there is no direct and explicit correspondence between the graph's nodes and the source tokens. AMR uses PropBank (Palmer et al., 2005), a resource whose concepts are represented by frames and whose relations are symbolized by an enumeration of arguments noted `:ARG0`, `:ARG1`, etc. This

[3]Other works have enriched UCCA at different levels: role labeling of core (Shalev et al., 2019) and non-core (Prange et al., 2019a) arguments based on the supersenses of Schneider et al. (2018), refinement of implicit argument types (Cui and Hershcovich, 2020), *inter alia*.

opacity has been preserved by UMR, but not by BMR, whose authors consider that it prevents an explicit understanding of the semantics attached to the relation. A selection of 25 of the 39 thematic roles of VerbNet (Schuler, 2005) were preferred.

Although these two resources (PropBank and VerbNet) are regularly chosen for relation labeling, other formalisms deviate from them. This is the case with UCCA, which exploits a smaller set of relations. UCCA is a multilingual formalism based on Basic Linguistic Theory (Dixon, 2010) that uses acyclic directed graphs (DAGs). Unlike AMR and its followers, UCCA is an anchored multi-layer formalism: for a given text, each token constitutes a leaf of the graph. The textual content is seen as a set of *Scenes* that can describe actions or states. Each *Scene* has a root node linked to the main relation (or main process) of the statement. To represent relations, the UCCA foundational layer has a dedicated set of only twelve labels, rendering the annotation process, according to the authors, quite simple, even for people without linguistic training. However, the semantics attached to a predicate's participants (all represented with the single label `A`) is far from fine-grained. In contrast, Universal Decompositional Semantics (UDS, White et al., 2016) is a formalism that does not use any discrete values to symbolize the relations between predicates and their arguments. Instead, the authors use proto-roles from Dowty (1991), which have numerical values appended to them. Instead of being labeled `Agent`, an argument can have a value related to its attributes `Awareness`, `Volition`, `Instigation`, etc. This representation, described as feature-based and opposed to traditional systems (White et al., 2020), has been extended to different phenomena, namely semantic typing of entities, factuality of events (Rudinger et al., 2018), genericity of entities and events (Govindarajan et al., 2019), and temporal relations between events (Vashishtha et al., 2019).

3 MR4AP's position

In this section, we focus on positioning MR4AP with respect to the other formalisms on the points that seem most salient to us.

Applicative aim. MR4AP is a formalism that has been designed with an industrial and, therefore, applicative aim. Although we base our choices on existing research works, we have made them with the requirement of being factual, meaning that the annotation should not be left to the subjective interpretation of the annotator. There should not be several possible annotations for the annotator to choose from. Therefore, despite the originality of their approach compared to other formalisms, we detach ourselves from UDS's choices of continuous representation, mainly because such representation using probabilities can make the annotation process complex and be difficult to assess accurately. Moreover, we move away from theoretical formalisms such as Discourse Representation Theory (DRT, Kamp et al., 1993) and Minimal Recursion Semantics (MRS, Copestake et al., 2005).

Genericity. MR4AP has been designed with *genericity* as its watchword. This applies both to the multilingual character of the representation and the invariance of the representations despite syntactic idiosyncrasies. Most recent formalisms aim at abstracting away from syntax, and MR4AP joins them on this point. Therefore, we detach ourselves from those that have a strong correlation with syntactic representations, as is the case of PDT-TL and UDS. On the same note, and although they are only notation variants (Oepen et al., 2019), the inverted arguments of AMR and its extensions (`:ARG0-of`) force parsers on the one hand to normalize relations (making graphs *de facto* multi-rooted), and on the other hand modify the graph, furthermore creating more cycles in supposedly acyclic graphs (Kuhlmann and Oepen, 2016). MR4AP being multi-rooted does not allow inverted arguments.

Explicitness. From our point of view, a meaning representation must be as explicit as possible. This explicitness is expressed at several levels. On the one hand, we agree with Di Fabio et al. (2019) on the need to name all relations between nodes: if a relation is not typed with a sufficiently specified label (UCCA), or is not usable without glossing (AMR/UMR), or is not represented by a discrete value (UDS), much of the semantics attached to the relationship is lost. Like BMR, MR4AP, therefore, uses a subset of VerbNet roles, to which some labels are added to specify temporal, spatial, discourse, and coreference relations. Likewise, it seems to us necessary to make entities' types as explicit as possible thanks to a label, mainly to avoid having to gloss their meanings.

Intra- and inter-sentence relations. We consider that a meaning representation would not be complete if it did not include the different rela-

tions that exist in a document at the intra- and inter-sentence levels. In this respect, MR4AP is close to UMR, because the latter includes a representation at the document level, although UMR's is parallel to the one at the sentence level while MR4AP's isn't. UMR's parallel document-level structure includes anaphora and coreference relations (between entities and between events), temporal relations between events, and modal relations, a representation unique to UMR and based on the work of Vigus et al. (2019). Thus, all discourse relations are excluded from their document-level representation, despite the carryover of the modal strength corresponding to the `:condition` and `:purpose` relations. MR4AP differs from UMR on several points regarding inter-sentence representation. On the one hand, there is no distinction between the two levels, which are perfectly inseparable. On the other hand, and this follows from this single structure, in addition to coreference and temporal relations, all discourse relations are represented at both levels, simply because they can occur in adjacent sentences (see section 4). Finally, modality is represented by an attribute linked to the predicate that is modified.

Attribute richness. Following Bonial et al. (2019), we believe that a meaning representation of the text must include a certain amount of information conveyed by the morphosyntax. Among this information, we can count grammatical tense, aspect, and number. It is precisely these three elements that are missing in AMR and that motivated BMR's authors to incorporate them in their formalism (Martínez Lorenzo et al., 2022), although in a minimal way. On the contrary, UMR adds a lot of complexity by introducing deep lattices, multiplying the possible labels for each phenomenon. That holds true in particular for aspectual values with twenty-three possible labels against two for BMR (`:ongoing +` and `:ongoing -`) and seven for MR4AP, based on UMR's work (`habitual`, `state`, `process`, `atelic process`, `activity`, `endeavor` and `performance`). This important multiplication of attributes and associated values is also visible for Universal Networking Language (UNL, Uchida et al., 1999) and PDT-TL. We prefer a smaller set of attributes while keeping those necessary for an objective and factual representation of the textual content.

4 MR4AP representation examples

In this section, we apply our formalism to represent three distinct examples. Each of these examples illustrates one or more parts of the formalism that we consider important.

4.1 Document-level representation and main points

The first example will be used to introduce the formalism in its broad outline. It will also allow us to demonstrate that a representation at the document level, taking into account all intra- and inter-sentence relations, is possible. It consists of the following three sentences:

1. *Luke and John are singing songs.*
2. *As a result, Mary cannot sleep.*
3. *She will reprimand them tomorrow morning.*

Predicate-argument structure. In Figure 1, the three main predicates (red squares with solid edges) are `vn:performance-26.7`, `vn:snooze-40.4`, and `vn:judgment-33`[4]. Each of these predicates is linked to its arguments by a thematic role (bold arcs). The conjunction of the proper nouns in (1) gives rise to the reification (red square with dotted edges) of an `:addition` node, whose arguments are the `:Agents` of the predicate `vn:performance-26.7`.

Inter-sentence relations. Inter-sentence relations are resolved at several levels. At the coreference level, the tokens *She* and *them* are linked to their respective antecedents (or to what symbolizes them), namely *Mary* and `:addition`, *via* the `:SameAs` relation. At the discourse level, the causal relation between the predicates `vn:performance-26.7` and `vn:snooze-40.4` is represented by the `:Cause` and `:Consequence` relations. At the temporal level, the predicate `vn:snooze-40.4` is linked by a relation `:TimeMax` to `vn:judgment-33`, *i.e.*, the former is realized before the latter.

Attributes. Each predicate and each entity has its own attributes (dotted arcs). The verbal predicates can have a modal value, an aspectual

[4]Even though those three are VerbNet's classes (hence the *vn:* prefix), MR4AP does not cling to one resource in particular. We consider that the formalism must remain at the conceptual level and that linking a specific resource to it would already be tantamount to instantiating it. This instantiation could be done from any resource, or even from any conjunction of resources, as is the case in Figure 2.

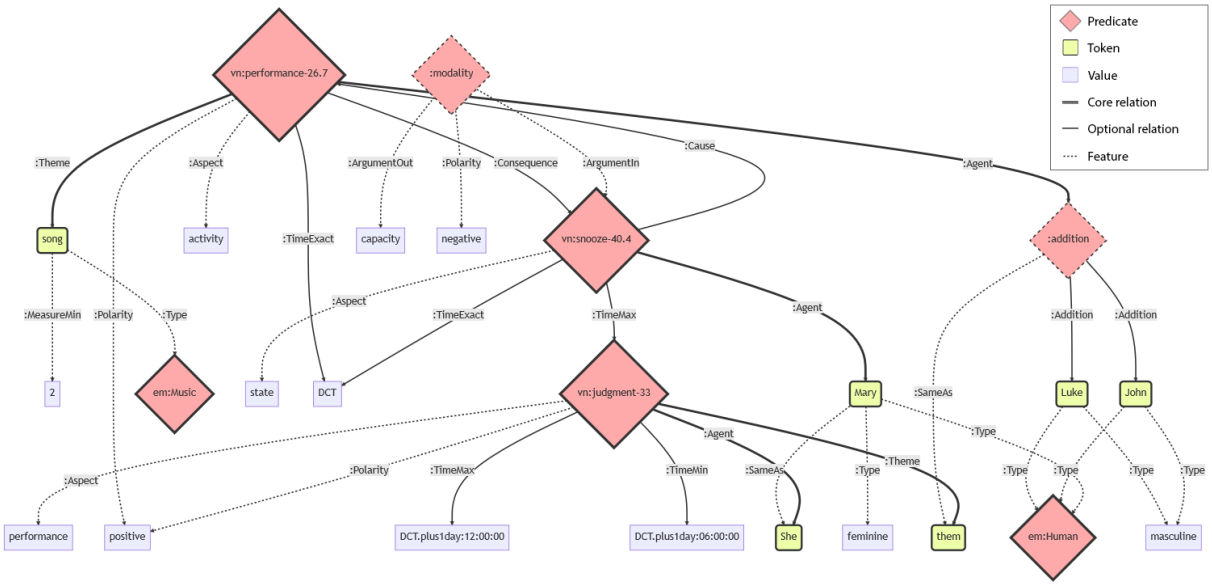


Figure 1: Document-level MR4AP representation example (see subsection 4.1). Note that all the attribute arcs are in fact by default reified, as is the case for the *:modality* node. Solely for readability reasons is it the only attribute node visibly reified in the graph. It is made so in order to take into account the negation's scope, hence the negative polar value for this node. The relations *:Argument*{*In,Out*} are empty relations meant to link a node to its value. They are used for every reified attribute node. To be perfectly clear, triples like *("Luke" :Type "masculine")* are in fact always two triples such as the following: *(:type :ArgumentIn "Luke")* and *(:type :ArgumentOut "masculine")*.

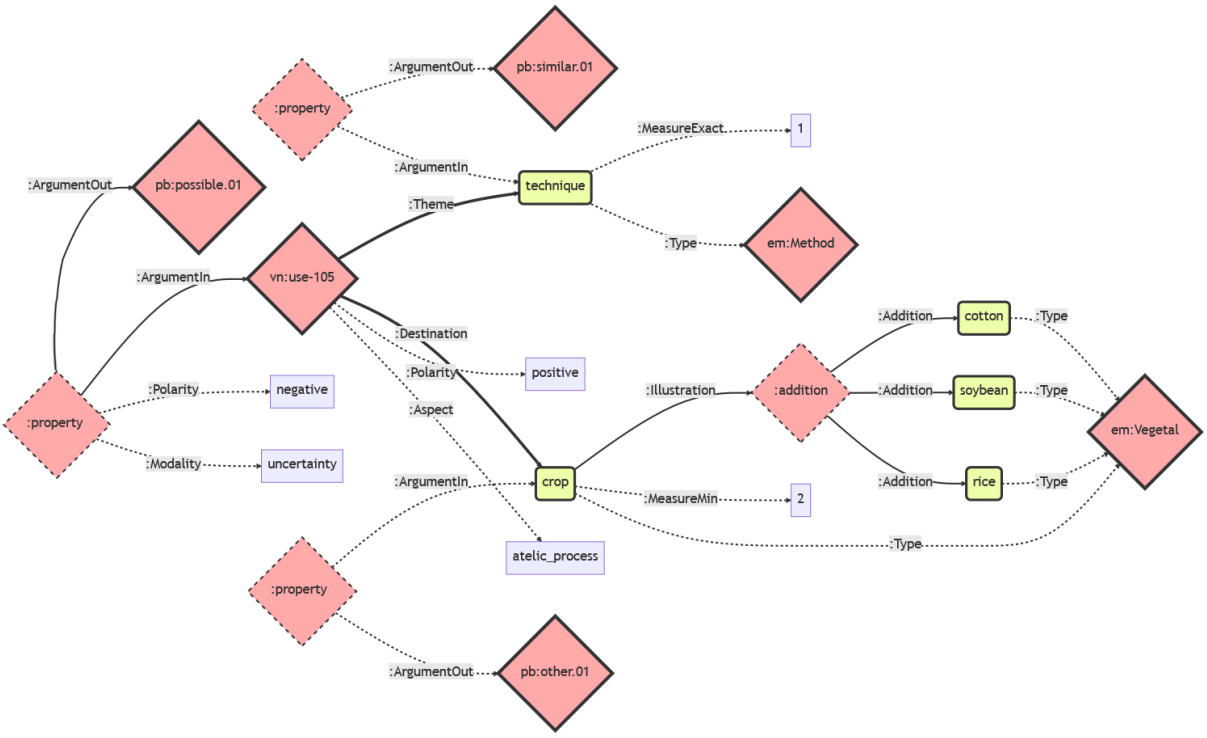


Figure 2: MR4AP representation example of MRP's running example (see subsection 4.2)

value, a polarity value, and temporal attributes (:Time{Min,Max,Exact,Fuzzy,Duration}). Nominal entities can have attributes related to their number, semantic type, and gender. Both semantic types (introduced by the em: prefix) and gender values are introduced with a :Type attribute.

4.2 MRP's running example

This subsection is dedicated to the representation of the running example used during the 2019 and 2020 Meaning Representation Parsing (MRP) shared tasks (Oepen et al., 2019, 2020). This will allow readers to more easily compare the frameworks that took part in those campaigns with our formalism. Here is the sentence:

4. *A similar technique is almost impossible to apply to other crops, such as cotton, soybean, and rice.*

This example was chosen because it presents a number of difficulties, namely a *tough* adjective *(impossible)*, a scopal adverb *(almost)*, and an appositive conjunction of more than two terms *(cotton, soybean, and rice)* illustrating a collection *(crops)* (Oepen et al., 2019).

***Tough* adjective.** *Tough* constructions (TCs) are a syntactic turn in which the logical object of an embedded non-finite verb is the main verb's syntactic subject (Hicks, 2009). In (4), the seemingly missing object of *to apply* is in fact the syntactic subject of *be (almost) impossible* (that is to say *technique)*. This can be paraphrased into two other configurations: either, acting as the subject, an expletive *it (it is almost impossible to apply a similar technique)* or an infinitival clause *(to apply a similar technique is almost impossible)*. To be as factual as possible and leave the annotator no choice, we always represent adjectives, whether attributive or predicative, with a :property attribute node linking the object and the adjective using the :Argument{In,Out} empty relations. Therefore, the attributive adjectives *other* and *similar*, which respectively trigger the pb:other.01 and pb:similar.01 nodes, are linked to *crop* and *technique via* :property nodes. As for *impossible*, it is treated in the same way. We consider that *to apply a technique is impossible* is similar to *the impossible application of a technique*. Thus, the two surface forms should produce the same graph. As a result, we link the pb:possible.01 node to vn:use-105 *via* a :property node. Also note that said node has a negative polar value triggered by its source token's prefix *(im-)*.

Scopal adverb. Regarding the scopal adverb *almost*, it modifies the adjective *impossible* and makes it uncertain. Consequently, the modal value uncertainty is added to the :property node linked to pb:possible.01. It should be remembered that the :Modality relation is in fact a reified node by default. Therefore, had the adverb been preceded by a negation particle (*i.e.*, *not almost impossible*), the reified :modality node would have had a negative polar value.

Enumeration in apposition. This representation does not differ from the conjunction of proper nouns in (1). An :addition node is reified, and each of the terms of the enumeration is linked to this node by an :Addition relation. The term referring to the collection of these examples is *crops*, and the :addition node is linked to it by the discourse relation :Illustration (*such as*).

4.3 Other difficult phenomena

The last example focuses on the representation of three arguably difficult elements: event coreference, interrogative sentences, and multiword expressions (MWEs), especially when they include event nominals. In addition, it helps to demonstrate the formalism's resistance to paraphrasing. The representation is the same for the following pair of paraphrases (see Figure 3):

5. *Who committed the murder of that police officer, and was it for revenge or for love?*
6. *Was this murder perpetrated out of revenge or out of love? And who killed that policeman?*

Event coreference. The event coreference appears in (5) between *(committed the) murder* and *it*, then in (6) between *murder (perpetrated)* and *killed*. In the same way that we represent coreference between entities in (1), we do not merge the nodes corresponding to each mention of the event, but rather join them with the :SameAs relation. The light verbs accompanying the event nominals are dropped from the representation. We discuss this further below.

Interrogative sentences. As for the representation of interrogative sentences, and especially that of unknown elements during the utterance of these sentences, we distinguish three types, to which are attached three different labels linked to the :Type attribute: polar questions (question-closed, whose answers are

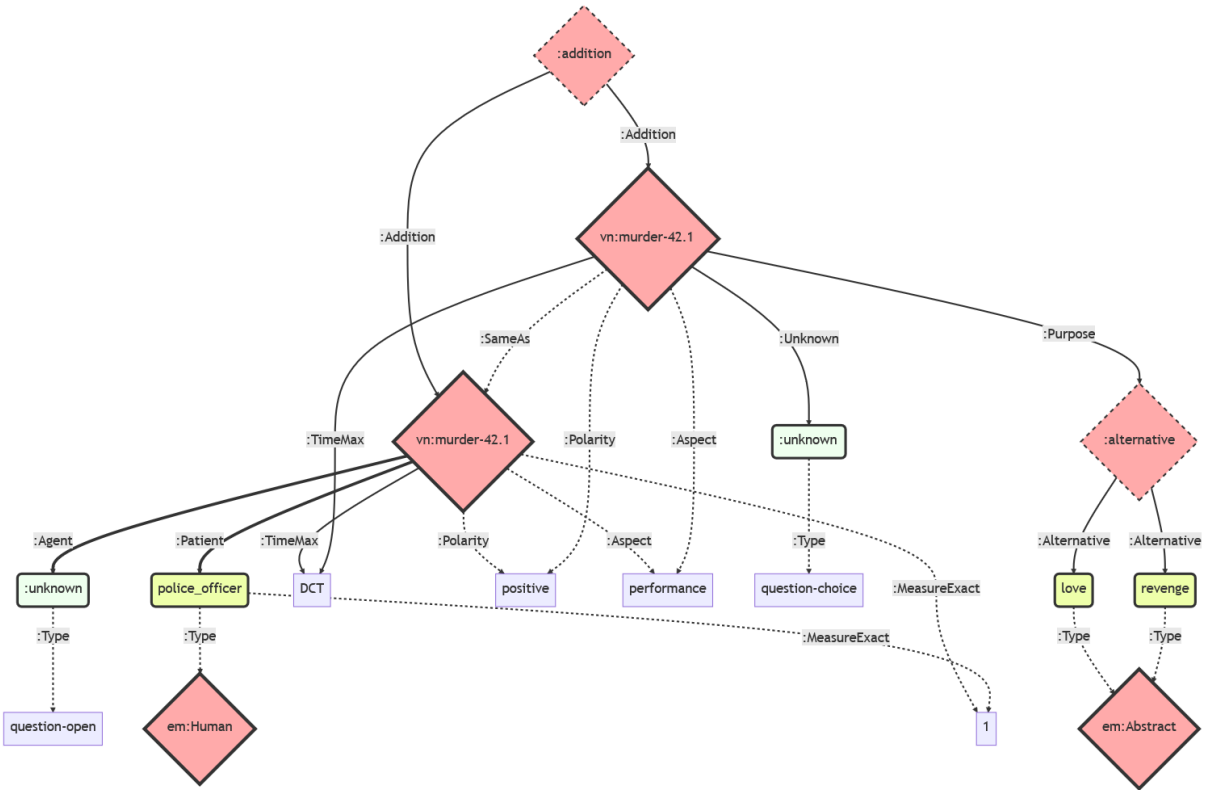


Figure 3: MR4AP representation example for subsection 4.3

either positive, negative, or doubtful), alternative questions (`question-choice`, whose answers are mentioned in the question with an alternative being offered), and variable questions (`question-open`, whose answers are quite open-ended, although governed by the nature of the unknown element). In examples (5-6), two elements are unknown: the perpetrator (*i.e.*, the `:Agent` of `vn:murder-42.1`) and the motive of the latter (*i.e.*, the possible choices linked to the `:alternative` node[5]).

For the first one, it is an `:Agent` relation to an `:unknown` node that is created from `vn:murder-42.1`. This same node is thus typed `question-open`. For the second one, the `:alternative` node, linked to its predicate with a `:Purpose` relation, offers a choice between two known options. To mark the interrogativeness attached to the `vn:murder-42.1` node, we use an `:Unknown` relation to point towards an `:unknown` node, itself pointing towards a `question-choice` value. Moreover, had the coordination of the questions not been made explicit by the conjunction *and* but had remained implicit with a paratactic conjunction, the representations would have remained identical. Our example does not display any polar questions, but had the question been *Was the police officer murdered?*, the `vn:murder-42.1` node would have had an `:Unknown` relation pointing towards an `:unknown` node whose `:Type` value would have been `question-closed`.

Multiword expressions. MWEs are known to be a *pain in the neck* (Sag et al., 2002), both because of their heterogeneity and their pervasiveness. We choose to consider them non-compositionally as most MWEs' meaning can not be broken down according to their constituents (Constant et al., 2017). Considering an MWE as a single semantic entity enables a greater graph similarity from one language to another (Navigli et al., 2022), or even when comparing a set of paraphrases. For instance, in (5), *police officer* (which could have been translated into *officier de police*, *agente di polizia*, or even *ḍābit aš-šurṭa* in French, Italian, and Arabic respectively) becomes in (6) the single word token *policeman* (which could have been translated into *policier*, *poliziotto*, or even *šurṭiyy*).

[5]Such nodes also appear in non-interrogative statements to represent disjunctions *(I reckon John wrote novels **or** poems)*.

Having a single node for both expressions of the same real world's concept seems mandatory as far as uniformity is concerned. MWEs can be the source of a relation (the prepositional locution *out of* for `:Purpose`), of a non-event entity (the compound *police officer*), or of an event (nominal or verbal, like the light verb construction *to commit murder*). In the latter case, the light verb is simply dropped because it is redundant, but the accompanying event nominal inherits the arguments and the various linguistic features attached to it. For example, *committed* in (5) is a preterite verb, hence the `:TimeMax` relation towards the DCT; it denotes a completed action that led to a result, hence the `performance` aspectual value; and there is no negation particle, hence the `positive` polar value. The *police officer* is the `:Patient` of the murder rather than its commission's, and the unknown subject is its `:Agent`. The `vn:murder-42.1` node actually inherits the arguments and linguistic features of what could have been a `vn:complete-55.2` node with the former as its `:Theme`.

5 Annotated corpora

We first applied MR4AP to French short sentences from the TaPaCo corpus (Scherrer, 2020) to ensure its usability on low-complexity sentences. We then applied it to more complex sentences from Wikipedia articles in five languages to also test its multilingual compatibility. Both datasets are available on the GitHub repository[6].

MR4AP-tapaco. In order to demonstrate the viability of our formalism, we produced an annotated dataset. Version 0.1 of the MR4AP-tapaco corpus[7] is relatively small, but it is bound to grow as contributions are made. So far, 100 short sentences in French from the TaPaCo paraphrase corpus have been automatically annotated using our own tool before being manually checked and validated. Choosing paraphrases is not insignificant as it will allow us to gauge the similarity of the graphs obtained after annotating sets of paraphrases once we have enough data. Some statistics regarding this corpus can be found on the remote repository.

Multilingual compatibility experiment. In order to validate MR4AP's multilingual compatibility in practice, as well as to explore ways to propose a protocol and a manual annotation tool, we started a first small-scale annotation project. First, we wrote guidelines[8] that present MR4AP in an exhaustive and extensive way. In a second step, to avoid content bias, we randomly selected five Wikipedia articles in French and kept the first three sentences of each. After setting up all the necessary parameters in the INCEpTION platform (Klie et al., 2018), our annotation tool of choice that allows the annotation of both explicit and implicit elements, we annotated the five texts. In a third step, we automatically translated them into English, Spanish, Italian and Modern Standard Arabic (MSA), and annotated them[9]. Despite the small scale of the annotation effort and the relatively modest language panel considered, we were able to determine that MR4AP seems to be multilingually compatible. Pursuing the annotation effort is however mandatory.

MR4AP-wikipedia. Having established that the formalism can be used with different languages, the five annotated texts in French, English, Spanish, Italian and MSA constitute the first annotated texts for the MR4AP-wikipedia corpus[10]. The objective is to obtain a fully manually annotated dataset that would serve as a gold standard. This dataset will be regularly enriched with new annotated texts.

Data format. We use JSON files with three fields: `id` (the document identifier), `text` (the document's textual content), and `rdf` (the RDF[11] representation of the text). An RDF data model consists of RDF triples where each RDF triple codifies a statement in the form of subject–predicate–object expressions. RDF triples have no ordering and triples can be linked to other triples according to their common elements (so that a graph is obtained). Using RDF, we make the assumption that regardless of the order in which the sentences are written, the text will systematically produce the same semantic graph (*i.e.*, the same set of triples). Finally, RDF triples applied with OWL[12] can be used as input for a reasoner that in turn could be used to saturate the graph with inferred annotations. Our dataset's RDF graphs can be viewed with an appli-

[6] https://github.com/Emvista/MR4AP/tree/main/corpora

[7] https://github.com/Emvista/MR4AP/tree/main/corpora/MR4AP-tapaco

[8] https://github.com/Emvista/MR4AP/tree/main/guidelines/guidelines.md

[9] The annotation was carried out by the two authors.

[10] https://github.com/Emvista/MR4AP/tree/main/corpora/MR4AP-wikipedia

[11] Resource Description Framework: https://www.w3.org/RDF/

[12] Web Ontology Language: https://www.w3.org/OWL/

cation such as Protégé[13] (Musen, 2015).

6 Limitations and perspectives

Although our formalism is able to address some shortcomings that other formalisms can't, some limitations remain. On the one hand, given all the elements that MR4AP represents, annotation remains a time-consuming and rather complex process. Moreover, this complexity only increases with the length of the texts. The multilingual annotation experiment described above only exacerbated the need for a perhaps more efficient annotation strategy.

On the other hand, while the annotation was carried out on texts in five different languages, the variety was somewhat limited: three of them are Romance languages with few differences; English is a standard; only MSA, due to its important differences with the other four languages, really allows us to conclude that MR4AP seems compatible with multilingualism. Continuing the annotation effort with languages from different families or low-resource languages would enable us to support this assertion.

Moreover, from a cross-formalism perspective and following the recent mapping effort made between AMR and UMR (Bonn et al., 2023), we would like to follow suit and align MR4AP to these two formalisms. This mapping would allow the production of a multi-formalism corpus, which could in turn allow the implementation of comparative experiments on the performance of each formalism from the same source material.

Important questions remain to be tackled: Which tool to use/develop to annotate more efficiently with a formalism such as MR4AP? And how to ensure annotation completeness for a given text? Does the level of anchoring have an impact on the explainability of semantic parsers (*e.g.*, to source graph nodes)? Knowing that graph linearization is an important topic and that edge ordering can have a "big negative effect" on the evaluation measures of some tasks (Bevilacqua et al., 2021), is RDF appropriate and what impact would this have on the performance of state-of-the-art parsers?

7 Conclusion

We have highlighted the divergences and convergences between ten meaning representation formalisms. On this basis, we have put forth and positioned MR4AP, our application-oriented formalism. We have extensively described it both through guidelines and through several examples demonstrating its efficiency in representing meaning at the document level by taking into account discourse, coreference and temporal relations, its potential to represent some of the most complex linguistic phenomena, and its robustness to paraphrasing and multilingualism. We have also briefly presented the first version of the MR4AP-tapaco corpus as well as a first small-scale manual annotation effort to assert the multilingual compatibility of the formalism in practice. We concluded that MR4AP is usable regardless of the text's language, and this annotation effort allowed us to create the MR4AP-wikipedia corpus, which will serve as a gold standard. Note that a hybrid semantic parser, which does not need any training data to annotate textual content, has been developed with this formalism, is already in production, and will be the subject of a future publication.

[13] Protégé: https://protege.stanford.edu/

Acknowledgements

We would like to warmly thank the three anonymous reviewers who had us thinking and helped in improving the content of the article thanks to their accurate and thorough feedback.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–89, Vancouver, Canada. Association for Computational Linguistics.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze and Johan Bos. 2019. Thirty musts for meaning banking. In *Proceedings of the First In-*

ternational Workshop on Designing Meaning Representations, pages 15–27, Florence, Italy. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3196–3200, Istanbul, Turkey. European Language Resources Association (ELRA).

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12564–12573.

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, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. 2023. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C. Association for Computational Linguistics.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3:281–332.

Ruixiang Cui and Daniel Hershcovich. 2020. Refining implicit argument annotation for UCCA. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 41–52, Barcelona Spain (online). Association for Computational Linguistics.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637, Hong Kong, China. Association for Computational Linguistics.

Robert MW Dixon. 2010. *Basic linguistic theory volume 2: Grammatical topics*, volume 2. Oxford University Press on Demand.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.

Jeffrey Flanigan, Ishan Jindal, Yunyao Li, Tim O'Gorman, Martha Palmer, and Nianwen Xue. 2022. Meaning representations for natural languages: Design, models and applications. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 1–8, Abu Dubai, UAE. Association for Computational Linguistics.

Venkata Govindarajan, Benjamin Van Durme, and Aaron Steven White. 2019. Decomposing generalization: Models of generic, habitual, and episodic statements. *Transactions of the Association for Computational Linguistics*, 7:501–517.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3153–3160, Istanbul, Turkey. European Language Resources Association (ELRA).

Daniel Hershcovich, Nathan Schneider, Dotan Dvir, Jakob Prange, Miryam de Lhoneux, and Omri Abend. 2020. Comparison by conversion: Reverse-engineering UCCA from syntax and lexical semantics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2947–2966, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Glyn Hicks. 2009. Tough-constructions and their derivation. *Linguistic Inquiry*, 40(4):535–566.

Hans Kamp, Uwe Reyle, Hans Kamp, and Uwe Reyle. 1993. Tense and aspect. *From Discourse to Logic:*

Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory, pages 483–689.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Alexander Koller, Stephan Oepen, and Weiwei Sun. 2019. Graph-based meaning representations: Design and processing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 6–11, Florence, Italy. Association for Computational Linguistics.

Marco Kuhlmann and Stephan Oepen. 2016. Squibs: Towards a catalogue of linguistic graph Banks. *Computational Linguistics*, 42(4):819–827.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.

Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, et al. 2006. Annotation on the tectogrammatical level in the prague dependency treebank. *annotation manual. Technical Report*, 30:5–11.

Mark A Musen. 2015. The protégé project: a look back and a look forward. *AI matters*, 1(4):4–12.

Roberto Navigli, Rexhina Blloshmi, and Abelardo Carlos Martinez Lorenzo. 2022. Babelnet meaning representation: A fully semantic formalism to overcome language barriers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12274–12279.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics.

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.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2022. How much of UCCA can be predicted from AMR? In *Proceedings of the 18th Joint ACL - ISO Workshop on Interoperable Semantic Annotation within LREC2022*, pages 110–117, Marseille, France. European Language Resources Association.

Jakob Prange, Nathan Schneider, and Omri Abend. 2019a. Made for each other: Broad-coverage semantic structures meet preposition supersenses. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 174–185, Hong Kong, China. Association for Computational Linguistics.

Jakob Prange, Nathan Schneider, and Omri Abend. 2019b. Semantically constrained multilayer annotation: The case of coreference. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 164–176, Florence, Italy. Association for Computational Linguistics.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. 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 731–744, New Orleans, Louisiana. Association for Computational Linguistics.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings 3*, pages 1–15. Springer.

Yves Scherrer. 2020. TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6868–6873, Marseille, France. European Language Resources Association.

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. Comprehensive supersense disambiguation of English prepositions and possessives. In *Proceedings*

of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 185–196, Melbourne, Australia. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Adi Shalev, Jena D. Hwang, Nathan Schneider, Vivek Srikumar, Omri Abend, and Ari Rappoport. 2019. Preparing SNACS for subjects and objects. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 141–147, Florence, Italy. Association for Computational Linguistics.

Hiroshi Uchida, Meiying Zhu, and Tarcisio Della Senta. 1999. A gift for a millennium. *IAS/UNU, Tokyo*.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3-4):343–360.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.

Meagan Vigus, Jens E. L. Van Gysel, and William Croft. 2019. A dependency structure annotation for modality. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 182–198, Florence, Italy. Association for Computational Linguistics.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Aaron Steven White, Elias Stengel-Eskin, Siddharth Vashishtha, Venkata Subrahmanyan Govindarajan, Dee Ann Reisinger, Tim Vieira, Keisuke Sakaguchi, Sheng Zhang, Francis Ferraro, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2020. The universal decompositional semantics dataset and decomp toolkit. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5698–5707, Marseille, France. European Language Resources Association.

Zdeněk Žabokrtský, Daniel Zeman, and Magda Ševčíková. 2020. Sentence meaning representations across languages: What can we learn from existing frameworks? *Computational Linguistics*, 46(3):605–665.

Claim Extraction via Subgraph Matching over Modal and Syntactic Dependencies

Benjamin Rozonoyer[1], Michael Selvaggio[2], David Zajic[2] and Ilana Heintz[3]

[1]University of Massachusetts Amherst
brozonoyer@cs.umass.edu
[2]Raytheon BBN Technologies
{michael.selvaggio,david.m.zajic}@rtx.com
[3]Synoptic Engineering
ilana@synopticengineering.com

Abstract

We propose the use of modal dependency parses (MDPs) aligned with syntactic dependency parse trees as an avenue for the novel task of claim extraction. MDPs provide a document-level structure that links linguistic expression of events to the conceivers responsible for those expressions. By defining the event-conceiver links as claims and using subgraph pattern matching to exploit the complementarity of these modal links and syntactic claim patterns, we outline a method for aggregating and classifying claims, with the potential for supplying a novel perspective on large natural language data sets.

Abstracting away from the task of claim extraction, we prototype an interpretable information extraction (IE) paradigm over sentence- and document-level parse structures, framing inference as *subgraph matching* and learning as *subgraph mining*. We make our code open-sourced at https://github.com/BBN-E/nlp-graph-pattern-matching-and-mining.

1 Introduction

A promise of natural language processing (NLP) tools is to bring fast understanding of large corpora of unstructured data. This has been achieved through tasks such as summarization (Prudhvi et al., 2020), knowledge base population (Glass and Gliozzo, 2018), and question-answering systems (Arbaaeen and Shah, 2020), among others. These outcomes provide different views into the chosen data source, highlighting aspects such as event timelines, known and novel relationships between entities, causality, and others. A less explored view of unstructured data may take the form of a Claim Bank, in which NLP tools provide access to a set of differentiated claims expressed by explicit claimants within a document corpus.

We will define a claim as an assertion that is explicitly linked to a source. The source may be a person or an organization. The source may be explicit or implicit; a common example of the latter case is when the author of a document is the source of a claim and is defined as part of the metadata, but is not explicit in the document content. Our goal is to automatically identify claims in, and learn claim structures from, natural language text. Such claims could then be the impetus for claim verification, clustering, provenance graph generation, etc., as described below.

2 Related Work

2.1 Claim Extraction

Recent work in claim verification is closely related to the effort at hand. In particular, Zhang et al. (2020) build a provenance graph for claims, linking each claim to its likely sources. The "query" claims are derived from the opinion corpus developed in Choi et al. (2005). In Zhang et al. (2020), sentences relevant to a query claim are first retrieved from a variety of documents, and the implicit (author) and explicit (named) sources identified via a Textual Entailment task, followed by a classification task to identify the relationship between source and statement. Thus, a set of related claims is derived from an original, provided claim. The provenance graph built from these efforts is used to identify supporting, contradictory, or neutral relationships between statements relating to a claim.

Similarly, FEVER (Thorne et al., 2018), HoVer (Jiang et al., 2020) and WiCE (Kamoi et al., 2023) are open source datasets of related facts for the NLP community to make shared progress on claim verification. The relationship between facts (the term is used interchangeably with claims in these studies) is of interest, rather than their relationship to sources. Facts or claims enter the corpus through a crowdsourced annotation effort or automatically from Wikipedia as in the case of WiCE.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 122–135
June, 2023.

An earlier important work (Choi et al., 2005) identifies sources responsible for opinions, emotions, and sentiment through a dataset collection effort. The authors test an automated approach for detecting these relationships with conditional random fields, as an information extraction task.

These works do not give a definition of a claim, though the intuitive notion seems to be that a claim is a declarative statement that could be given a truth value. In the present work, we add the constraint that a claim must be associated with a claimer. The DARPA Active Interpretation of Disparate Alternatives (AIDA) program (Onyshkevych, 2017; Hovy, 2020) has helpfully defined the truth-valued statement as the "inner claim," and its epistemic or sentimental association with a claimer as the "outer claim" – together, the inner and outer claim constitute a claim that can be compared to others in terms of support, contradiction, relevance, etc.

The NEWSCLAIMS benchmark (Reddy et al., 2022) is the most recent and closely related parallel work on claim detection since it also stems from AIDA definitions of claims — the released dataset consists of claims annotated over the subset of articles from the LDC corpus LDC2021E11 related to COVID-19 (cf. §6.2). We recommend it as a reference for the task at hand despite slight terminological differences; unlike the methodology explored here, the authors experiment with zero-shot and prompt-based baselines.

The goal of this work is to introduce a novel way of automatically identifying claims according to this definition; in particular, we identify that inner and outer claims are often, but not always, identifiable through a sentence-level predicate-argument structure. In cases where the outer and inner claim are expressed over multiple sentences, a document-level structure must be accessed to reveal the relationship. We describe an algorithm for combining structural and semantic information from the sentence and document levels to automatically identify claims. This effort might be seen as a precursor to the studies described above; we aim to automate the initial task of finding the claims for analysis.

2.2 Subgraph Matching and Mining

DotMotif (Matelsky et al., 2021), a declarative library for identifying and extracting frequent motifs from large graphs of connectomes, begs the exploration of an analogous approach in NLP, where text can be viewed as the tip of the iceberg in a rich network of underlying syntactic, semantic and pragmatic parses or "deep structures". We describe our algorithmic approach to subgraph isomorphism in §4.7; "soft", neural implementations such as NeuroMatch of (Ying et al., 2020a) use graph neural network (GNN) encoders to achieve 100x speedup over traditional combinatorial approaches. While the latter GNN architectures were designed with molecular graphs in mind, Marcheggiani and Titov (2017); Bastings et al. (2017); Nguyen and Grishman (2018); Rozonoyer (2021) successfully tailored the GCN (Kipf and Welling, 2016) and GAT (Veličković et al., 2017) architectures to encode dependency syntax for semantic role labeling, neural machine translation, event detetion, and AMR parsing, respectively.

3 Linguistic Motivation

Our approach to claim extraction leverages Modal Dependency Parsing (MDP), a document-level annotation scheme for modality introduced by Vigus et al. (2019) and adopted by Yao et al. (2021) to crowdsource a dataset and train a neural parser. We use MDP in conjunction with sentence-level syntactic dependency parses. Document-level MDP restricts the extraction space to a pool of potential claims that include inter-sentence inner/outer claim relationships. Sentence-level dependency parses provide the grammatical structure that allows us to further constrain the pool of potential claims to those that match our analysis of the clausal structure of claims.

3.1 Modal Dependency Parsing

MDPs provide a document-level structure that links events to their conceivers, including the author conceiver, which is at the root of the document. Furthermore, the MDP edges provide epistemic values of the relationship between the conceiver and the event: whether the conceiver is certain that the event occurred, uncertain, or believes the event did not occur. This provides essential information to understanding how claimers and claims interact.

In Figure 1, we show the MDP for sentence (1) below, cited and manually annotated in Vigus et al. (2019):

(1) [About 200 people were believed **killed** and 1,500 others were **missing** in the Central Philippines on Friday when a **landslide buried** an entire village], **the Red Cross** said.

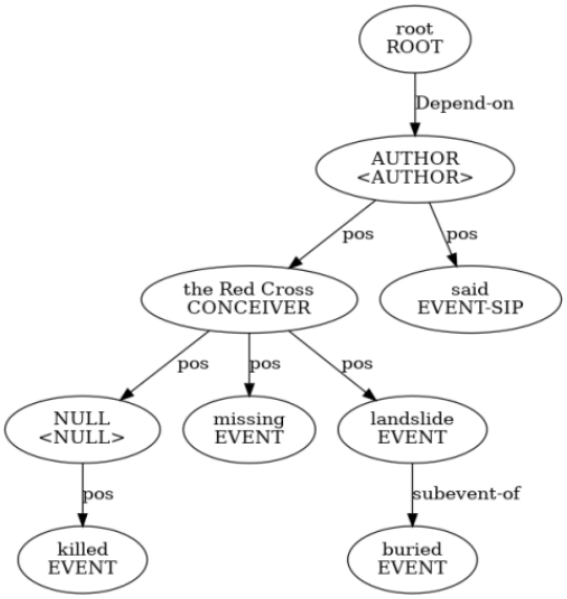


Figure 1: MDP for sentence (1)

In this sentence-sized document, the Red Cross is a conceiver making a claim about the "missing", "landslide", and "buried" events with positive (certain) epistemic value. The passive construction *believed killed* introduces a null conceiver ("believed by *whom*?") in the mental space of the governing "the Red Cross". The source introducing predicate "said" is a clue to attributing the other events in this sentence to "the Red Cross", while the saying event itself can be attributed to the author of the document with positive epistemic value.

3.2 Claim Structure in Syntax

We observe that there are typical syntactic structures associated with our intuition of claim. Consider the sentence:

(2) [**US officials** $_{\text{nsubj}}$] [said $_{\text{SIP}}$] [numerous social media sites **launched** an effort to spread misinformation $_{\text{ccomp}}$].

"Said" serves as the source introducing predicate (SIP) that syntactically governs the claimant US officials (its noun subject) and its clausal complement, the semantic "inner claim."

We can convey the exact same claim information in an altogether different lexical and syntactic structure:

(3) Numerous social media sites **launched** an effort to spread misinformation, [[according to $_{\text{case}}$] **US officials** $_{\text{obl}}$].

The conceiver *US officials*, which used to be the subject of the main clause, is now an oblique argument of the inner claim's predicate *launched*. Our claim-finding approach aims to account for this variation in expression, among other syntactic possibilities, and the presence of inter-sentence claim/claimant relationships.

3.3 Complementarity of Modal and Syntactic Structure

The modal and syntactic structures inform each other to the extent that the former is responsible for providing the evidential semantics of who-claims-what-with-what-certainty, while the latter comprises the clausal or grammatical form to express these semantic relations.

To see that the relationship between these structures is not trivial, consider the sentence:

(4) [[According to $_{\text{case}}$] the Pythagorean theorem $_{\text{obl}}$], the square of the hypotenuse equals the sum of the squares of the sides.

Although this sentence has the same syntactic skeleton as the previous sentence, the Pythagorean theorem is nevertheless not a viable claimant semantically. It is the job of the modal dependency parser to predict that *US officials* is a conceiver but the *Pythagorean theorem* is not; the basic syntactic structure does not provide us with the lexicosemantic information to tell apart potential claimants from non-claimants.

However, Modal Dependency Parsing is a budding technology that is not entirely robust, and due to the broad event annotation scheme in the crowdsourced dataset (Yao et al., 2021) the parsers in practice tend to be high-recall and low precision, such that automatically-generated MDPs are expected to contain false positive edges. This provides a motivation to use more reliable dependency parse (DP) clausal structures for pruning the original claim space consisting of every possible *Conceiver-Event* edge. In one preliminary study, a seedling set of DP patterns allowed us to focus our attention on 34% of the proposed MDP edges.

4 Claim Extraction via Subgraph Pattern Matching

In order to algorithmically exploit this synergy between evidential/modal and clausal/syntactic structure, we explore the task of claim extraction within the framework of subgraph isomorphism.

4.1 Problem Definition as Subgraph Matching

Our approach entails matching query graphs corresponding to basic *claim structures* against a composition of the document-level modal parse and sentence-level syntactic dependency parses.

The document-level MDP is a directed acyclic graph, and the sentence-level DP is a directed tree between token nodes. To compose the MDP and per-sentence DPs into a single graph, we construct directed edges from the MDP nodes to their corresponding token nodes, resulting in a composition that is also a directed acyclic graph for the entire document.

4.2 Node and Edge Types

Node Types. We define two node types in the composed modal-syntactic graph:

- ***modal nodes*** to represent abstract *Conceiver* and *Event* nodes
- ***token nodes*** to represent words

Edge Types. We define three edge types:

- ***modal edges*** connect conceivers with conceivers or conceivers with events, and are labeled with modal relations e.g. *pos*, *neg*, *pp*[1]
- ***syntax edges*** connect tokens, and are labeled with syntactic dependency relations e.g. *nsubj*, *ccomp*, *advcl*
- ***modal-token edges*** connect modal nodes with token nodes, e.g. the *Conceiver* node corresponding to the Washington Post entity in the MDP with every token in the multiword expression "The Washington Post".

4.3 Graph Structure Formalization

We formalize the definition of graph structures in our domain as $G = (V, E, \phi, \psi)$ where V is the set of nodes, E is the set of edges, and ϕ and ψ are the node and edge type assignment functions, respectively, for the edges described in §4.2:

$$\phi : V \rightarrow \{\textit{modal, token}\}$$
$$\psi : E \rightarrow \{\textit{modal, syntax, modal-token}\}$$

Additional categorical node and edge feature functions are contingent on the node's ϕ or edge's ψ type, respectively, as shown in Table 1.

[1]partially positive

Node/Edge Type	Feature Function(s)
$\phi(n) = \textit{modal}$	$\mu(n) \in \{\textit{Conceiver}, \textit{Event}\}$
$\phi(n) = \textit{token}$	$\tau(n) =$ text of token $\upsilon(n) =$ UPOS of token $\chi(n) =$ XPOS of token
$\psi(e) = \textit{modal}$	$\mu(e) \in \{\textit{pos}, \textit{neg}, \textit{neut}\}$
$\psi(e) = \textit{syntax}$	$\sigma(e) =$ syntactic relation
$\psi(e) = \textit{modal-token}$	no further typing

Table 1: Node and edge feature functions for composed modal-syntactic graphs

4.4 Document Digraph

We produce a document-level graph in the NetworkX[2] API by 1) storing the document-level modal dependency parse as a NetworkX DiGraph (directed graph), 2) storing each sentence's syntactic dependency parse as a NetworkX DiGraph, and 3) composing the graphs into a single NetworkX DiGraph. We connect the document-level modal nodes with the sentence-level token nodes via modal-token token edges described in §4.2.

4.5 Pattern Digraphs

We create pattern NetworkX DiGraphs as small graph structures that combine the core elements of the syntactic and modal structures constituting a claim. We use a subgraph isomorphism algorithm (§4.7) to match these claim pattern digraphs against the document digraph in order to discover claims. Each pattern digraph is accompanied by a node-match and edge-match function (§4.6) that allows a pattern node/edge to be underspecified with respect to irrelevant features but still match a fully annotated node/edge in the document digraph. In effect, we can view the pattern digraph as a "query" graph that is as generic a structure as possible for the intuition of the claim structure we are trying to match. The pattern digraphs expressing the two claim structures in sentences (2) and (3) are shown in Figures 2 and 3.

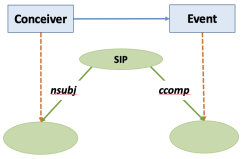


Figure 2: *ccomp* pattern graph

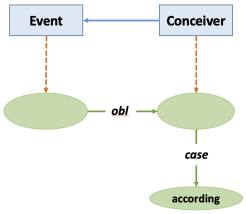


Figure 3: *according_to* pattern graph

[2]https://networkx.org/

4.6 Node-Match and Edge-Match Functions

In addition to simply defining modal-syntactic pattern structures to match claims in the document, as mentioned in §4.5, we define **node-match** and **edge-match** functions (with a Boolean return value) that allow us to specify custom criteria when checking for node or edge equivalence between a pattern structure and document structure. We show a base set of functions in Table 2. For example, we may require an exact match for the syntactic relation on a syntax edge, while allowing any value for the modal relation on a modal edge because it merely specifies the epistemic stance of the claimant toward the inner claim, and does not determine the inherent claim structure.

4.7 Algorithm

We employ the NetworkX GraphMatcher[3] API over the document digraph and a pattern digraph in order to return the nodes in the document graph isomorphic to the nodes in the pattern graph. The matcher uses the VF2 (sub)graph isomorphism algorithm (Cordella et al., 2004). While subgraph isomorphism is an NP-complete problem, the time complexity of the VF2 algorithm is $\Theta(N^2)$ in the best case and $\Theta(N!N)$ in the worst case, and maintains a $\Theta(N)$ space complexity. Given that N is the union of all tokens in a given document with abstract modal dependency nodes (which never exceed the number of tokens in sufficiently complex sentences), neither time nor space complexity poses an obstacle to the algorithm's practical application in prototyping this approach.

4.8 Relaxed Patterns with On-Match Filtering for Generalized Structures

Some claim structures may be accounted for with less deterministic pattern definitions. A salient instance is found in sentence (1) above and many other annotated examples in our analysis: multiple events are assigned the same modal pattern, and are grammatically subordinated to the clausal complement of the same SIP. However, each event trigger has a different location at a potentially different depth in the SIP subtree.

We generalize the 1-hop relation in the *ccomp* pattern graph into a k-hop relation in the *relaxed_ccomp* pattern, shown in Figure 4, by defining two distinct nodes, α and β, to match the immediate (1-hop) clausal complement of the SIP and any other (k-hop) event token node subordinated to it, respectively. The definition imposes no syntactic requirement on β (only that it be the token of a modal event node). To ensure that the match for β is actually subordinate to the SIP's clausal complement, we implement an on-match filter[4] that for each returned isomorphism checks $is\_ancestor(\alpha, \beta)$ and discards matches where β falls outside the subtree governed by α.

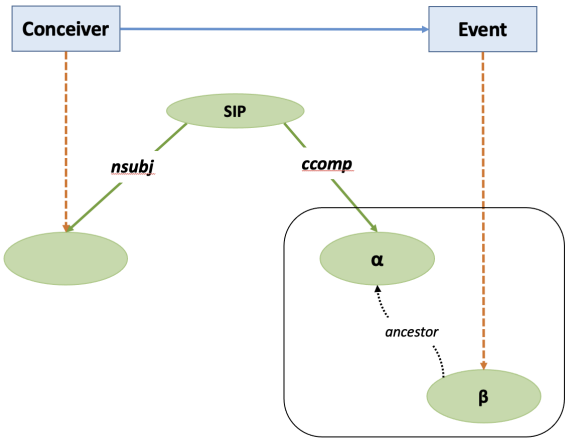


Figure 4: *relaxed_ccomp* pattern graph that requires ancestor-checking filter

On-match filtering allows us to generalize claim patterns insofar as we can exploit the topology of the graph to prune away uncompliant extractions.

5 Building a Claim Bank

We apply our claim extraction algorithm to the crowdsourced English modal dependency dataset (Yao et al., 2021), over which we additionally ran the default Stanza dependency parser[5] (Qi et al., 2020) (English *ewt* model) to have both the MDP and DP information available for every document.

As a minimal qualitative assessment for the prototyped approach, we build a Claim Bank by running the subgraph pattern matcher with the *ccomp* and *according_to* claim patterns over the train, dev and test portions of the crowdsourced dataset with 289, 32 and 32 parsed documents, respectively. We show in Table 3 raw MDP conceiver-event edge counts to illustrate the large cardinality of the claim

[3] https://networkx.org/documentation/stable/reference/algorithms/isomorphism.vf2.html#graph-matcher

[4] "on-match" terminology borrowed from spaCy's rule-based matching API (https://spacy.io/usage/rule-based-matching)

[5] Version 1.2

node type match	$\phi(n_1) = \phi(n_2)$
node match on feature function γ	$node\_type\_match(n_1, n_2) \wedge$ $(\gamma(n_1) \wedge \neg\gamma(n_2)) \vee (\neg\gamma(n_1) \wedge \gamma(n_2)) \vee (\gamma(n_1) = \gamma(n_2))$
edge type match	$\psi(e_1) = \psi(e_2)$
edge match on feature function δ	$edge\_type\_match(e_1, e_2) \wedge$ $(\delta(e_1) \wedge \neg\delta(e_2)) \vee (\neg\delta(e_1) \wedge \delta(e_2)) \vee (\delta(e_1) = \delta(e_2))$

Table 2: Customizable Boolean logic for determining equivalence of nodes and edges during subgraph matching of claim pattern graphs to document graph. γ and δ stand for generic node and edge feature functions, respectively

	train	**dev**	**test**
#C-E total	14460	1732	1641
#C-E, C linked to token(s)	8001	1022	1044
#C-E cross-sentence[6]	1736	240	163
#ccomp	558	76	86
#ccomp$_{1\text{-hop}}$	544	71	85
*#*ccomp$_{>1\text{-hop}}$*	1887	233	242
#ccomp$_{>1\text{-hop}}$	926	111	130
#according_to	61	5	6

Table 3: Conceiver-event (C-E) edges and claim extraction counts, grouped by pattern

pool provided by MDP that gets pared down by constraining MDP with DPs. We categorize the *ccomp* family of extractions in terms of how far away the event token is from the clausal complement of the source introducing predicate (*ccomp*$_{\geq 1\text{-hop}}$ correspond to the *relaxed_ccomp* pattern, and an asterisk indicates the number of extractions before applying the on-match *is_ancestor* filter).

We confirm that there is no intersection between the node isomorphism matches returned by each of the claim patterns of interest. We randomly sampled 10 examples from the train set for the extracted *according_to* claims and the *ccomp*-family claims. For all 20 examples, the extractions match our intuition of a claim as an assertion ("inner claim") evidentially related to an opinion-holding conceiver. Sentences (5) and (6) are examples of *according_to* and *ccomp* extractions, respectively, from our random samples:

(5) As of Wednesday, the state **had** more than 16,460 known cases and 539 known deaths, according to **the department**.

(6) The DPA is being **used** to obtain about 60,000 test kits, **Gaynor** told CNN's New Day.

A high-quality Claim Bank, containing overt claimants linkable to real-world entities, facilitates an exploration of claims by individuals of interest, and provides an avenue for sifting through conflicting perspectives on events.

6 Subgraph Matching as Inference, Subgraph Mining as Learning

Our approach outlines an entirely algorithmic inference procedure to extract knowledge elements (KEs), and may therefore be reminiscent of symbolic AI. We have so far discussed how to extract claims with predefined human-curated patterns, but curating such patterns manually is as unreliable and time-consuming as feature engineering, and we want to automatize extracting such patterns from parsed, annotated corpora. We propose subgraph mining as a general-purpose methodology for "learning" patterns, as capable of extracting schematically-defined KEs as the parses are expressive of them. The generality and interpretability of the method we formulate below make it as an appealing alternative to task-specific and at times brittle neural extractors, or to powerful but even less interpretable prompt-based approaches.

An annotated corpus of claims such as LDC2021E11 or NEWSCLAIMS consists of KEs containing the token indices a of the claimer and b of the inner claim, and labels for the claim topic and claimer stance. We can therefore view the claims corpus as $\mathcal{C} = \{c_1, ..., c_n\}$ where each claim c_i has the structure[7]:

$$c_i = \left\{ \begin{array}{l} \textsc{Text} = \text{sentence or document} \\ \textsc{Spans} = \left\{ \begin{array}{l} \{a_1, ..., a_k\} \\ \{b_1, ..., b_l\} \end{array} \right\} \\ \textsc{Labels} = \left\{ \begin{array}{l} \textsc{claim topic} \\ \textsc{claimer stance} \end{array} \right\} \end{array} \right\}$$

Instead of learning span extractors and classifiers from token-level "surface" annotations, we first

[7]This structure can be generalized to various KEs such as entity-relations, event-relations and event-argument frames

apply sentence- and document-level parsers and compose the resulting parses into a digraph as outlined in §4.4. [8]

We hypothesize that in the graph composed of available parse types $\mathcal{P}$, features predictive of the KEs of interest will be contained in the graph neighborhoods of the annotated tokens. We thus define a neighborhood function that returns the k-hop portion of the digraph $\mathcal{D}$ surrounding given tokens t, where directionality d can be specified to only outgoing ($\uparrow$) or only incoming ($\downarrow$) edges from the token node(s), or both ($\updownarrow$). Subgraphs created via those neighborhoods are then passed into a subgraph mining procedure (§6.1) for subgraph-matching-based inference. The learning paradigm is summarized in Algorithm 1:

Algorithm 1 Pattern discovery algorithm

Input: $\mathcal{C}, \mathcal{P}, k \geq 1, d \in \{\uparrow, \downarrow, \updownarrow\}$)
Output: Claim patterns for subgraph matching

$\mathcal{G} \leftarrow \emptyset$
for c_i in $\mathcal{C}$ **do**
  $\text{parses}_i \leftarrow \bigcup_{\text{Parser} \in \mathcal{P}} \text{Parser}(\textsc{Text}(c_i))$
  $\mathcal{D}_i \leftarrow \text{CreateDigraph}(\text{parses}_i)$
  $t \leftarrow \bigcup \textsc{Spans}(c_i)$
  $g \leftarrow \text{neighborhood}(t, \mathcal{D}_i, k, d)$
  add g to $\mathcal{G}$
end for
patterns $\leftarrow \text{Mining}(\mathcal{G})$
return patterns

6.1 SPMiner

SPMiner[9] (Ying et al., 2020b) is the first and only neural approach we are aware of to extract frequent subgraphs from a collection of graphs. SPMiner uses a GNN encoder to embed graphs in an order embedding space, and is trained to enforce a partial ordering such that subgraphs reside to the lower-left of their super-graphs in this space. This neural matching subroutine is used in a search (greedy search, beam search or Monte Carlo tree search; we only explored the first of these) that identifies frequent motifs of size k by iteratively expanding nodes and edges of candidate motifs (starting with seed nodes at random), and selecting those that retain the most points to their top right in the embedding space (i.e. the motifs with highest frequency).

A challenge of applying SPMiner to our NLP domain is that it was developed with molecular graphs in mind that have more variable connectivity patterns than parses, while allowing for at most one label per node or edge (unlike the token nodes in our setting). To mitigate this mismatch, we expand out the graphs so that each attribute is encoded by a synthetic node-edge connection, cf. Figures 5 and 6. The transformation is invertible, such that the expanded graph can be unambiguously collapsed into the original. Given the unreliability of SPMiner's default GNN encoder for our graphs, in the search procedure we swap out the neural (batched) subgraph isomorphism with the much slower VF2 algorithm to ensure that the random walk's results are not confounded by the encoder's performance. This substantially slows down our exploration and is a point for future work (cf. §6.3). Finally, instead of arbitrary seed nodes, we force SPMiner to start the growth with the token nodes for the claimer and inner claim, so as to force the resulting motifs to contain the full claim information.

6.2 Mining for Claims

In another proof-of-concept evaluation, we create a silver corpus of high-precision within-sentence claims from LDC corpus LDC2021E11 (10 documents with 1219 sentence total) that we parsed for dependency syntax and MDP and annotated with the human-curated "seed" patterns discussed in §4. We then use SPMiner to mine for syntax-only patterns from the silver claims. We obtain 100 patterns, which we then apply over the same corpus, examining the quality of the predicted claims. Many of the mined patterns are very simple (high-recall, low-precision), resulting in numerous spurious matches[10]. However, after filtering out any patterns that found over 1000 matches, we are left with syntactic claim patterns that are interpretable and that found reasonable claims not identified by our MDP-constrained seedling patterns. Figure 7 visualizes a non-trivial mined pattern that recovers the overall *ccomp* structure (with some admittedly superfluous edges), Sentence (7) is a claim from

[8] Our implementation supports composing sentence-level syntactic dependency parses (DP) and abstract meaning representation (AMR) (Banarescu et al., 2013), and document-level modal and temporal dependency parses (MDP and TDP) (Zhang and Xue, 2018; Zhang, 2020), into a composite digraph that is "held together" at the token nodes, since every parse type includes them (AMR nodes can be aligned to tokens). We can use any relevant subset of {DP, AMR, MDP, TDP} as our input to the pattern mining procedure.

[9] http://snap.stanford.edu/frequent-subgraph-mining

[10] 449536 total matches, 53630 unique matches

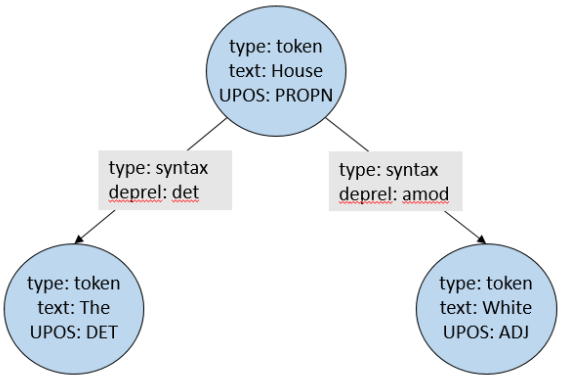


Figure 5: Unexpanded dependency syntax graph for "The White House"

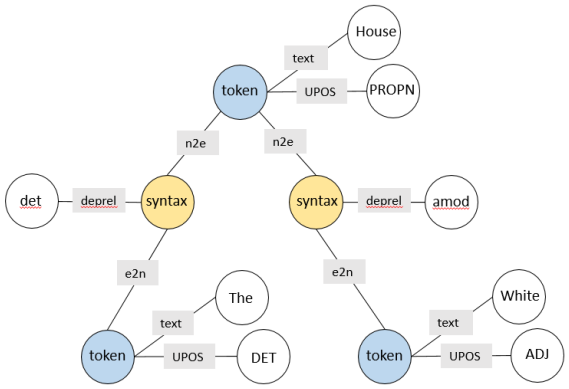


Figure 6: Expansion of dependency syntax graph for "The White House" into graph with a single attribute per each node and edge. *n2e* stands for *node-to-edge* and *e2n* for *edge-to-node*

the silver corpus and Sentence (8) is a novel claim extracted by the mined pattern:

(7) [Tim Trevan, a biological safety expert based in Maryland $_{\text{Claimer}}$], said [most countries had largely abandoned $_{\text{Inner Claim}}$] their bioweapons research after years of work proved fruitless.

(8) [Richard Ebright, a professor of chemical biology at Rutgers University $_{\text{Claimer}}$], said earlier this year in an interview with The Washington Post: ["Based on the virus genome and properties, there is no indication whatsoever that it was an engineered virus." $_{\text{Inner Claim}}$]

Mining the silver corpus after parsing it for AMR yielded 21 AMR-only claim patterns, and after parsing it for both AMR and dependency syntax yielded 29 composite DP-AMR claim patterns.

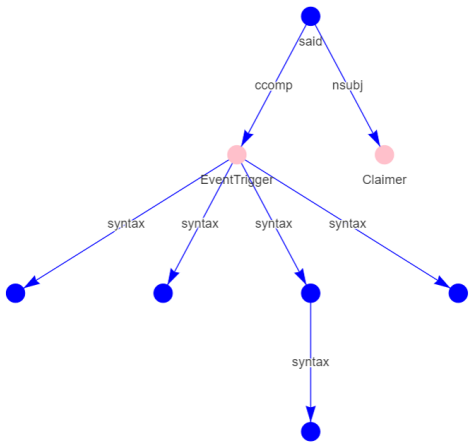


Figure 7: Over 10 documents, this dependency syntax pattern detected 46 claims, 27 of which were not captured by our original DP+MDP patterns

6.3 Challenges and Future Work

The subgraphs returned by SPMiner do not come equipped with node- or edge-match functions as defined in §4.6. This works out in the case of the expanded single-attribute-per-node/edge graph input to SPMiner, as we can simply require that the functions match on all attributes and trust that the mining algorithm will exclude irrelevant/non-predictive attributes from its frequent motifs. We leaving mining for frequent attributes jointly with frequent motifs to future work.

We have yet to explore training NLP-specific GNN encoders such as discussed in §2.2 for accurate neural subgraph isomorphism to speed up the mining procedure and allow for a thorough hyperparameter search that is not prohibitively slow.

Finally, we would like to explore this approach at different linguistic levels, including discourse-level argumentation structures such as that of Stab and Gurevych (2017) or Rhetorical Structure Theory (RST) (Mann and Thompson, 1987).

7 Conclusion

We demonstrate the viability of a simple paradigm for extracting and learning KE structures from a variety of parses. We outline avenues to make this approach more efficient and robust, and surmise that as linguistic representations and parsers continue to improve in scope and in accuracy, the NLP community will benefit from interpretable graph-based techniques over them.

8 Acknowledgements

This research was developed with funding from the Air Force Research Lab (AFRL) and Defense Advanced Research Projects Agency (DARPA), via Contract No.: FA8750-18-C-0001. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This report has been Approved for Public Release, Distribution Unlimited.

References

Ammar Arbaaeen and Asadullah Shah. 2020. Natural language processing based question answering techniques: A survey. In *2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–8. IEEE.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint arXiv:1704.04675*.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 355–362.

Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. 2004. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1367–1372.

Michael Glass and Alfio Gliozzo. 2018. A dataset for web-scale knowledge base population. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 256–271. Springer.

Eduard Hovy. 2020. Active interpretation of disparate alternatives (aida). *Defense Advanced Res. Projects Agency*, 2020.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. Hover: A dataset for many-hop fact extraction and claim verification. *arXiv preprint arXiv:2011.03088*.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. *arXiv preprint arXiv:2303.01432*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

Jordan K. Matelsky, Elizabeth P. Reilly, Erik C. Johnson, Jennifer Stiso, Danielle S. Bassett, Brock A. Wester, and William Gray-Roncal. 2021. DotMotif: an open-source tool for connectome subgraph isomorphism search and graph queries. *Scientific Reports*, 11(1).

Thien Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

B Onyshkevych. 2017. Active interpretation of disparate alternatives.

Kota Prudhvi, A Bharath Chowdary, P Subba Rami Reddy, and P Lakshmi Prasanna. 2020. Text summarization using natural language processing. In *Intelligent System Design: Proceedings of Intelligent System Design: INDIA 2019*, pages 535–547. Springer.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Revanth Gangi Reddy, Sai Chetan Chinthakindi, Zhenhailong Wang, Yi Fung, Kathryn Conger, Ahmed Elsayed, Martha Palmer, Preslav Nakov, Eduard Hovy, Kevin Small, et al. 2022. Newsclaims: A new benchmark for claim detection from news with attribute knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6002–6018.

Benjamin Rozonoyer. 2021. *Graph Convolutional Encoders for Syntax-aware AMR Parsing*. Ph.D. thesis, Brandeis University.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Meagan Vigus, Jens EL Van Gysel, and William Croft. 2019. A dependency structure annotation for modality. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 182–198.

Jiarui Yao, Haoling Qiu, Jin Zhao, Bonan Min, and Nianwen Xue. 2021. Factuality assessment as modal dependency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1540–1550.

Rex Ying, Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, and Jure Leskovec. 2020a. Neural subgraph matching. *ArXiv*, abs/2007.03092.

Rex Ying, A Wang, Jiaxuan You, and Jure Leskovec. 2020b. Frequent subgraph mining by walking in order embedding space. In *Proc. Int. Conf. Mach. Learn. Workshops*.

Yi Zhang, Zachary Ives, and Dan Roth. 2020. "who said it, and why?" provenance for natural language claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4416–4426.

Yuchen Zhang. 2020. *Temporal Dependency Structure Modeling*. Ph.D. thesis, Brandeis University.

Yuchen Zhang and Nianwen Xue. 2018. Neural ranking models for temporal dependency structure parsing. *arXiv preprint arXiv:1809.00370*.

A Subgraph Isomorphism vs Subgraph Monomorphism

We conducted our subgraph matching experiments with the NetworkX subgraph isomorphism matcher, unaware of the subtle difference between subgraph isomorphism and subgraph monomorphism. Our implementation of the *relaxed_ccomp* pattern (cf. §4.8), where we deliberately did not define any edge between α and β to generalize their distance from each other, kept failing to match claims in which α was the parent of β (i.e. exactly 1-hop above it). This led us to implement the pattern $ccomp_{1\text{-}hop}$ in addition to $ccomp_{>1\text{-}hop}$ to get full coverage; the latter pattern having no edge between α and β (as intended), while the former containing the edge corresponding to the 1-hop distance between those two nodes.

We refer the reader to https://networkx.org/documentation/stable/reference/algorithms/isomorphism.vf2.html#subgraph-isomorphism, from which we cite, for a mathematical definition of subgraph isomorphism and monomorphism: "*to say that G1 and G2 are graph-subgraph isomorphic is to say that a subgraph of G1 is isomorphic to G2*".

The key point to note is that in the NetworkX VF2-based subgraph isomorphism, "subgraph" always refers to *node-induced subgraph*:

- If $G' = (N', E')$ is a node-induced subgraph, then:
  - N' is a subset of N, E' is **the subset of edges** in E relating nodes in N'
- If $G' = (N', E')$ is a monomorphism, then:
  - N' is a subset of N, E' is **a subset of the set of edges** in E relating nodes in N'

The node-induced subgraph requirement of isomorphism necessitates the *complete* subset of edges connecting nodes in N'. This explains the failure of the $relaxed\_ccomp$ pattern to match the case where α is a parent of β, since the pattern subgraph does not contain the edge between α and β. By contrast, monomorphism *does* yield a match in this case, as it requires the pattern to define merely *a subset* of the set of edges connecting nodes in N': "*if G' is a node-induced subgraph of G, then it is always a subgraph monomorphism of G, but the opposite is not always true, as a monomorphism can have fewer edges.*"

B SPMiner Hyperparameters

We used the following hyperparameters for SPMiner:

Parameter	Value
node_anchored	true
n_neighborhoods	3000
n_trials	100
min_pattern_size	10
max_pattern_size	50
min_neighborhood_size	10
max_neighborhood_size	60
search_strategy	greedy

Table 4: SPMiner hyperparameters

https://github.com/snap-stanford/neural-subgraph-learning-GNN/blob/master/subgraph_mining/config.py

C Visualizations

Figure 8: Syntactic dependency parse for "*The World Health Organization also warned Monday that the virus had not suddenly become less lethal.*"

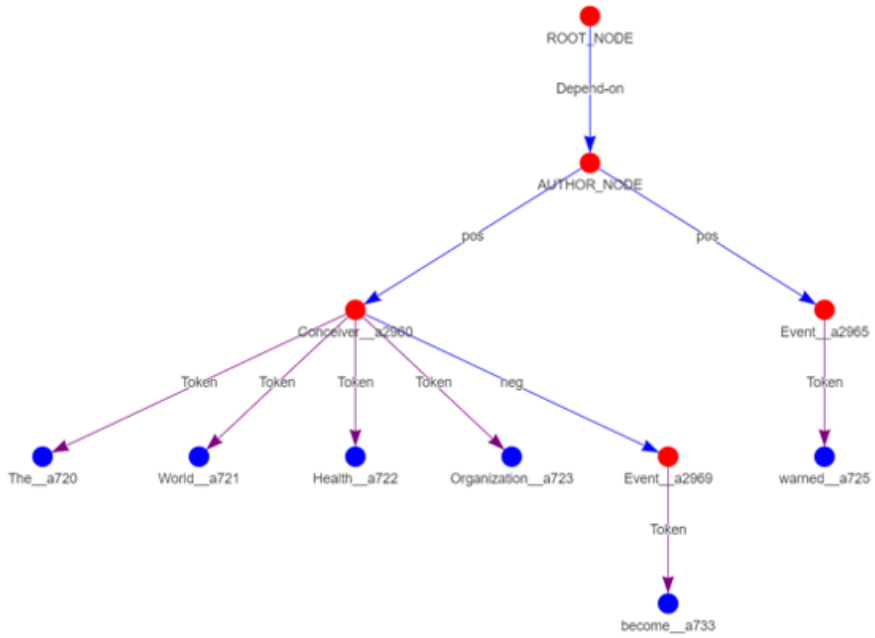


Figure 9: Modal dependency parse for "*The World Health Organization also warned Monday that the virus had not suddenly become less lethal.*"

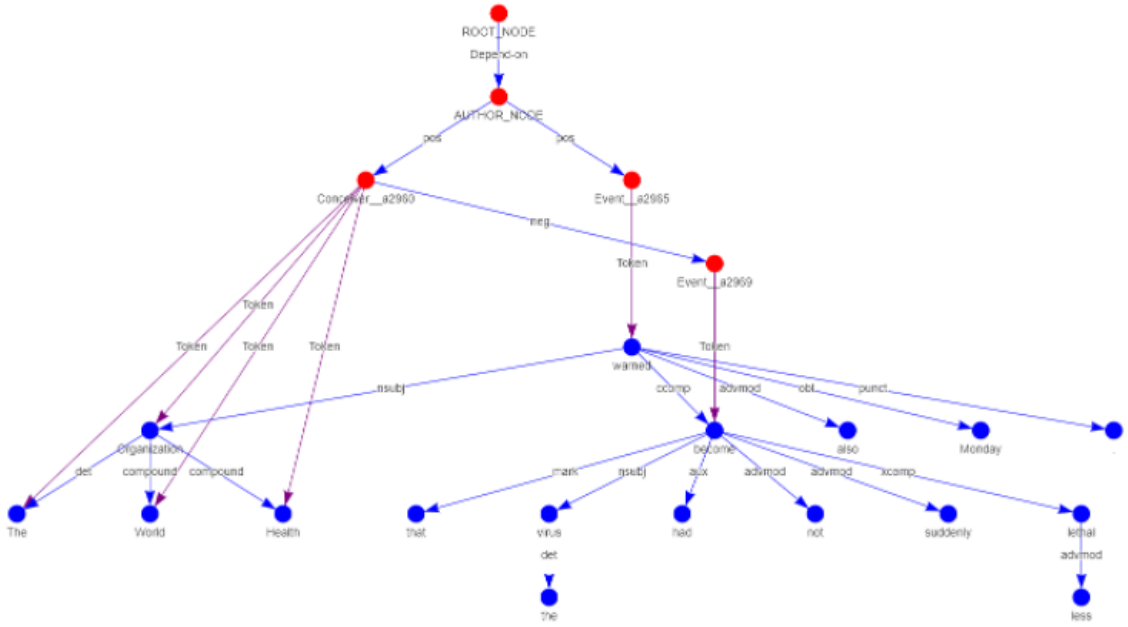


Figure 10: Composed modal and syntactic parse for "*The World Health Organization also warned Monday that the virus had not suddenly become less lethal.*"

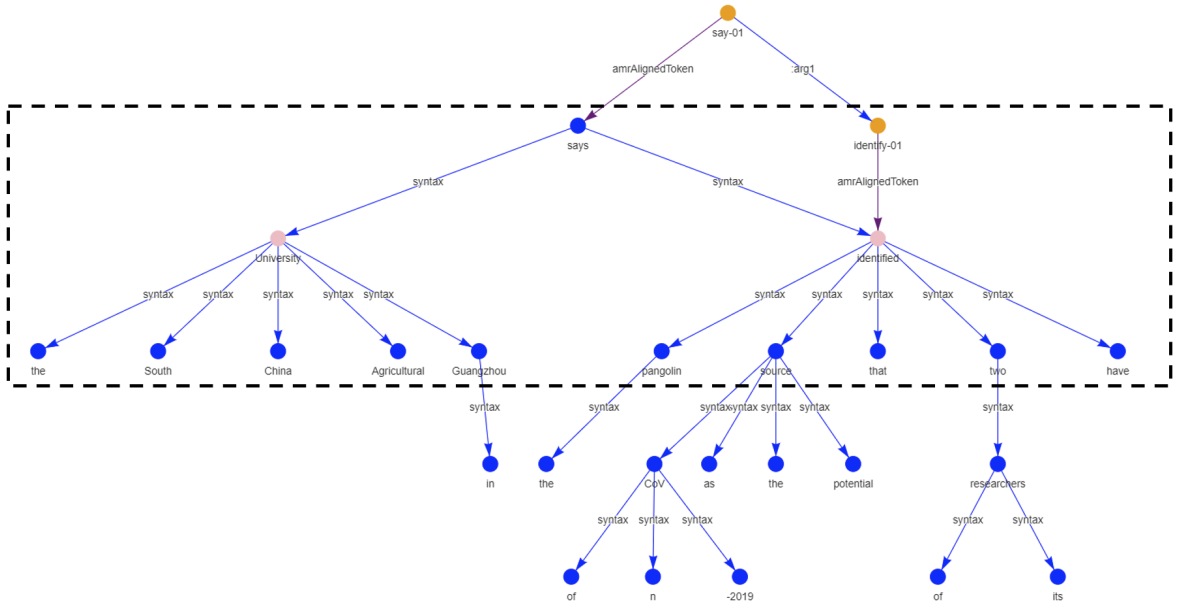


Figure 11: Composed syntactic and AMR parse for "*The Guangzhou South China Agricultural University says that two of its researchers have identified the pangolin as the potential source of COVID-19.*", with 1-hop neighborhood around claimer and inner claim head tokens. Note `amrAlignedToken` edges that connect an AMR node to the token it has been aligned to

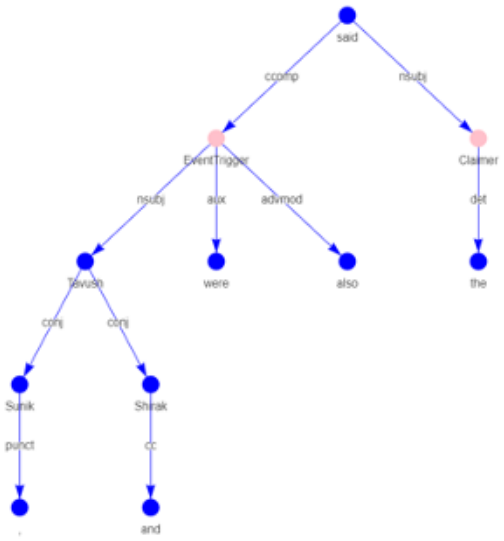


Figure 12: Example local neighborhood graph for dependency syntax parse of a claim

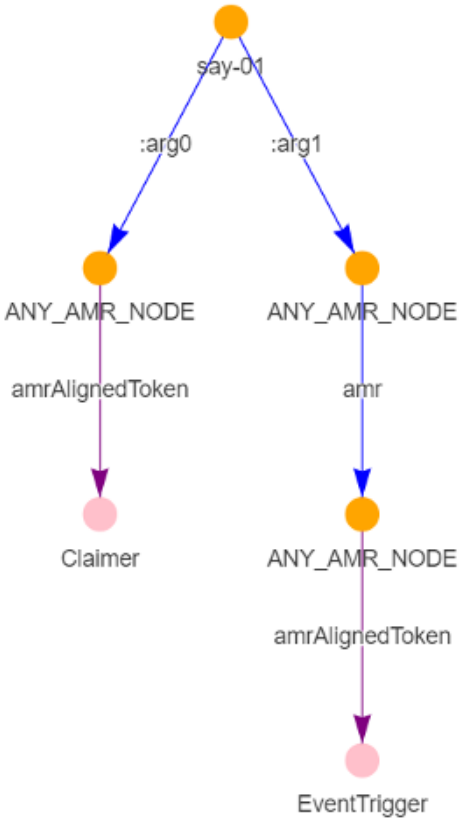


Figure 13: AMR pattern example, analogous to dependency syntax example in Figure 7

Which Argumentative Aspects of Hate Speech in Social Media can be reliably identified?

Damián Furman[1,2], **Pablo Torres**[3], **José A. Rodríguez**[3],
Diego Letzen[3], **Vanina Martínez**[4], **Laura Alonso Alemany**[5,6]
[1] Departamento de Computación, Universidad de Buenos Aires, Argentina
[2] Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
[3] Facultad de Filosofía, Universidad Nacional de Córdoba, Argentina
[4] Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain
[5] Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, Argentina
[6] Fundación Via Libre, Argentina

Abstract

With the increasing diversity of use cases of large language models, a more informative treatment of texts seems necessary. An argumentative analysis could foster a more reasoned usage of chatbots, text completion mechanisms or other applications. However, it is unclear which aspects of argumentation can be reliably identified and integrated in language models.

In this paper we present an empirical assessment of the reliability with which different argumentative aspects can be automatically identified in hate speech in social media. We have enriched the Hateval corpus (Basile et al., 2019) with a manual annotation of some argumentative components, adapted from Wagemans (2016)'s Periodic Table of Arguments. We show that some components can be identified with reasonable reliability. For those that present a high error ratio, we analyze the patterns of disagreement between expert annotators and errors in automatic procedures, and we propose adaptations of those categories that can be more reliably reproduced.

1 Introduction

With the impressive advances obtained in Large Language Models (LLMs), applications of automated language generation are quickly expanding to affect more and more areas of human activity, specially with the generalization of conversational chatbots. It is known that these models tend to amplify stereotypes, resulting in the naturalization of prejudices and finally the dehumanization of social groups in the form of hate speech.

Hate speech is a grave danger. The International Convention on the Elimination of all Forms of Racial Discrimination states that hate speech "*rejects basic human rights principles of human dignity and equality and seeks to degrade the position of individuals and groups in society's esteem*"[1].

Through the amplification provided by social media and LLMs, its effects are also amplified, as it can deepen prejudice and stereotypes (Citron and Norton, 2011). That is why great efforts have been made to detect and neutralize it. The most common form of neutralization to date has been banning hate speech from public forums. However, this strategy collisions with the right to freedom of expression. In addition, it is usually implemented by resorting to human moderators who are exposed to toxic content for long workdays.

Automatic argumentation analysis enables alternatives to censorship like argument retrieval and organization or automatic generation of counter-arguments. The recent developments of LLMs make these tasks more feasible. But, although they behave in a competent way from a purely conversational point of view, they do have not been designed to reason or argue. Moreover, they do not seem to be able to prevent harmful effects beyond very shallow guardrails, which is a critical concern when dealing with hate speech. That is why it seems necessary to enhance them beyond pure unannotated text, to obtain a more nuanced treatment of the argumentative dimension of texts.

The question remains, *how can we know which argumentative aspects will be useful for LLMs to improve their performance in nuanced, risky tasks like automatic generation of counter-arguments against hate speech in social media?*

In this work we present the Argumentation Structure Of Hate Messages Online (ASOHMO), a protocol to annotate argumentative information in hate tweets, and an annotated dataset of tweets to train automatic classifiers. These annotations are an adaptation of Wagemans (2016)'s proposal for hate speech in Twitter, where much of the argumentation refers to implicit elements, and one finds typos,

[1]United Nations Strategy and Plan of Action on Hate Speech: Detailed Guidance on Implementation for United Nations Field Presences, 2020.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 136–153
June, 2023.

incomplete phrases and incoherent syntax. Despite this challenging context, by applying our protocol, we obtained substantial agreement between different human judges to identify the argumentative structure of tweets. We also found that LLMs can successfully detect some of these argumentative components, even when few annotated examples are provided, which seems to indicate that it is feasible to finetune them to address some specific argumentation tasks and domains.

The rest of the paper is organized as follows. In the next Section we discuss relevant work, including the foundational Wagemans (2016)'s proposal. Section 3 describes the categories that we distinguish in our annotation framework, and in Section 4 we present how they apply to hate speech in social media, more concretely, to the manual annotation of the Hateval corpus (Basile et al., 2019). Finally, in Section 5 we show how LLMs can identify some argumentative components, but not others, with varying degrees of success. We analyze the causes of low success and propose how to adapt the definition of the target argumentative aspects to improve their reliability of annotation, both manual and automatic.

2 Relevant Work

There are many different proposals on how to model the argumentative aspects of texts, even if we only consider those aimed or used for computational application. We are not providing an exhaustive overview of approaches here, but just some examples to motivate and frame the model of argument that we present in this work.

One of the main distinctions between proposals is whether they are general purpose or domain specific. Domain-specific approaches propose tailored categories, like Teufel et al. (1999)'s "*background*", "*aim*" or "*comparison*" for scientific papers, or Al-Khatib et al. (2016)'s "*anecdote*" or "*statistics*" for the argumentative analysis of editorials. They tend to achieve good inter-annotator agreement and good accuracy in automatic identification, but are not portable to different domains.

General-purpose argumentation models have very different approaches. Many computation-oriented proposals are based on Toulmin (2003)'s theory of practical argument. They distinguish between two main components of arguments, "*conclusion*" (also called "*claim*") and "*fact*" (also called "*justification*" or "*premise*"). They usually try to identify relations between components and between arguments, aiming to create a full argument tree that accounts for the argumentative structure of a text. This kind of model has been applied to essays (Stab and Gurevych, 2014) or user-generated discourse (Habernal and Gurevych, 2017). It is very general, thus easily portable to different domains. At the same time, it is not very stable, since inter-annotator agreement is not high, and the information it provides about the argument is not as rich as in the case of domain-specific approaches.

Another approach to modelling argument in texts are schemes. Argument schemes are "*patterns of informal reasoning*" (Walton et al., 2008) that "*represent forms of argument that are widely used in everyday conversational argumentation*" (Macagno et al., 2018). Argument Schemes specify a pattern of reasoning and a set of critical questions oriented to test the defeasibility conditions on the pattern. This pattern and critical questions provide very insightful, actionable information about the argument, which can be later used for applications like building a counter-argument.

Several authors have adapted Walton's schemes to specific purposes, even proposing alternatives to critical questions (Atkinson and Bench-Capon, 2018; Kökciyan et al., 2018). The main drawback of these proposals is that the inventory of scheme is very profligate, and it has become clear that, identifying a scheme within a given text becomes quite difficult, both manually and automatically.

2.1 The Periodic Table of Arguments

Trying to find a trade-off between the excessive detail of schemes and the scarce information provided by claim-premise approaches, Wagemans (2016) proposes an analytic approach to argument schemes, aimed to obtain the core schemes proposed by Walton et al. (2008), with fewer categories based on a limited set of general argument features.

This is a characteristic that we find particularly useful for building a simple system that is easy to annotate without an enormous effort and achieving a high level of agreement between human annotators, which leads us to expect higher reproducibility in inferred models. Moreover, an analytic approach allows determining which aspects of argumentation are more feasible to detect automatically, and identifying which particular aspects are more useful for a given application, such as components that could

be used to elaborate a response.

All arguments under Wagemans's system have a premise and a conclusion labeled with one Type of statement each. But it goes beyond the mere premise-conclusion information. The PTA is a factorial typology of arguments that offers a comprehensive overview of the various types of arguments by describing them as a unique combination of three basic characteristics (Wagemans, 2019):

1. **first order or second order** argument. A common term between premise and conclusion transfers the acceptability from one to the other. If this common term is explicit, then it is a first order argument. If a reconstruction is needed, then it is a second order argument.

2. **predicate or subject** argument. If the common term is in the subject of the propositions making the premise and the conclusion, then it is a subject argument, otherwise, it is a predicate argument.

3. **policy, fact or value**. The conclusion and premise can be labeled each as a statement of policy (the speaker mandates or states that something should be done), a statement of value (the speaker issues an opinion about something), or a statement of fact (the speaker conveys a proposition as a true fact).

Visser et al. (2021) conducted an exhaustive research on annotating the US 2016 presidential debate corpus using both Walton's schemes and Wagemans's Periodic Table of Arguments. They reported a higher agreement for Wagemans's typology, specially without considering classification between first and second order arguments. Moreover, they sustain that for Wagemans's typology, "*the division into independent sub-tasks simplifies the annotation while maintaining reliability*".

We adapted Wagemans's proposal to hate speech on social media, with the goal of identifying elements that can be relevant to either a human or a machine in the task of analyzing or countering hate speech.

Focusing on hate speech on Twitter, we have to take into account that many argumentative hate tweets are based on assumptions justified by prejudice or context information that is difficult to recover. This means that in many cases, it is difficult to rebut them from the perspective of formal deductive logic. We believe that an approach based on informal logic, like the one proposed by Wagemans (2016), is more adequate to capture this kind of arguments that are organized with informal relations.

In the following Section we describe our approach. We provide an overview of other social media corpora annotated with argumentative information in Appendix H.

3 A Framework to Identify Argument Components on Twitter Hate Speech

The goal of our argumentation model is to provide an argumentative analysis that can help expose the core of the reasoning supporting a hate message. We believe that this can help both humans and automatic models to better address hate speech.

We are labeling two kinds of information: domain-specific and argumentative-general. Domain-specific information allows to exploit particular characteristics of hate messages on Twitter: they always mention a collective that is implicitly or explicitly associated with a negative property, action or consequence. Argumentative-general structure is based on a simplification of Wageman's proposal that is aimed to increase inter-annotator agreement. Reaching acceptable levels of inter-annotation agreement is very important to our purpose, as it indicates that the annotation process can be systematized and possibly automatized.

We created an annotation protocol[2] where both kinds of argumentative information are defined in a procedural manner. This protocol was applied by human analysts to annotate hate speech tweets, with five steps that are described as follows. The annotation team and environment are described in Appendix A.

3.1 Argumentative or Non-argumentative

Following (Wagemans, 2019), "*an argument (...) consist(s) of two statements, namely a conclusion – the statement that is doubted – and a premise*". We consider a tweet to be argumentative if it is possible to divide it in these two components. Examples of non-argumentative tweets can be found in Appendix E: exhortations to some action without justification, insults, name callings, support for a particular policy or description of facts without an explicit conclusion.

[2]Annotation guidelines can be found at `shorturl.at/cv458`.

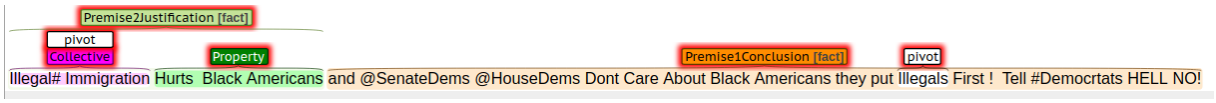


Figure 1: Example of labeled argumentative hate tweet.

3.2 Domain-specific components: Collective and Properties

All hate messages are directed towards a specific group by definition. Usually, the content of the message is to associate this group with a negative property or an undesirable action or consequence. If this property[3] is explicit, we label it.

3.3 General argumentative components: Justification and Conclusion

All argumentative tweets are labeled with one and only one Conclusion and Justification, though these can be separated in many non-contiguous parts inside the tweet. Annotators were instructed to choose the longest Conclusion and Justification that they could find, leaving out only hashtags indicating topics, links, user mentions or non-relevant words or information. Justifications may be arguments themselves, having their own inner structure involving different premises, but this is not annotated as we are only interested in capturing the main standpoint that the user wants to gain acceptability for.

When labeling these components, we are not considering the subject-predicate structure proposed by Wagemans. Visser et al. (2021) warned about how this model presupposes that premises and conclusions of arguments consist of complete categorical propositions comprising a clear subject-predicate structure, which is not always the case in user-generated, informal social media text.

3.4 Argumentative relation: Pivot

Following Wagemans, argumentative components transfer reason from one to the other. We assume that there can be textual cues of this transfer, in the form of an element that is common to both components. We call this element the pivot. We identify pivots as two sequences of words, one for each premise, that can be related to the element that those premises have in common.

This relation is generally not unique; the underlying common ground between the premises could be expressed in different forms or could present multiple aspects signaled by different words. Whenever this element is explicit in the text (it might be not), we annotate it.

The pivot holds a relation with Wagemans's categories of first and second order arguments. If an argument is considered first-order, it means that the common element between premises must be explicit (by definition, it must be either of the form A is X because A is Y or B is Z because C is Z). For a second-order argument, there might still be an explicit pivot or not.

3.5 Types of Proposition

Wagemans proposes "a characterization of the types of arguments based on the combination of the types of propositions they instantiate" (Wagemans, 2016). These types are taken from debate theory (Schut, 2014), where three distinctions on propositions are made: (1) policy (P), (2) value (V) and (3) fact (F).

We label our propositions using the same types and add to our annotation manual different guidelines on how to recognize each one: a policy proposition is a mandate often expressed as orders, imperatives, or actions that need to be accomplished in the public domain. Fact and value propositions were reported to be more difficult to differentiate. As a general criterion, a proposition to be labeled as value must have explicit markers of the speaker being involved in the assertion expressed (opinionated adjectives, verbs of thought, etc.). Otherwise, the premise is considered as fact. Examples can be seen in Appendix E.

4 The ASOHMO Corpus

We applied our argumentation model via the annotation protocol described in the previous Section to the HatEval 2019 corpus (Basile et al., 2019). Focusing on argumentative tweets, we did not annotate tweets labeled as "aggressive", consisting mostly of abusive language (name callings, insults, exhortations to action and other types of attacks), nor tweets targeted against specific individuals or women, as they were almost exclusively abusive and non-argumentative. After these filters, a corpus

[3] A property is anything that is associated with the targeted community, whether it is an adjective, a consequence, an action, etc.

		Domain-specific		Argument-general				
	Argumentative	Collective	Property	Pivot	Justif.	Concl.	Type of Conc.	Type of Just.
κ	.85	.64	.60	.52	.62	.64	.60	-.03

Table 1: Agreement scores between two annotators for 150 tweets.

of 970 tweets in English and 196 tweets in Spanish remained.

The dataset is released[4] for the free use of the scientific community, together with the scripts for reproducing experiments.

4.1 Inter-annotator Agreement

We calculated inter-annotator agreement to assess the reproducibility of the annotations and the feasibility of automatic identification. While the whole corpus was annotated by a single annotator, 150 tweets (15% of the corpus) were labeled by a second annotator[5]. Then, per-category agreement was calculated with Cohen's κ (Cohen, 1960). Agreement was calculated in a per-tweet basis for the Argumentative vs. Non-Argumentative category using a binary label, and for the Type of Conclusion and Justification categories, using one label with three possible values representing *fact*, *value* and *policy*. For all other categories, agreement was calculated in a per-word basis with a binary label assigned to each word, marking whether it belongs to the category or not.

In Table 1 we can see that annotators can reach a substantial level of reproducibility, around $\kappa = .6$ for Collective, Property, Justification, Conclusion and Type of Conclusion and .85 for the distinction between Argumentative or non-Argumentative tweets. In contrast, the Pivot presents a moderate level of inter-annotator agreement, and the Type of Justification presents no agreement at all.

To calculate agreement, we follow a criterion similar to that of Visser et al. (2021): while comparing two annotators, if at least 50% of the words in the smallest component marked by one of the annotators overlaps with words marked by the other one, then it is considered an agreement. For example, if one annotator marked "*the damage illegals do*" as a Property associated to a Collective and the other annotator marked just "*damage*" as a Property we consider that 100% of the words in the shortest "*damage*" in both examples and assume that all the other words are marked as not being part of the Property in both cases.

@user @user *sanctuary cities are* ***against the law.*** PLEASE SHUT THEM DOWN & ARREST/PROSECUTE ALL **CRIMINAL** GOVERNORS & MAYORS

Figure 2: Disagreement concerning Pivot. One annotator is underlined, while the other is bolded. Justification is marked with italics and Conclusion with capitalization

When inspecting examples of disagreement between annotators for Pivot, as shown in Figure 2, we found that in many cases both annotations could be considered accurate, as there may be more than one possibility for annotators to tag. Furthermore, as the relation is very deep in the layers of meaning, annotators may interpret it as signalled by different surface features, and as a consequence they may tag different sequences of words while considering the same relation.

Finding patterns in the disagreements between annotators can be used to redefine categories (Teruel et al., 2018). In a second annotation phase, we will be redefining the Pivot category to obtain more agreement between annotators. We understand that this element is particularly challenging, because it signals a very deep relation and its correspondence with surface textual phenomena may not be direct, or multiple. That is why we plan to rethink it as a binary classification problem, where human judges are presented with one or more possibilities of Pivots for a given argument, and they have to say whether they consider any of them to be a valid Pivot for the example.

5 Automatic Identification of Arguments

We conducted several experiments to assess the feasibility that LLMs can automatically identify different argumentative aspects.

For each set of hyperparameters used, we fine-tuned the same language models using different random tweets for each partition, always respecting this proportion. We report the average of these three fine-tuned models' F1, Precision and Recall to detect or classify argument components. For

[4] https://github.com/ASOHMO/ASOHMO-Dataset

[5] The sample's size for the test is proportionally higher than many of the previous works: Bosc et al. (2016) used 100 tweets to calculate agreement over a dataset of 4000 whereas Dusmanu et al. (2017) used 100 tweets for its first dataset of 1887 tweets, 80 tweets for its second dataset of 1459 tweets and used the whole third dataset of 368 tweets.

	RoBERTa			BERTweet			XLM-RoBERTa-Mix			XLM-RoBERTa-XL		
	F1	Pr	Rec	F1	Pr	Rec	F1	Pr	Rec	F1	Pr	Rec
Arg./Non-Arg.	**.89**±.02	.84	.95	.88±.01	.84	.93	.87±.04	.84	.91	.84±.03	.84	.85
Justification	.73±.05	.69	.76	**.77**±.05	.75	.78	.76±.05	.71	.81	.75±.01	.71	.80
Conclusion	.55±.02	.60	.51	**.61**±.03	.64	.58	.60±.02	.59	.61	.54±.03	.59	.49
Type of Just.	.41±.09	.48	.39	**.42**±.09	.48	.41	.35±.05	.34	.37	.33±.03	.33	.35
Type of Conc.	.58±.05	.62	.57	**.65**±.11	.67	.65	.61±.06	.65	.62	.63±.02	.66	.62
Collective	**.59**±.03	.56	.64	.58±.05	.55	.62	**.59**±.06	.58	.60	.27±.07	.41	.21
Property	.46±.04	.52	.41	.47±.03	.50	.43	**.50**±.03	.57	.43	.42±.04	.42	.43
Pivot	**.45**±.04	.52	.41	.40±.08	.43	.39	.39±.08	.42	.38	.33±.08	.41	.27

Table 2: F1, precision and recall for the target class in the automatic detection of argument components in tweets. Each experiment was carried out with three randomized partitions, the mean and standard deviation of the F1 are presented. Best results for F1 for each category are highlighted in boldface.

multi-label classification, the macro average is calculated, otherwise, we report the score of the target class. We also report per-class F1 scores for the three possible Types of premises: Fact, Value and Policy.

Models. We fine-tuned the following LLMs:

RoBERTa (Liu et al., 2019): a BERT-like (Devlin et al., 2018) LLM, pre-trained with more data.

BERTweet (Nguyen et al., 2020): a RoBERTa-based LLM trained on data from Twitter.

XLM-Roberta (Conneau et al., 2019): a RoBERTa based multilingual LLM. We fine-tuned it with a Mixed Language (Mix) version using both English and Spanish for training and testing and with a Cross-Lingual version (XL) using English for training and Spanish for testing.

5.1 Predicting Individual Components

We trained different kinds of models to automatically recreate the annotation process one component at the time: one for sequence binary classification, to predict if a tweet is argumentative or not; five models for token classification, to predict for each word, if it is labeled as part of the collective, the property associated to that collective, the pivot, the justification or the conclusion, respectively; and two models for sequence classification, fed only with the correspondent text of the premise (Justification or Conclusion), to predict the Type associated with it (fact, value or policy). Results of this experiment are shown in table 2.

Distinguishing argumentative from non-argumentative tweets achieves a very satisfying .89 F1. In general, components with higher inter-annotator agreement perform better, with justifications identified with .77 F1. Components with low inter-annotator agreement are also identified with more errors: conclusions have an F1 of .61 ($\kappa = 64$), collective F1=59, ($\kappa = 64$), property F1=.50 ($\kappa = 60$) and finally pivots only reach an F1 of .45 ($\kappa = 52$).

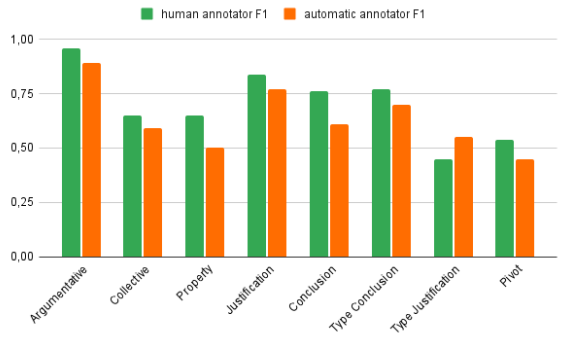


Figure 3: F1 score for "predictions" done by a human annotator and compared with predictions done by the best performing automatic classifiers (BERTweet for Justification, Conclusion and their Types - trained with all the premises -, Roberta for Argumentative and Pivot and XLM-Roberta for Collective and Property).

In Figure 3 we compared the F1 scores of the best performing models with inter-annotator agreement. We calculated the F1 score of the 150 tweets labeled by two judges using one as the ground truth and the other as the one being evaluated. We can see that both scores are highly correlated, although human annotators tend to agree slightly more than the automatic predictor with respect to the human ground truth, so there is still room for improvement for automatic predictors.

Analyzing predictions for the worst performant components (Property and Pivot), we can see that the models predicting Properties have a tendency to recognize any word with a negative charge, disregarding if it is referring to the Collective itself. Models recognizing Pivots sometimes find more than one possibility to label. Figure 4 shows how the model predicts the real pivot, but then also predicts another one not labeled on the original example that could be valid. This shows that, at least partially, some mistakes are made because of

the subjective nature of the task and the multiple valid possibilities of labelling. To overcome this problem, annotators should consider the possibility of multiple pivots and try to label them all.

Salvini prosecuted for defending italian sovereignty and finally preventing hundreds of migrants to **invade** Italy grande **Salvini**, help us preserve the european culture against the **invasion** #StopIslamization #ComplicediSalvini #StopInvasion #RefugeesNotWelcome

Figure 4: Example of prediction of Pivot. Labeled justification is in blue, while conclusion is red. Real pivot is underlined, while predicted Pivot is bolded.

Regarding the different models, BERTweet achieves the best performance on most experiments involving Justifications, Conclusions or their types, and is close to the best results on other components. RoBERTa achieves higher results for Pivots.

Multilingual experiments achieve a performance similar to their monolingual counterparts for most components, specially Properties, indicating that training with mixed languages does not decrease, and can even improve, performance.

Results on cross-lingual experiments where models are trained with English and tested against Spanish, on the other hand, show different behavior depending on the component: for finding argumentative tweets, Justifications, Types of Justification and Types of Conclusion, results are similar to their counterparts on monolingual experiments. Collective, on the other hand, has a major drop in performance for all experiments compared to all other model settings. This is explained because of the very specific lexicon used for naming collectives, with lots of out of vocabulary and slang words. The pivot also suffers a drop in performance on both multilingual settings, but more so on cross-lingual.

5.2 Predicting Components Simultaneously

The goal in this case is to measure the performance of the models when simultaneously predicting components labeled on the same annotation step. We want to assess whether training with information about both components helps to improve the performance when predicting each of them individually or not. We ran an experiment to jointly predict Collective and Property and another for Justifications and Conclusions. Each word is assigned one of three labels, indicating if they belong to either of the two searched components or not.

Joint prediction of components labeled on the same annotation step produces almost the same results as predicting them individually. This has the advantage of consuming half of the resources and time; however, the definition of the problem changes, as each token can only be part of one or none component, but not both.

5.3 Predicting the Type of Premises

The Type of Conclusion or Justification (Fact, Policy or Value) should be independent of its premise (Justification or Conclusion), so in terms of semantic information, to predict this, it should not matter if models are trained with just one or both of them.

Moreover, using both kinds of premises increases the number of training examples and can help to overcome the unbalance between Fact and Policies (specially on Justifications, where facts are the vast majority).In Table 3 we can see that models trained to predict the Type of Premise with both Justifications and Conclusions perform much better than models trained with just one or the other. For Type of Justification, these models achieve F1 scores that are between 10 and 20 points higher. For Type of Conclusions, their F1 scores are around 5 points higher. When checking the per-class F1 scores, the improvement in performance is concentrated on the minority classes. For Type of Justification, both Value and Policy classes improve highly, and for Type of Conclusion the most difference is on the Value class.

5.4 Impact of training dataset size

We want to assess how much data is needed for the models to achieve an acceptable performance. For this purpose, we ran several experiments following the same settings as in 5.1 but using smaller portions of the original datasets. Our goal is to measure the impact of having smaller datasets for each component and the relative gain of adding new examples, considering that the task of labeling them is expensive. We used a random sample of 25%, 50% and 75% of the corpora used for training and compare the F1 scores with those obtained by the models trained with the whole corpus.

Figure 5 compares the F1 scores of the best performing models for each component in 5.1 with those obtained by the same models trained with smaller portions of the same datasets. On the left, we show the evolution of the F1 score when increasing the size of the training dataset. On the right, we show the percentage of improvement of the F1 score between each size of the dataset for each com-

	RoBERTa				BERTweet				XLM-RoBERTa-Mix				XLM-RoBERTa-XL			
	Macro	F	V	P	Macro	F	V	P	Macro	F	V	P	Macro	F	V	P
Models trained with both Justifications And Conclusions																
Type of Just	.49±.07	.92	.13	.41	.53±.08	.93	.19	.47	**.55**±.01	.94	.37	.34	.52±.17	.93	.41	.21
Type of Conc	.63±.14	.82	.22	.85	**.70**±.14	.85	.37	.87	.67±.16	.78	.37	.86	.57±.04	.78	.34	.60
Type of both	.66±.05	.90	.28	.79	**.69**±.12	.91	.34	.82	.67±.04	.88	.35	79	.60±.03	.89	.39	.53
Models trained with just one of them																
Type of Just	.41±.09	.95	.28	.00	**.42**±.09	.95	.13	.17	.35±.05	.97	.00	.08	.33±.03	.95	.0	.05
Type of Conc	.58±.05	.81	.07	.85	**.65**±.11	.84	.20	.89	.61±.06	.70	.28	.85	.63±.02	.78	.45	.65

Table 3: Results for identification of Type of premises tested against both Justification and Conclusion, only Justifications and only Conclusions. Results are compared against those achieved by the best performing model trained with only one of the two kinds of premises.

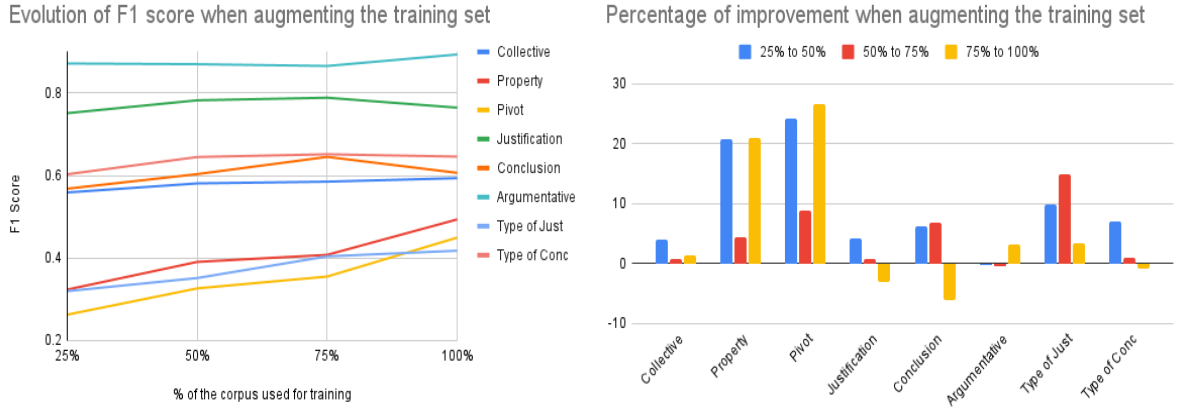


Figure 5: Evolution of F1 scores per argumentative component when increasing the size of the dataset used for training. The figure on the left shows F1 score absolute values, while the one in the right shows the percentage of the score wrt the final value obtained when reducing the dataset.

ponent. For example, the model predicting Pivot trained with 75% of the dataset achieved a score of 0.36 while the model trained with the whole dataset (100%) achieved a score of 0.45, which represents an improvement of 26.6% of this score.

When looking at performance of models trained with smaller fractions of the dataset (figure 5.1) we can see that those components with better scores can achieve similar results using fewer data, while components with worse performance (Property, Pivot and Type of Justification) are much more sensible to the amount of examples on the dataset. This could be considered as an indicator that the size of the dataset is enough for most components but for these last three, if more examples were added to the dataset performance could improve.

6 Discussion of results

We have seen that some argumentative aspects of hate speech in tweets can be successfully identified by Large Language Models (LLMs), namely, whether a tweet is argumentative or not and Justifications, Conclusions and the Type of Conclusions.

This kind of information may be useful to provide an argumentative analysis of tweets, possibly for argument retrieval. It is probably also useful to guide the (semi-)automatic generation of some counter-narratives, like those that are aimed to question the Justification or those aimed to some kinds of Conclusions, like Values or Policies.

Domain-specific argument information, like Collective and Property, are not very successfully identified. Different strategies, like Named Entity Recognition approaches, may yield better results.

Pivots, aiming to identify the relation between Justification and Conclusions, and a key component to reconstruct Wagemans's typology, cannot be successfully identified, either by humans or automatically. It seems that a different approach must be taken to identify them manually, possibly identifying all possible sequences of words that elicit a relation between Justification and Conclusion.

These results will be instrumental for the annotation of a bigger annotated corpus, specially for

Spanish, and to integrate these concepts into LLMs.

7 Summary and Future Work

We have presented an approach to determine which aspects of argumentative information from hate speech in social media is liable to be integrated into LLMs. We have adapted the analytic approach of an informal logic based on (Wagemans, 2016) and have developed annotation guidelines which have then used to enrich a reference dataset for hate speech with argumentative information.

We developed a robust annotation process and guidelines to obtain high agreement between annotators. Indeed, an initial assessment of inter-annotator agreement, shows agreement above $\kappa = .6$ for most categories, except the most interpretative ones. Considering we are dealing with user-generated text, we find this a very hopeful scenario. We are also working on adapting the categories with more disagreement, like Pivot, based on the patterns of the disagreemeent between annotators, so that in further annotation efforts they can be identified in a more reproducible ways, both by humans and automatic methods.

We show to which extent it is possible for Large Language Models to automatically identify the argumentative components, so that this kind of information can be integrated with purely data-driven approaches to enrich the analysis of text and produce more insightful, reasoned outputs.

Finally, the published dataset is also a contribution to the existing corpora of argument mining on social networks. It is publicly available at `https://github.com/ASOHMO/ASOHMO-Dataset`.

For future work, we plan to annotate bigger corpora, focusing on improving reliability on difficult, yet potentially useful, components, like Pivot. We also plan to add counter-narratives associated to each tweet and train models to automatically generate them. We want to assess to which extent the argumentative information helps in better generating automatic responses.

8 Limitations and Ethical Considerations

In the first place, we would like to make it clear that the human annotations presented here are the result of the subjectivity of the annotators. Although they have been instructed through a manual and training sessions, there are still significant variations between interpretations, and further researchers may assign different categories to examples.

Also, it is important to note that the automatic procedures obtained are prone to error, and should not be used blindly, but critically, with attention to possible mistakes and how they may affect users, groups and society.

Then, it is also important to note that the corpus used for this research is very small, specially in the Spanish part, so the results presented in this paper need to be considered indicative. A bigger sample should be obtained and annotated to obtain more statistically significant results.

The findings of this research can potentially inform the development and improvement of language models and chatbot systems. However, we emphasize the importance of responsible use and application of our findings. It is essential to ensure that the identified argumentative components are utilized in a manner that promotes reasoned usage and does not contribute to the spread of hate speech or harmful rhetoric. We encourage researchers and developers to consider the ethical implications and societal impact of incorporating argumentative analysis into their systems.

The data have been adequately anonymized by the original creators of the Hateval corpus.

Studying hate speech involves analyzing and processing content that may be offensive, harmful, or otherwise objectionable. We acknowledge the potential impact of working with such content and have taken steps to ensure the well-being of the research team involved. We have provided comprehensive guidelines and training to our annotators to mitigate any potential emotional distress or harm that may arise from exposure to hate speech. Additionally, we have implemented strict measures to prevent the dissemination or further propagation of hate speech during the research process.

Finally, we have not specifically conducted a study on biases within the corpus, the annotation or the automatic procedures inferred from it, nor on the LLMs that have been applied. We warn researchers using these tools and resources that they may find unchecked biases, and encourage further research in characterizing them.

Acknowledgments

Annotation was done using the brat annotation tool (Stenetorp et al., 2012). This work used computational resources from CCAD – Universidad Nacional de Córdoba (https://ccad.unc.edu.ar/), which are part of SNCAD – MinCyT, Argentina.

References

Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. A news editorial corpus for mining argumentation strategies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3433–3443, Osaka, Japan. The COLING 2016 Organizing Committee.

Katie Atkinson and Trevor Bench-Capon. 2018. Taking account of the actions of others in value-based reasoning. *Artificial Intelligence*, 254:1–20.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. pages 54–63.

Muhammad Mahad Afzal Bhatti, Ahsan Suheer Ahmad, and Joonsuk Park. 2021. Argument mining on Twitter: A case study on the planned parenthood debate. In *Proceedings of the 8th Workshop on Argument Mining*, pages 1–11, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Bosc, Elena Cabrio, and Serena Villata. 2016. DART: a dataset of arguments and their relations on Twitter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1258–1263, Portorož, Slovenia. European Language Resources Association (ELRA).

Danielle Keats Citron and Helen Norton. 2011. Intermediaries and hate speech: Fostering digital citizenship for our information age. *BUL Rev.*, 91:1435.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357.

Mihai Dusmanu, Elena Cabrio, and Serena Villata. 2017. Argument mining on Twitter: Arguments, facts and sources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2317–2322, Copenhagen, Denmark. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Nadin Kökciyan, Isabel Sassoon, Peter Young, Martin Chapman, Talya Porat, Mark Ashworth, Vasa Curcin, Sanjay Modgil, Simon Parsons, and Elizabeth Sklar. 2018. Towards an argumentation system for supporting patients in self-managing their chronic conditions.

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.

Fabrizio Macagno, Douglas Walton, and Chris Reed. 2018. *Argumentation Schemes*, pages 517–574.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, and Valerio Basile. 2019. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *6th Italian Conference on Computational Linguistics, CLiC-it 2019*, volume 2481, pages 1–6. CEUR.

Iyad Rahwan and Guillermo R Simari. 2009. *Argumentation in artificial intelligence*, volume 47. Springer.

Robin Schaefer and Manfred Stede. 2020. Annotation and detection of arguments in tweets. In *Proceedings of the 7th Workshop on Argument Mining*, pages 53–58, Online. Association for Computational Linguistics.

Robin Schaefer and Manfred Stede. 2022. German climate change tweet corpus (gercct).

& Wagemans Schut, D. 2014. Argumentatie en debat.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Milagro Teruel, Cristian Cardellino, Fernando Cardellino, Laura Alonso Alemany, and Serena Villata. 2018. Increasing argument annotation reproducibility by using inter-annotator agreement to improve guidelines.

Simone Teufel, Jean Carletta, and Marc Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 110–117, Bergen, Norway. Association for Computational Linguistics.

Stephen E. Toulmin. 2003. *The Uses of Argument*, 2 edition. Cambridge University Press.

Jacky Visser, John Lawrence, Chris Reed, Jean Wagemans, and Douglas Walton. 2021. Annotating argument schemes. *Argumentation*, 35.

Jean Wagemans. 2019. Four basic argument forms. *Research in Language*, 17:57–69.

Jean H. M. Wagemans. 2016. Constructing a periodic table of arguments.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.

APPENDIX

A Annotation team and environment

Two annotators (a philosopher and a computer scientist) have been trained with the guidelines described in section 3, with a three stage training process, where they labeled a first set of examples, discussed their difficulties, systematized further hints and criteria, updated the annotation manual and started again. We prioritized having the lesser amount of annotators doing the most possible amount of annotations. Our hypothesis is that the more annotators, the more difficult it is to reach a uniform criterion that can be understood in the same way by everyone. So fewer annotators doing more work should lead to more reliable annotations and to better inter-annotator agreement.

The average time for annotators to label a tweet is approximately 4 minutes per example. The annotation time changes depending on whether the tweet is argumentative or not. For argumentative tweets, the average time is around 5 minutes, while for non-argumentative tweets the average time is less than 1 minute.

The first annotator annotated 800 tweets in English and 196 in Spanish, while the second annotated 170 tweets in English.

B Corpus statistics

Table 4 show the percentage of tweets that are labeled as non-argumentative in English and in Spanish, and also the percentage of tweets in each language that have a pair of Collective and Property and a Pivot labeled. Considering only the non-targeted and non-aggressive hate tweets against immigrants from HatEval, the majority of tweets are labelled as Argumentative in both languages. Regarding the Collective-Property pair and the pivot, the table shows the percentage of the final dataset that have them labeled. Table 5 shows the percentage of Justifications and Conclusions that are labeled as **F**act, **P**olicy or **V**alue. Justifications have an ample majority of examples labeled as Fact, while the distribution between classes is more even when observing conclusions. In both cases, the "Value" class is the least frequent.

C Preprocessing

Preprocessing is very important when dealing with tweets, since they tend to have lots of non-alphanumeric characters, user handles (@user-

	Non-Arg	Collective	Pivot
English	25.3%	58.2%	45.1%
Spanish	26.5%	61.1%	37.5%

Table 4: Percentage of tweets labeled as Non-Argumentative and with Collective-Property and Pivot labeled

	Justification			Conclusion		
	F	P	V	F	P	V
English	93%	4%	3%	37%	57%	6%
Spanish	97%	2%	1%	56%	28%	16%

Table 5: Percentage of Justifications and Conclusions labeled as **F**act, **P**olicy or **V**alue

name), hashtags, emojis, misspellings, and other non-canonical text. Following (Nguyen et al., 2020) and (Polignano et al., 2019) we used a soft normalization strategy consisting of:

- Character repetitions are limited to a max of three
- User handles are converted to a special token `@usuario`
- Hashtags are replaced by a special token `hashtag` followed by the hashtag text and split into words if this is possible
- Emojis are replaced by their text representation using *emoji* library[6], surrounded by a special token `emoji`.

D Experiment settings

For all monolingual experiments we used 770 tweets of the English portion of the dataset as training (79%), 100 tweets as development (10.5%) and 100 tweets as test (10.5%). Multilingual experiments were twofold: using both English and Spanish for both training and testing, and using English for training and development and Spanish for test. In the first case, we used 770 English and 120 Spanish tweets as training (76.3% of the dataset), 100 English and 26 Spanish tweets as development (10.8%) and 100 English and 50 Spanish tweets as test (12.9%). In the second case, we used 850 English tweets as training (73% of the total), 120 English tweets as development (10%) and all the 196 Spanish tweets for testing (17%).

In all cases, we tried 5 different values for learning rate (1e-05, 2e-05, 5e-05, 5e-04 and 5e-06) and used the development dataset to implement early stopping with a maximum of 10 epochs. Table 6 shows the values for the hyperparameters used on all models trained with our examples.

Batch Size	16
Optimizer	AdamW
Dropout	0.1
Epochs	10
Weight Decay	0.01
Adam ϵ	1e-06
Adam $\beta 1$	0.9
Adam $\beta 2$	0.99

Table 6: Hyperparameters used for training all models used on our experiments

E Examples And Decisions From The Annotation Process

In the following section, we show examples of labeled tweets that illustrate particular decisions taken when defining the annotation protocol. Example 6 shows a frequent case of a non-aggressive, non-targeted and non-argumentative tweet, consisting on the expression of one or many stances or exhortation to one or many actions but without any explicit intention of connecting them.

Example 7 shows an example of a premise where a user states her opinion as a verifiable fact. Although it could be arguable that she is expressing a Value about a subject (immigration or assimilation), we consider all tweets that could be fact checked (specially if the user doesn't use explicit markers of her involvement in the statement) to be of type Fact. Example 8 shows an annotation of a Collective-Property pair. The Property is any negative concept, adjective, consequence or aspect of reality that is explicitly or implicitly associated with the target of the hate message. In this case, the tweet is stating that immigrants are not wanted by the people. The cases where there is no explicit association between Collective and Property are diverse, but we present three examples that we believe represent the majority of the cases. Example 9 shows a case where instead of defining a negative property associated with the targeted collective, the user defines a positive Property associated with the absence of that Collective. Example 10 shows a case where a negative Property is associated with the targeted Collective but in an indirect manner that must be reconstructed using contextual information not included in the hate message. In this case, the reference is made through the mention of "Operation SOAR", an operation made by the

[6] https://github.com/carpedm20/emoji/

ICE in the United States specifically targeting immigrants registered as sex offenders and by the hashtag "StopTheInvasion" referring to a narrative built against immigrants as if there were a coordinated plan to invade a country. Example 11 shows a case where the main standpoint of the tweet is an action that must be taken and there is no explicit mention of any Property. In these cases, the Property could potentially be reconstructed by appealing to find the motivation of these advertised actions, but it can not be labeled explicitly on the text of the tweet.

Example 12 shows a Justification labeled as Fact and a Conclusion labeled as Policy. The main standpoint of the message must be first identified as Conclusion, and then any part of the tweet that fulfills the role of providing reasons for that standpoint is identified as Justification. One typical pattern frequent in many tweets is to express a mandate or policy that must be followed, usually with the form of a phrase or hashtag using the imperative mode, and a Fact (or less frequently also another mandate) that supports and aims to explain why that mandate must be followed.

Example 13 show a tweet with a Pivot. In this case, the user binds the "money" as a cause of immigration to conclude that "money" is not needed. All tweets present some aspect that links Justification with Conclusions, but not always that relation is mentioned directly. Example 14 shows a tweet where no Pivot was labeled. The link between the premises relies on the implicit assumptions that the hate that they supposedly bring to the EU is against Christians and that because of that hate, Christians are not safe. But to recognize it, the relationship must be reconstructed using implicit information defined by the context, otherwise it is impossible to establish. In these cases, we do not label the Pivot for the sake of simplicity. We require that the relation between the two phrases constituting a Pivot is direct and easy to spot.

```
No to #EU migrant camps in Libya,
PM al-Serraj
```

Figure 6: Example of non-argumentative tweet

```
@user Time to leave the uk
commonwealth and Europe that
would end immigration people do
not want more refugees enough is
enough
```

Figure 8: Example of Collective and Property labeled. Collective is underlined while Property is bolded

```
Good this makes it a safe country
immigrants can now go home
```

Figure 9: Example of tweet without Collective and Property labeled. In this case, Property is associated with the absence of immigrants, therefore it is indirectly defined and not mentioned explicitly

```
Anyone who, ACTIVELY OR
PASSIVELY, subscribes to
immigration and especially
assimilation is joining the
battle to destroy White
```

Figure 7: Premise of a tweet labeled as “fact"

F Disagreement between annotators

In the following section, we analyze examples of disagreement between annotators to better understand the aspects that are most difficult to systematize about annotating argumentative components. Example 15 show a disagreement concerning Collective and Property. Here, one annotator didn’t consider that there was a Collective and Property to label, while the other did. We found that most disagreements regarding these components are of this kind. If both annotators agree that the tweet has a Collective and Property to label, in most cases they agree also what parts of the text constitutes them. In the few cases where both annotators labeled a Collective and a Property, but they did not match exactly, they had a major overlap and only differed on adding a few more words at the beginning or at the end. Example 16 shows a disagreement of

```
ICE officers arrest 32 sex
offenders on Long Island as
part of 'Operation SOAR' :link:
#StopTheInvasion #SecureTheBorde
```

Figure 10: Example of tweet without Collective and Property labeled. In this case, the collective is not explicitly mentioned but referred through contextual information

Canada is an immigrant country Don't change it to refugee country please

Figure 11: Example of tweet without Collective and Property labeled. In this case, the focus of the message is put into an action that must be taken and not on associating the Collective with a Property

Victims of Illegal Alien Crime describe heartbreak, frustration #BuildTheWall #ProtectAmerica #EndChainMigration #EndIllegalBirthrightCitizenship #NeverForget the American Victims of Illegal Alien Migration

Figure 12: Example of labeling of Justification (in blue) and Conclusion (in red). Justification is labeled as Fact while Conclusion is labeled as Policy

Why do foreign individual dump **money** (and refugees) into our country? We don't need their **money** and their programs.

Figure 13: Example of a tweet with a labeled Pivot. Justification is shown in blue while Conclusion is in red. Labeled Pivot is shown bolded

Nice tweet , Joyce, Truth is they flee Iran etc but want to bring their hate to the Eu even in refugee camps Christians not safe.

Figure 14: Example of a tweet without a Pivot labeled. The Justification is shown in Blue while the Conclusion is in Red. The link between the two premises relies on the relation between "hate" and "not safe"

@user @user The idea is to bring in the "dreamers" **so that they vote for Democrats** because Dems know they have to import their voters. That is literally the only reason the Democrats care about this issue. In the meantime, YES THEIR PARENTS

Figure 15: Example of disagreement concerning the Collective and Property. One annotator did not label any of them. Collective labeled by the other annotator is underlined while Property is bolded

Mexico's not sending their best. **They are dumping their killers aka garbage on us.** #StopTheInvasion #DeportThemAll #NoAmnesty #BuildTheWall

Figure 16: Disagreement concerning the Property. Collective labeled by both annotators is shown in red. Property labeled by one annotator is bolded while the one labeled by the other annotator is underlined

such kind. Example 17 shows how both annotators agree on how to split the text but disagree on which part is the Justification and which is the Conclusion. To improve the annotation process, the guidelines should emphasize that the main standpoint of the tweet should be identified before labeling the Justification. Example 2 shows disagreement about labeling the pivot. In this case, each annotator found a different Pivot that could be considered correct. The annotation guidelines enforce each annotator to label only one Pivot but there are examples, like the one mentioned above, where multiple Pivots could be found. This indicates that there could be an opportunity of improving the system if we enforce annotators to label all possible Pivots. Example 18 shows a disagreement on the Type of a Justification. The premise has declarative sentences with informative content (like "It is the third anniversary of her death") mixed with mandates or actions that must be followed ("Remember Kate Steinle today" and "We must not forget"). Depending on the part of the sentence

G Analysis of differences between automatic classifications and ground truth

We analyze the errors made by automatic classifiers when recognizing argumentative components,

@user @user you come with the usual lies an insults.
Fact is that mass immigration into Ireland has been going on for decades, most illegal and from other EU countries, still trans-formative. All the people seeking asylum

@user @user you come with the usual lies an insults. **Fact is that mass immigration into Ireland has been going on for decades, most illegal and from other EU countries, still trans-formative. All the people seeking asylum**

Figure 17: Disageement between annotators concerning Justification and Conclusion. Justification is bolded while Conclusion is underlined. While both annotators split the argument in the same fashion, they disagree on which part is the justification and which is the Conclusion

Remember Kate Steinle today. It is the third anniversary of her death We must not forget.
#KateSteinle#IllegalAliens
#OpenBorders#BuildThatWall
#MondayMorning#ImmigrationReform
#ImmigrationIsAWeapon

Figure 18: Example of disagreement concerning type of premise. Justification (bolded) was labeled as Fact by one annotator and as Policy by another

trying to determine possibilities of improvement either in the annotation process or in the settings of the task of automatic recognition.

Example 19 show an example of a non-argumentative tweet that was classified as argumentative by the automatic predictor trained as described in 5.1. The tweet has several hashtags calling for actions, but there is no explicit intention of using any of them as a justification of the others. The tweet refers to a mother who supposedly needs prayers, indicating that the author is aware of a context that is missing for us.

Example 20 shows a prediction done by a model trained following the settings described in 5.1. Here, the model correctly identifies a Collective mentioned in a xenophobe tweet, but there is no explicit Property assigned to them and because of this, it shouldn't have been labeled. Though this model was sometimes able to distinguish when the Collective should have been labeled or not, we found this error to be very frequent in experiments done with these settings. This led us to propose the experiment described in **??** separating the problem in two: first identifying if there is a pair of Collective and Property to label and then finding them on the tweet. When scoping the problem to find a Collective in a tweet that we know it is present, most errors produced by the automatic classifiers are discrepancies on the amount of words used to refer to the collective (like in example 22) or whenever the tweet mentions multiple collectives besides the target of the hate message (like example 21). We think that the first case reveals an opportunity for improvement on the annotation process, where sometimes a collective might have been labeled using one word and other times using many.

Example 23 show an incorrect prediction on the Property done by a model trained following the experiment described in 5.1. Although human trafficking could be considered as a negative consequence, the tweet does not explicitly associate it to a particular Collective. These models tend to identify phrases with negative connotations as Properties, disregarding if they are associated with the target group. This problem arises independently of the presence of a real Property and usually all words or phrases that could be considered as "negatives" are labeled by automatic predictors. Another error that automatic models are prone to are labeling bigger or smaller portions of text. Example 24 shows a prediction made by a model trained

as described in section **??**. The model correctly identified "illegally invade the U.S." as part of the Property, but missed the rest.

Regarding Pivots, we found that a common problem derivates from the incapacity of the models to jointly learn to find the pivot and the separation of the tweet into premises. Example 4 shows predictions made by a model trained following the settings described in 5.1 that found two words in different parts of the tweet that are directly related, but that are both within the justification, so they are not really a pivot between premises. A new setting for experimentation could provide the model with the information of where are the Justification and the Conclusion, and enforce to predict exactly one phrase within each of them. Another error found when predicting pivots comes from where multiple valid Pivots can be found within the premises. Example 27 shows prediction of a model also trained as described in 5.1 that found two valid Pivots: "Salvini-Salvini" and "invade-invasion". Each one of them could be considered a valid Pivot, though the only one that was labeled by the human annotator was "Salvini-Salvini". This phenomenon is related and could be considered as a consequence of the disagreement between annotators shown in the example 2. In order to avoid this kind of error, annotators should be instructed to label all the possible Pivots if there were more than one.

For Premises and Conclusions, we found also several cases where the model correctly divided the tweet in two premises but failed to assign the kind of the premise: if it was a Justification or a Conclusion. Example 34 shows a prediction done by a model trained to jointly predict both Justification and Conclusion at the same time, as explained in 5.2. Here, the model correctly identifies both parts of the argument but fails to correctly assign the Justification and Conclusion in itself. It is interesting to note that models predicting a single component as described in 5.1 do the same mistake when predicting Justification and Conclusion for this same example. This correlates with similar discrepancies between annotators shown in example 17.

For the Types of premises, models trained following the settings described in 5.1 usually fail to predict the minority classes ('Value' for Conclusions and 'Value' and 'Policy' on Justifications). On the contrary, performance on these classes improves when models are trained following the settings described in 5.3. We found that using both kind of premises for training instead of just one no only increases the amount of examples but also leverages the distribution among classes, which leds to a significant boost in performance, as shown in table 3. Example 35 shows a Justification predicted as Policy by a model trained using only justifications and then correctly predicted as Value by a model trained using both Justifications and Conclusions.

Video: (part 1) London #BNP a frame trailer with patriotic sound system on the road in and around our capital city "say no to **immigration**" #Brexit #Immigration #ImmigrationBan #London #England #BrexitBorder #Brexiteer #Brexiteers #BrexitGoodNews #BrexitChaos

Figure 20: Example of prediction of Collective from experiment described in 5.1. Though the model finds a mention of a Collective that seems to be accurate, there is no explicit Property associated so it shoudn't have been labeled

At this time, w-organized crime/returning **jihadists** it's a matter of national security. #Italy #Salvini must ignore international social engineers/cultural **marxists** #V4 Itali Kurz others must challenge empty threats from un-eu **migration** pimps. What can they really do about it?

Figure 21: Model predicting only on tweets that have a Collective, besides correctly finding 'immigration', also labeled 'jihadists' and 'marxists', which are being used as properties for either the target collective or other groups (like 'international social engineers')

#Prayers for this mother #NoIllegals #SendThemAllBack w/ their families #NoDACA #BuildTheWallNow

Figure 19: Example of Non-Argumentative tweet incorrectly labeled as argumentative by automatic model. The tweet refers to a context that is missing on the text

Chain migration imported 120K foreign nationals from terrorist-funding countries since 2005 - breitbart @user @user #EndChainMigration #EndDACA #NoAmesty #EndBirthrightCitizenshipForIllegalAliens #BuildTheWall #KeepAmericaSafe

Figure 22: Example of prediction of Collective from experiment described in 5.1. The prediction seems to be accurate, but it included the word "chain" associated with migration. Differences like this arise whenever there are frequent phrases like "Chain Migration" or "Illegal immigrants"

@user the disgrace is the illegal parent who brought their kids on their cirme spree to **illegally invade the U.S.** so taxpayers pay for their kids education wic and medicaid. We don't owe illegals our tax dollars #SendThemBack #WalkAway #Trump #MAGA #RedNationRising

Figure 24: Example of prediction of Property from experiment described in **??**. Real Property is undelined while prediction is bolded. The model predicted just a portion of the real Property and left most of it unlabeled

Please dont call it ""rescue"" - it's **human trafficking** #PortsClosed #SendThemBack #BenefitSeekers

Figure 23: Example of prediction of Property. Predicted Property is bolded. There was no real property labeled in this example.

Americans agree with @user on immigration. We can not afford to give welfare to illegals while U.S. citizens are homeless **#VoteDemsOut #FamiliesBelongTogheterMarch**

Figure 25: Example of prediction of Conclusion. Real conclusion is underlined while predicted is bolded. Here, the two parts of the argument were correctly identified but predictor chose the conclusion incorrectly

Americans agree with @user on immigration. **We can not afford to give welfare to illegals while U.S. citizens are homeless** #VoteDemsOut #FamiliesBelongTogheterMarch

Figure 26: Example of prediction of Justification. Real justification is underlined, while predicted is bolded. Again, the two parts of the argument were correctly identified but predictor chose the incorrect half

Pressure on **Spain's** maritime border: Boatloads of #Illegal #Migrants Storm **Spanish** Tourist Beaches & Scatter #StopTheInvasion #Unregistered #UnVetted

Figure 27: Example of pivot predicted by model trained as described in section 5.1. Justification is in blue, while conclusion is in red. Although the words selected establish a relation between themselves, they are both part of the justification, so they are not really a pivot between both premises

Rich African Countries don't take in African Migrants. Rich muslim countries don't take in muslim migrants. Rich latin american countries don't take it latin migrants. **But white countries are supposed to acept them??**

Figure 28: The conclusion (bolded) was predicted as Fact though it is a Policy

Angry that UN @user does its job and checks Lebanon isn't coercing Syrian refugees into returning home, **Lebanon will stop giving residence permits to the agencys international staff**

Figure 29: This conclusion was predicted as Policy though it is a Fact

@user Amen: **See 'Canada in Decay' by Ricardo Duchesne for the similar reality of Canada.** We are not nations of immigrants.

Figure 30: The justification (bolded) was predicted as Fact though it is a Policy

Good news. We are against illegal immigrants

Figure 31: The justification (bolded) was predicted as Fact though it is a Value

@user Immigration in a picture :link: **Some basic truths: Access to White people is not a human right.**

Figure 32: Example of prediction of Justification and Conclusion. Predicted Conclusion is shown in blue while the real one is bolded. Predicted Justification is shown in red while the real one is underlined. Models were able to correctly divide the tweet in two premises but failed to correctly recognize Justification and Conclusion

@user Immigration in a picture :link: **Some basic truths: Access to White people is not a human right.**

Figure 33: Example of prediction of Conclusion. Predicted Conclusion is underlined while the real one is bolded.

@user **Immigration in a picture** :link: Some basic truths: Access to White people is not a human right.

Figure 34: Example of prediction of Justification. Predicted Conclusion is underlined while the real one is bolded.

I do not want those vile thugs in our country

Figure 35: Justification labeled as Value by human annotator. This premise was predicted as Policy by a model trained following the settings described in 5.1 and was correctly identified as Value by a model trained as described in 5.3

H Argument annotated social media corpora

There exist several datasets with argument annotations, but only a few of them annotate arguments on Twitter. DART relies on Argumentation Theory (Rahwan and Simari, 2009) finding relationships between tweets as a single unit, considered to be arguments within an Abstract Argumentation Framework (Dung, 1995). Tweets are considered as argumentative if they express opinion or claims showing stance about a particular topic, and then they are defined according to how they interact with other tweet-arguments. The work of Dusmanu et al. (2017) extends the *#Grexit* subset of DART (987 tweets) with another 900 labeled for argument detection and adds labels for factual arguments recognition and source identification. However, abstract frameworks do not consider the inner structure of arguments and are not useful in providing an argumentative analysis in the context of a single tweet.

Schaefer and Stede (2020) labeled 300 replies to context tweets about Climate Change in German language with claims and evidence to support the claims. This was later expanded to 1200 tweets and the annotation scheme was refined to focus on particular argument properties (Schaefer and Stede, 2022). This is the only work, to our knowledge, where spans are annotated within a tweet, but it is not a hate dataset and does not have domain specific information.

Finally, Bhatti et al. (2021) created a dataset of 24100 tweets searching two hashtags supporting and attacking Planned Parenthood. The whole tweet is assigned a single label (i.e., support or not the claim) and there is no argumentative structure segmentation within, so it is impossible to differentiate aspects of argumentative information.

Author Index