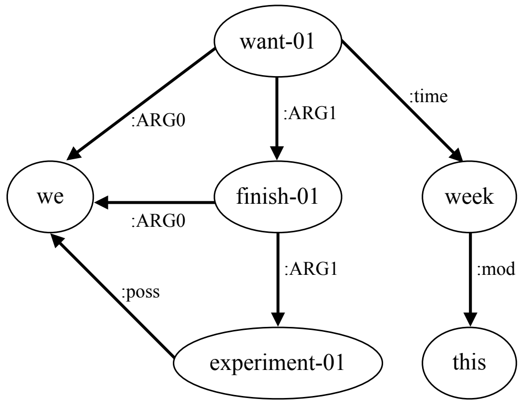
Ambiguity and Disagreement in Abstract Meaning Representation

Shira Wein
Amherst College
swein@amherst.edu

Abstract

Abstract Meaning Representation (AMR) is a graph-based semantic formalism which has been incorporated into a number of downstream tasks related to natural language understanding. Recent work has highlighted the key, yet often ignored, role of ambiguity and implicit information in natural language understanding. As such, in order to effectively leverage AMR in downstream applications, it is imperative to understand to what extent and in what ways ambiguity affects AMR graphs and causes disagreement in AMR annotation. In this work, we examine the role of ambiguity in AMR graph structure by employing a taxonomy of ambiguity types and producing AMRs affected by each type. Additionally, we investigate how various AMR parsers handle the presence of ambiguity in sentences. Finally, we quantify the impact of ambiguity on AMR using disambiguating paraphrases at a larger scale, and compare this to the measurable impact of ambiguity in vector semantics.




```
(w1 / want-01
    :ARG0 (w2 / we)
    :ARG1 (f / finish-01
        :ARG1 (e / experiment-01
            :poss w2)
        :ARG0 w2)
    :time (w3 / week
        :mod (t / this)))
```

Figure 1: The AMR annotation for the sentence "we want to finish our experiments this week," as a graph (top) and as a string in PENMAN notation (bottom).

1 Introduction

Abstract Meaning Representation (AMR; Banarescu et al., 2013) is a semantic representation which formally encodes the meaning of a sentence or phrase in the form of a rooted, directed graph. Figure 1 shows an example AMR graph of a sentence, in both PENMAN (string-based) and graph-based form. AMR has recently been leveraged for a range of downstream tasks (Wein and Opitz, 2024). While progress has been made on incorporating AMR for engineering purposes, there has not yet been consideration for how ambiguity affects AMR graph structure.

Ambiguity is a key factor in understanding the meaning of a sentence (Zipf, 1949; Piantadosi et al., 2012) and is also a pain point for current NLP systems (Yuan et al., 2023; Liu et al., 2023), making this an important consideration for formal semantic representations such as AMR. Further, ambiguity in the form of "differences in interpretation" is cited as one of the primary causes of disagreement in AMR annotation (§2.1). Therefore, if we want to effectively leverage AMR as a meaning representation for downstream tasks, it is important to investigate how ambiguity affects AMR given the critical role ambiguity plays in meaning.

In this work, we investigate the role of ambiguity on English AMR graph structure by (1) examining which *types* of ambiguity affect graph structure, (2) determining how three top-performing text-to-AMR parsers handle ambiguity in text, and (3) measuring the effect of ambiguity on AMR in comparison to vector semantics.

First (in §3), in order to assess which types of ambiguity affect AMR graph structure, we apply the ambiguity taxonomy from Li et al. (2024). Using the ambiguous sentences and their possible

Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation, pages 145–154
January 19, 2025.

interpretations provided from Li et al. (2024); Liu et al. (2023), we parse the sentences into AMR graphs for each interpretation, with the notion that if the ambiguous sentence breaks down into multiple AMRs (with the AMR graph being dependent upon interpretation), then the type of ambiguity present in the sentence affects AMR.

Second (in §4), we examine how ambiguity in text is handled by text-to-AMR parsers. Text-to-AMR parsing is the task of automatically converting a sentence or phrase into its corresponding AMR annotation. We elicit parses of the ambiguous sentences from three high-performing AMR parsers and assess whether they parse AMRs corresponding to the same or different interpretations.

Third (in §5), for a large set of disambiguating paraphrases of ambiguous sentences, we measure AMR graph overlap (via Smatch (Cai and Knight, 2013)) and BERTscores (Zhang et al., 2019) of the sentences in order to see a broader picture of the measurable effect of ambiguity on both forms of semantic representations.

2 Background

2.1 AMR Disagreement

Abstract Meaning Representation (AMR) is a semantic representation which reflects "who does what to whom," capturing the core concepts and relationships of elements of meaning (Banarescu et al., 2013). AMRs are rooted, directed graphs in which the nodes correspond with concepts in the sentence and edges indicate the relationships between those concepts; the root typically reflects the main action verb. AMR was originally designed for English but has since been extended to a number of other languages (Wein and Schneider, 2024). Annotation is fairly lightweight but still requires annotator training. Inter-annotator agreement (IAA) is often calculated using Smatch (Cai and Knight, 2013), a hill-climbing algorithm which measures graph overlap on a scale from 0 to 1; 1 indicates graph isomorphism and 0 indicates no shared graph attributes.

In existing AMR corpora, reported IAA has ranged from 0.71 to 0.89. Numerous causes have been cited as the reason for annotator disagreement. Persian AMR (Takhshid et al., 2022), Portuguese AMR (Sobrevilla Cabezudo and Pardo, 2019), Korean AMR (Choe et al., 2020), Spanish AMR (Wein et al., 2022), and Chinese AMR (Li et al., 2016) all cited different *interpretations of sentences* as being causes of different AMR graphs. Specific sources of difference included modality, conjunctive markers with multiple meanings, and verb sense labels. Thus, it is important to investigate how ambiguity, as it relates to different possible interpretations of sentences, quantitatively affects AMR annotation.

Multilingual issues, such as a lack of in-language frame sets or individual collocations not represented in the guidelines (Takhshid et al., 2022; Sobrevilla Cabezudo and Pardo, 2019; Choe et al., 2020), errors (Li et al., 2016; Sobrevilla Cabezudo and Pardo, 2019; Oral et al., 2024; Wein et al., 2022), and confusion with guidelines (Sobrevilla Cabezudo and Pardo, 2019; Choe et al., 2020; Wein et al., 2022) were also cited as causes of annotator disagreement. English AMR (Banarescu et al., 2013) did not describe causes of annotator disagreement.

2.2 Related Work on Ambiguity in Symbolic Representations

As we do for AMR in this work, prior work has considered the role of ambiguity in other symbolic representations. In particular, prior work investigated the impact of ambiguous input on semantic parsing with regard to synchronous context free grammars (Arthur et al., 2015), logical forms (Stengel-Eskin et al., 2023), and synactic parse trees (Church and Patil, 1982).

Dumitrache et al. (2019) produced a crowd-annotated FrameNet corpus which contains multiple annotations per frame and disagreement-based confidence scores, as opposed to the single most-chosen frame, in order to account for ambiguity in the text which would alter the frame annotation. Similarly, Vossen et al. (2018) created a data-to-text corpus with incorporated referential ambiguity.

On the other hand, it is also possible to *address* the presence of ambiguity using formal representations. Koller et al. (2008) addressed scope ambiguity by computing the most likely reading using a regular tree grammar and Duan et al. (2016) used CCG to produce disambiguating paraphrases.

3 Effect of Each Type of Ambiguity on AMR Graphs

In this section, we investigate which types of ambiguity have an effect on AMR via analysis on a small dataset.

3.1 Data and Approach to AMR Parsing

We extract the ambiguous sentences and their individual interpretations from Li et al. (2024), which contains sentences collected from various sources plus newly generated sentences, and an appendix with taxonomically annotated sentences from the AmbiEnt dataset (Liu et al., 2023). We use all of the sentences included in the work, which results in 25 sentences occupying 11 categories of ambiguity, with two or three sentences per category.

We produce AMRs for each interpretation by automatically parsing (1) the original ambiguous sentence, (2) the first interpretation, and (3) the second interpretation, using SPRING (Bevilacqua et al., 2021). We manually write the sentences corresponding to the two interpretations based on the descriptions of the source of ambiguity provided in Li et al. (2024). Then, we manually fix any errors in the automatically parsed AMRs and ensure that they do in fact represent the two distinct possible interpretations of the ambiguous sentence.

In producing and analyzing the AMRs, we determine whether, for each of the explored types of ambiguity, different AMR graph structures are necessary to reflect the individual interpretations. If different AMR graph structures result from each interpretation, this indicates that *the type of ambiguity has an effect on AMR graph structure*.

3.2 Results

The results for this experiment, with different AMRs being parsed for the individual interpretations indicating the ambiguity has an effect on AMR, can be found in Table 1.

Of the 11 types of ambiguity, four (syntactic, elliptical, idiomatic, and coreferential) have an effect on the AMR graph for all sentences in that category, five (collective, implicative, presuppositional, generic, and type/token) have no effect on AMR for any sentences in that category, and two (lexical and scopal) have mixed effects. All AMR graphs along with their interpretations and IDs (which ambiguity they contain) can be found in Appendix A.

Type of Ambiguity	Sent. 1	Sent. 2	Sent. 3
Lexical	✗	✓	
Syntactic	✓	✓	✓
Scopal	✗	✓	
Elliptical	✓	✓	✓
Collective	✗	✗	
Implicative	✗	✗	
Presuppositional	✗	✗	
Idiomatic	✓	✓	
Coreferential	✓	✓	
Generic	✗	✗	✗
Type/Token	✗	✗	

Table 1: For each type of ambiguity, shows which sentences have the same (X) versus different (check) AMRs for both interpretations, where having different AMRs indicates that the ambiguity does have an effect on AMR graph structure for that sentence.

Consistent effect. Syntactic ambiguity consistently has an effect on AMR graph structure because it changes argument placement. For example, in the case of "superfluous hair remover," for the interpretation *remover of superfluous hair remover*, superfluous modifies hair, whereas for the *hair remover which is superfluous* interpretation, superfluous modifies remover.

Elliptical and coreferential ambiguity consistently affect AMR graph structure because they dictate the content of the coreferent concept. For example, for elliptical ambiguity, "Peter walked his dog, and Dan did, too" could indicate that Dan walked either his own or Peter's dog, which is represented differently in AMR because the dog walked by Dan will either be possessed by Peter or Dan. For coreferential ambiguity, such as "Abby told Brittney that she upset Courtney" the `:ARG0` of *upset*, i.e. the actor doing the upsetting, will be either Abby or Brittney depending on the interpretation.

Idioms are incorporated in AMR as special frames. Therefore, whether the idiom is the intended meaning or the literal interpretation is the intended meaning will change whether the special frame is used (e.g. `(z1 / kick_bucket-05)` versus `(z1 / kick-01 :ARG1 (z2 / bucket))` for "kick the bucket").

Mixed effects. Lexical ambiguity sometimes has an effect on AMR graph structure depending on whether one interpretation receives special treatment in AMR. For example "bank" is not represented differently if it is a financial or river bank, but "speaker" could be represented as `(z1 / person :ARG0-of (z2 / speak-01))` or `(z1 / speaker)` if it is a person speaking or a loudspeaker, respectively. This is indicative of the fact that people are rooted by `person` in AMR, and other entities such as organizations would receive similar treatment, and thus would similarly induce an AMR divergence based on lexical ambiguity.

Scope is not represented in AMR and as a result, scopal ambiguity generally should not affect AMR graph structure. However, for the sentence "he wants to attend a school in New York," the emphasis could change based on the interpretation to have the school be an argument of `be-located-at-91` or have location be a modifier of the school.

No effect. The types of ambiguity which do not have an effect on AMR graph structure generally rely on commonsense knowledge or assumptions, which are not incorporated into AMR. For example, "the students wrote a paper" (which is affected by collective/distributive ambiguity) is represented the same in AMR whether the students wrote a paper together or individually.

Similarly, implicative, presuppositional, and generic/non-generic ambiguity all rely on assumptions about content not contained in the sentence, which therefore does not affect AMR structure, since implied content does not appear in an AMR annotation.

Type/token ambiguity can be closer to underspecification than outright ambiguity, as in the case of "you should visit Norway in the summer." This is represented the same in AMR whether it is interpreted as "you should visit Norway *this* summer" or "you should visit Norway during *a* summer," but they would differ if the text explicitly said "this/a summer."

4 AMR Parsers and Ambiguity

In this section we investigate how text-to-AMR parsers handle the presence of ambiguity in text, using the same data as in §3.

Methodology. For this experiment, we test how ambiguity affects the output of the SPRING (Bevilacqua et al., 2021), XFM-BART-large, and T5-based text-to-AMR parsers.[1]

Results. For all cases except for two where the ambiguity resulted in two different AMR structures (one for each interpretation), the three parsers produced graphs corresponding with the same interpretation.

The first case where the parsers produced different interpretations was due to an error made by the T5-based parser, which for the sentence "Calvin will honor his father and Otto will too" produced an AMR reflecting that Otto will honor himself. The SPRING parser also had difficulty with this sentence, as even when explicitly stating that Otto will honor Otto's father, the output still indicated that Otto too will honor Calvin's father.

[1]On the AMR 3.0 dataset (Knight et al., 2020), XFM-BART-large and SPRING both achieve Smatch scores of 0.84, while the T5-based parser achieves a Smatch score of 0.82. We run all three parsers through the `amrlib` package.

The next case of different interpretation production amongst the parsers was for the sentence "My roommate and I met the lawyer for coffee, but she became ill and had to leave." The SPRING and XFM-BART-large parsers both produced AMRs indicating that the roommate became ill, while the T5-based parser produced an AMR indicating that the lawyer became ill.

In general, the parsers accommodated the ambiguity by outputting one acceptable interpretation, though ambiguity is a possible cause of parser disagreement and/or error, as demonstrated here by the two cases where parser disagreement/error did occur. Still, the quite consistent parsing of ambiguous sentences into the same meaning suggests that there is perhaps a "default" or more likely meaning for the ambiguous sentence from the perspective of text-to-AMR parsers, which is the interpretation reflected in the automatic AMR parse.

5 Overall Effect of Ambiguity on AMR Similarity

Now, we measure the quantitative impact of ambiguity on AMR by parsing a large set of disambiguating paraphrases and comparing the Smatch scores of the AMRs against their corresponding sentences' BERTscore values.

5.1 Approach to Calculating Overall Effect

For this analysis, we use the linguistically annotated sentences from the AmbiEnt dataset (Liu et al., 2023), a natural language inference test set of ambiguous sentences; the premises form disambiguating paraphrases of the original sentence (if the original sentence is ambiguous, which not all are). We use only the sentences which have disambiguations and pair their premises, resulting in 919 disambiguated sentence pairs. Then, we use the SPRING parser to produce the 1,838 AMRs of these sentences. This allows us to then calculate Smatch similarity between each of the different interpretations of the original sentence.

5.2 Results

Overall, we find that the Smatch similarity between the different AMRs of the interpretations is 0.83. The pairwise scores range from 0.17 to 1.0, with 387 of the items having a Smatch score of 1.0.[2] A number of the especially low scores (including

[2]The variance was 0.03 and the median was 0.88.

the 0.17 case) were caused by multi-sentence differences (i.e. whether one or both AMRs were rooted by `multi-sentence`), which is not a divergence in AMR structure that conveys a difference in meaning.

One example of an AMR pair with a low Smatch score (0.67) is for the following sentence pair: "the vote was close because many people were unsure of their vote" and "the vote was close because many people abstained due to indecision." The AMRs diverge because the argument of their shared `cause-01` root either reflects the abstention or the indecision, making the AMRs quite different. However, these sentences have a BERTscore of 0.78.

The average `bert-base-uncased` BERTscore of the sentences is 0.91, noticably higher than the 0.83 average Smatch score. Thus, in line with prior work (Leung et al., 2022; Wein et al., 2023; Opitz et al., 2023), we find that AMR metrics are *even more sensitive* to finer-grained differences in meaning than embedding-based semantics. While AMR reflects finer-grained differences in meaning, in particular with respect to predicate-argument structure, BERTscore and similar vector-based representations of meaning are less sensitive to these nuances of meaning. Therefore, taking ambiguity into account is even more important when working with AMR than with vector-based models.

6 Conclusion & Future Work

In this work, we investigated the effect of ambiguity on AMR by determining whether different interpretations of ambiguous sentences result in different AMR graphs. Ultimately, we find that syntactic, elliptical, idiomatic, and coreferential ambiguity consistently affect AMR graph structure, and lexical and scopal ambiguity can also affect AMRs depending on the specific sentence. We manually examine a small amount of sentences, which makes it possible that the other types of ambiguity have edge cases which may impact an AMR; still, in our sample and as a general rule, they do not have an impact and we reason through why in §3.

The results of our experiments indicate that ambiguity not only has an effect on AMR, but likely has an even greater effect on AMR than on embedding-based semantics. Therefore, when calculating IAA, it is important for AMR dataset curators to verify to what extent ambiguity is present in the data. Further, our results suggest that when ambiguity can be resolved by presenting additional context to annotators, the extra-sentential context should be provided.

Finally, these findings motivate future work providing AMR datasets with multiple acceptable AMRs per sentence, following Dumitrache et al. (2019). Similarly to how Huang et al. (2023) created a dataset where each AMR led to the production of multiple paraphrased sentences, our work suggests the utility of datasets containing multiple AMRs per sentence (of which ours is the first).[3]

Limitations

Our qualitative analysis, though supplemented with a larger-scale quantitative analysis, is limited to the sentences contained in Li et al. (2024) and is small-scale. However, we contextualize the observed effects within the AMR schema to further unpack which types of ambiguity affect AMR generally.

While we leverage a thorough taxonomy of ambiguity for NLP, it is possible that there are other kinds of ambiguity which may be relevant. Also, this investigation is for English data, so it is yet to be seen how this would extend to other languages.

Regarding additional future work, if in the future the AmbiEnt dataset (Liu et al., 2023) is annotated with the types of ambiguity presented in Li et al. (2024), we could also quantify the effect of each category (rather than overall effect).

References

Philip Arthur, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Semantic parsing of ambiguous input through paraphrasing and verification. *Transactions of the Association for Computational Linguistics*, 3:571–584.

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 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 AAAI*.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association*

[3]The full set of 1838 automatically parsed AMRs from §5 are available at https://github.com/shirawein/amr-ambiguity/.

for Computational Linguistics (Volume 2: Short Papers), pages 748–752, Sofia, Bulgaria. 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.

Kenneth Church and Ramesh Patil. 1982. Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics*, 8(3-4):139–149.

Manjuan Duan, Ethan Hill, and Michael White. 2016. Generating disambiguating paraphrases for structurally ambiguous sentences. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 160–170, Berlin, Germany. Association for Computational Linguistics.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2019. A crowdsourced frame disambiguation corpus with ambiguity. 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 2164–2170, Minneapolis, Minnesota. Association for Computational Linguistics.

Kuan-Hao Huang, Varun Iyer, I-Hung Hsu, Anoop Kumar, Kai-Wei Chang, and Aram Galstyan. 2023. ParaAMR: A large-scale syntactically diverse paraphrase dataset by AMR back-translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8047–8061, Toronto, Canada. Association for Computational Linguistics.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, et al. 2020. Abstract Meaning Representation (AMR) Annotation Release 3.0. Technical Report LDC2020T02, Linguistic Data Consortium, Philadelphia, PA.

Alexander Koller, Michaela Regneri, and Stefan Thater. 2008. Regular tree grammars as a formalism for scope underspecification. In *Proceedings of ACL-08: HLT*, pages 218–226, Columbus, Ohio. Association for Computational Linguistics.

Wai Ching Leung, Shira Wein, and Nathan Schneider. 2022. Semantic similarity as a window into vector- and graph-based metrics. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 106–115, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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.

Margaret Y Li, Alisa Liu, Zhaofeng Wu, and Noah A Smith. 2024. A taxonomy of ambiguity types for nlp. *arXiv preprint arXiv:2403.14072*.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Juri Opitz, Shira Wein, Julius Steen, Anette Frank, and Nathan Schneider. 2023. AMR4NLI: Interpretable and robust NLI measures from semantic graphs. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 275–283, Nancy, France. Association for Computational Linguistics.

Elif Oral, Ali Acar, and Gülşen Eryiğit. 2024. Abstract meaning representation of turkish. *Natural Language Engineering*, 30(1):171–200.

Steven T Piantadosi, Harry Tily, and Edward Gibson. 2012. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

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.

Elias Stengel-Eskin, Kyle Rawlins, and Benjamin Van Durme. 2023. Zero and few-shot semantic parsing with ambiguous inputs. In *The Twelfth International Conference on Learning Representations*.

Reza Takhshid, Razieh Shojaei, Zahra Azin, and Mohammad Bahrani. 2022. Persian abstract meaning representation. *arXiv preprint arXiv:2205.07712*.

Piek Vossen, Filip Ilievski, Marten Postma, and Roxane Segers. 2018. Don't annotate, but validate: a data-to-text method for capturing event data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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.

Shira Wein and Juri Opitz. 2024. A survey of AMR applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6856–6875, Miami, Florida, USA. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2024. Assessing the cross-linguistic utility of abstract meaning representation. *Computational Linguistics*, pages 1–55.

Shira Wein, Zhuxin Wang, and Nathan Schneider. 2023. Measuring fine-grained semantic equivalence with Abstract Meaning Representation. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 144–154, Nancy, France. Association for Computational Linguistics.

Yuewei Yuan, Chaitanya Malaviya, and Mark Yatskar. 2023. AmbiCoref: Evaluating human and model sensitivity to ambiguous coreference. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1023–1030, Dubrovnik, Croatia. Association for Computational Linguistics.

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

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort: An introduction to human ecology*.

A All Categorized AMR Graphs

```
# ::snt We finally reached the bank. ::id
    lex_amb_1_interpret_1
# ::interpretations: We finally reached the
    river bank.; We finally reached the
    financial bank.
(z1 / reach-01
     :ARG0 (z2 / we)
     :ARG1 (z3 / bank)
     :time (z4 / final))

# ::snt The speaker is at the front of the
    room. ::id lex_amb_2_interpret_1
# ::interpretations: The person who is
    speaking is at the front of the room.
(z1 / person
     :ARG0-of (z2 / speak-01)
     :location (z3 / front
          :part-of (z4 / room)))

# ::snt The speaker is at the front of the
    room. ::id lex_amb_2_interpret_2
# ::interpretations: The loudspeaker is at
    the front of the room.
(z1 / speaker
     :location (z2 / front
          :part-of (z3 / room)))

# ::snt superfluous hair remover ::id
    synt_amb_1_interpret_1
# ::interpretations: remover of superfluous
    hair
(z1 / remove-01
     :ARG1 (z2 / hair
          :mod (z3 / superfluous)))

# ::snt superfluous hair remover ::id
    synt_amb_1_interpret_2
# ::interpretations: hair remover which is
    superfluous
(z1 / remove-01
     :ARG1 (z2 / hair)
     :mod (z3 / superfluous))

# ::snt The girl hit the boy with the book.
    ::id synt_amb_2_interpret_1
# ::interpretations: With the book, the girl
     hit the boy.
(z1 / hit-01
     :ARG0 (z2 / girl)
     :ARG1 (z3 / boy)
     :ARG2 (z4 / book))

# ::snt The girl hit the boy with the book.
    ::id synt_amb_2_interpret_2
# ::interpretations: The girl hit the boy
    who had the book.
(z1 / hit-01
     :ARG0 (z2 / girl)
     :ARG1 (z3 / boy
          :ARG0-of (z4 / have-03
               :ARG1 (z5 / book))))

# ::snt He's drawing all over the bus with
    graffiti. ::id synt_amb_3_interpret_1
# ::interpretations: He is drawing graffiti
    on the surface of the bus.
(z1 / draw-01
     :ARG0 (z2 / he)
     :ARG1 (z3 / graffiti)
     :location (z4 / bus)
     :extent (z5 / all-over))

# ::snt He's drawing all over the bus with
    graffiti. ::id synt_amb_3_interpret_2
# ::interpretations He is on the bus,
    drawing graffiti.
(z1 / bus
     :location-of (z2 / he
          :ARG0-of (z3 / draw-01
               :ARG1 (z4 / graffiti))))

# ::snt Every student read two poems. ::id
    scop_amb_1_interpret_1
# ::interpretations Every student read two (
    possibly different) poems.; Two poems
    were read by every student (same poems).
(z1 / read-01
     :ARG0 (z2 / person
          :ARG0-of (z3 / study-01)
          :mod (z4 / every))
     :ARG1 (z5 / poem
          :quant 2))

# ::snt He wants to attend a school in New
    York. ::id scop_amb_1_interpret_1
# ::interpretations There is a school in New
     York that he wants to attend.
(z1 / want-01
     :ARG0 (z2 / he)
     :ARG1 (z3 / attend-01
          :ARG0 z2
          :ARG1 (z4 / school
```

```
                :location (z5 / city
                      :name (z6 / name
                            :op1 "New"
                            :op2 "York")))))

# ::snt He wants to attend a school in New
     York. ::id scop_amb_1_interpret_1
# ::interpretations He wants to attend
     school in New York.
(z1 / want-01
      :ARG0 (z2 / he)
      :ARG1 (z3 / attend-01
            :ARG0 z2
            :ARG1 (z4 / be-located-at-91
                :ARG1 (z5 / school)
                :ARG2 (z6 / city
                      :name (z6 / name
                            :op1 "New"
                            :op2 "York")))))

# ::snt Peter walked his dog, and Dan did,
     too. ::id ellip_amb_1_interpret_1
# ::interpretations Peter and Dan walked
     Peter's dog.
(z1 / walk-01
      :ARG0 (z2 / and
            :op1 (z3 / person
                  :name (z4 / name
                        :op1 "Peter"))
            :op2 (z5 / person
                  :name (z6 / name
                        :op1 "Dan")))
      :ARG1 (z7 / dog
            :poss z3))

# ::snt Peter walked his dog, and Dan did,
     too. ::id ellip_amb_1_interpret_2
# ::interpretations Peter walked his dog,
     and Dan walked his own dog.
(z1 / and
      :op1 (z2 / walk-01
            :ARG0 (z3 / person
                  :name (z4 / name
                        :op1 "Peter"))
            :ARG1 (z5 / dog
                  :poss z3))
      :op2 (z6 / walk-01
            :ARG0 (z7 / person
                  :name (z8 / name
                        :op1 "Dan"))
            :ARG1 (z9 / dog
                  :poss z7)))

# ::snt Sam loves Jess more than Jason. ::id
      ellip_amb_2_interpret_1
# ::interpretations Sam loves Jess more than
      Sam loves Jason.
    (z1 / love-01
      :ARG0 (z2 / person
            :name (z3 / name
                  :op1 "Sam"))
      :ARG1 (z4 / person
            :name (z5 / name
                  :op1 "Jess"))
      :ARG1-of (z6 / have-degree-91
            :ARG3 (z7 / more)
            :ARG4 (z8 / love-01
                  :ARG0 z2
                  :ARG1 (z9 / person
                        :name (z10 / name
                              :op1 "Jason")))))

# ::snt Sam loves Jess more than Jason. ::id
      ellip_amb_2_interpret_2
# ::interpretations Sam loves Jess more than
      Jason loves Jess.
    (z1 / love-01
      :ARG0 (z2 / person
            :name (z3 / name
                  :op1 "Sam"))
      :ARG1 (z4 / person
            :name (z5 / name
                  :op1 "Jess"))
      :ARG1-of (z6 / have-degree-91
            :ARG3 (z7 / more)
            :ARG4 (z8 / love-01
                  :ARG0 (z9 / person
                        :name (z10 / name
                              :op1 "Jason"))
                  :ARG1 z4)))

# ::snt Calvin will honor his father and
     Otto will too. ::id
     ellip_amb_3_interpret_1
# ::interpretations Calvin and Otto will
     honor Calvin's father.
(z1 / and
      :op1 (z2 / honor-01
            :ARG0 (z3 / person
                  :name (z4 / name
                        :op1 "Calvin"))
            :ARG1 (z5 / person
                  :ARG0-of (z6 / have-rel-role
                      -91
                        :ARG1 z3
                        :ARG2 (z7 / father))))
      :op2 (z8 / honor-01
            :ARG0 (z9 / person
                  :name (z10 / name
                        :op1 "Otto"))
            :ARG1 z5
            :mod (z11 / too)))

# ::snt Calvin will honor his father and
     Otto will too. ::id
     ellip_amb_3_interpret_2
# ::interpretations Calvin will honor Calvin
     's father, and Otto will honor Otto's
     father.
(z1 / and
      :op1 (z2 / honor-01
            :ARG0 (z3 / person
                  :name (z4 / name
                        :op1 "Calvin"))
            :ARG1 (z5 / person
                  :ARG0-of (z6 / have-rel-role
                      -91
                        :ARG1 z3
                        :ARG2 (z7 / father))))
      :op2 (z8 / honor-01
            :ARG0 (z9 / person
                  :name (z10 / name
                        :op1 "Otto"))
            :ARG1 (z11 / person
                  :ARG0-of (z12 / have-rel-
                      role-91
                        :ARG1 z9
                        :ARG2 (z13 / father)))
            :mod (z14 / too)))
```

```
# ::snt The students wrote a paper. ::id
     coll_amb_1_interpret_1
# ::interpretations The students wrote a
     paper together.; Each student wrote a
     paper separately.
(z1 / write-01
      :ARG0 (z2 / person
            :ARG0-of (z3 / study-01))
      :ARG1 (z4 / paper))

# ::snt Jenny and Zoe solved the puzzle. ::
     id coll_amb_2_interpret_1
# ::interpretations Jenny and Zoe each
     solved the puzzle individually.; Jenny
     and Zoe solved the puzzle together.
(z1 / solve-01
      :ARG0 (z2 / and
            :op1 (z3 / person
                  :name (z4 / name
                        :op1 "Jenny"))
            :op2 (z5 / person
                  :name (z6 / name
                        :op1 "Zoe")))
      :ARG1 (z7 / puzzle-01))

# ::snt Some gems in this box are fake. ::id
      impl_amb_1_interpret_1
# ::interpretations Some, and not all, gems
     in this box are fake.; Some, and perhaps
      all, gems in this box are fake.
(z1 / fake-02
      :ARG1 (z2 / gem
            :quant (z3 / some)
            :location (z4 / box
                  :mod (z5 / this))))

# ::snt Carolyn had talked to two senators.
     ::id impl_amb_2_interpret_1
# ::interpretations Carolyn had talked to (
     exactly) two senators.; Carolyn had
     talked to (at least) two senators.
(z1 / talk-01
      :ARG0 (z2 / person
            :name (z3 / name
                  :op1 "Carolyn"))
      :ARG2 (z4 / person
            :quant 2
            :ARG0-of (z5 / have-org-role-91
                  :ARG1 (z6 / government-
                       organization
                        :name (z7 / name
                              :op1 "Senate"))))
                                  )


# ::snt Jane left early too. ::id
     pres_amb_1_interpret_1
# ::interpretations (e.g. Robert left early
     .) Jane left early too.; (e.g. Jane
     arrived early.) Jane left early too.
(z1 / leave-11
      :ARG0 (z2 / person
            :name (z3 / name
                  :op1 "Jane"))
      :time (z4 / early)
      :mod (z5 / too))


# ::snt The new software is also available
     in a Spanish-language version. ::id
     pres_amb_2_interpret_1
# ::interpretations The new software is also
      available in a Spanish-language version
      (in addition to older software).; The
     new software is also available in a
     Spanish-language version (in addition to
      other languages).
(z1 / available-02
      :ARG2 (z2 / software
            :ARG1-of (z3 / new-01))
      :mod (z4 / also)
      :manner (z5 / version
            :mod (z6 / language
                  :name (z7 / name
                        :op1 "Spanish"))))

# ::snt kick the bucket ::id
     idiom_amb_1_interpret_1
# ::interpretations die
(z1 / kick_bucket-05)

# ::snt kick the bucket ::id
     idiom_amb_1_interpret_2
# ::interpretations hit a bucket with one's
     foot
(z1 / kick-01
      :ARG1 (z2 / bucket))

# ::snt He didn't see the big picture. ::id
     idiom_amb_2_interpret_1
# ::interpretations He didn't see the
     physical big picture.
(z1 / see-01
      :polarity -
      :ARG0 (z2 / he)
      :ARG1 (z3 / picture
            :mod (z4 / big-01)))


# ::snt He didn't see the big picture. ::id
     idiom_amb_2_interpret_2
# ::interpretations He didn't see the
     metaphorical big picture.
(z1 / see-01
      :polarity -
      :ARG0 (z2 / he)
      :ARG1 (z3 / big-picture-01))


# ::snt Abby told Brittney that she upset
     Courtney. ::id coref_amb_1_interpret_1
# ::interpretations Abby told Brittney that
     Abby upset Courtney.
(z1 / tell-01
      :ARG0 (z2 / person
            :name (z3 / name
                  :op1 "Abby"))
      :ARG1 (z4 / upset-01
            :ARG0 z2
            :ARG1 (z5 / person
                  :name (z6 / name
                        :op1 "Courtney")))
      :ARG2 (z7 / person
            :name (z8 / name
                  :op1 "Brittney")))

# ::snt Abby told Brittney that she upset
     Courtney. ::id coref_amb_1_interpret_2
# ::interpretations Abby told Brittney that
     Brittney upset Courtney.
```

```
(z1 / tell-01
      :ARG0 (z2 / person
            :name (z3 / name
                  :op1 "Abby"))
      :ARG1 (z4 / upset-01
            :ARG0 (z5 / person
                  :name (z6 / name
                        :op1 "Brittney"))
            :ARG1 (z7 / Courtney))
      :ARG2 z5)
```

```
# ::snt My roommate and I met the lawyer for
      coffee, but she became ill and had to
     leave. ::id coref_amb_2_interpret_1
My roommate and I met the lawyer for coffee,
      but the lawyer became ill and had to
     leave.
(z1 / meet-02
      :ARG0 (z2 / and
            :op1 (z3 / person
                  :ARG0-of (z4 / have-rel-role
                       -91
                        :ARG1 (z5 / i)
                        :ARG2 (z6 / roommate)))
            :op2 z5)
      :ARG1 (z7 / lawyer)
      :purpose (z8 / coffee)
      :concession-of (z9 / and
            :op1 (z10 / become-01
                  :ARG1 z7
                  :ARG2 (z11 / ill-01
                        :ARG1 z7))
            :op2 (z12 / obligate-01
                  :ARG1 z7
                  :ARG2 (z13 / leave-11
                        :ARG0 z7))))
```

```
# ::snt My roommate and I met the lawyer for
      coffee, but she became ill and had to
     leave. ::id coref_amb_2_interpret_2
# ::interpretations My roommate and I met
     the lawyer for coffee, but my roommate
     became ill and had to leave.
(z1 / meet-02
      :ARG0 (z2 / and
            :op1 (z3 / person
                  :ARG0-of (z4 / have-rel-role
                       -91
                        :ARG1 (z5 / i)
                        :ARG2 (z6 / roommate)))
            :op2 z5)
      :ARG1 (z7 / lawyer)
      :purpose (z8 / coffee)
      :concession-of (z9 / and
            :op1 (z10 / become-01
                  :ARG1 z3
                  :ARG2 (z11 / ill-01
                        :ARG1 z3))
            :op2 (z12 / obligate-01
                  :ARG1 z3
                  :ARG2 (z13 / leave-11
                        :ARG0 z3))))
```

```
# ::snt dinosaurs ate kelp ::id
     gen_amb_1_interpret_1
# ::interpretations In general, dinosaurs
     ate kelp.; On one occasion, some
     dinosaurs ate kelp.
(z1 / eat-01
      :ARG0 (z2 / dinosaur)
      :ARG1 (z3 / kelp))
```

```
# ::snt John ate breakfast with a gold fork.
      ::id gen_amb_2_interpret_1
# ::interpretations John generally ate
     breakfast with a gold fork.; During one
     breakfast, John ate with a gold fork.
(z1 / eat-01
      :ARG0 (z2 / person
            :name (z3 / name
                  :op1 "John"))
      :ARG1 (z4 / breakfast)
      :instrument (z5 / fork
            :consist-of (z6 / gold)))
```

```
# ::snt If an athlete uses a banned
     substance, they will be disqualified
     from the competition. ::id
     gen_amb_3_interpret_1
# ::interpretations As a rule, if an athlete
      uses a banned substance, they will be
     disqualified from the competition.; If
     the referenced athlete uses a banned
     substance, they will be disqualified
     from the competition.
(z1 / disqualify-01
      :ARG1 (z2 / athlete)
      :ARG2 (z3 / compete-01
            :ARG0 z2)
      :condition (z4 / use-01
            :ARG0 z2
            :ARG1 (z5 / substance
                  :ARG1-of (z6 / ban-01))))
```

```
# ::snt I paid for the same car. ::id
     type_amb_1_interpret_1
# ::interpretations I paid for the same car
     as another person.; I paid for the same
     car twice.
(z1 / pay-01
      :ARG0 (z2 / i)
      :ARG3 (z3 / car
            :ARG1-of (z4 / same-01)))
```

```
# ::snt You should visit Norway in the
     summer. ::id type_amb_2_interpret_1
# ::interpretations You should visit Norway
     this summer.; You should visit Norway
     during a summer.
(z1 / visit-01
      :ARG0 (z2 / you)
      :ARG1 (z3 / country
            :name (z4 / name
                  :op1 "Norway"))
      :time (z5 / date-entity
            :season (z6 / summer))
      :ARG1-of (z7 / recommend-01))
```