Predicting Coreference in Abstract Meaning Representations

Tatiana Anikina and **Alexander Koller**
Dept. of Language Science and Technology
Saarland Informatics Campus
Saarland University

Michael Roth
Institute for NLP
University of Stuttgart

{tatianak|koller}@coli.uni-saarland.de
michael.roth@ims.uni-stuttgart.de

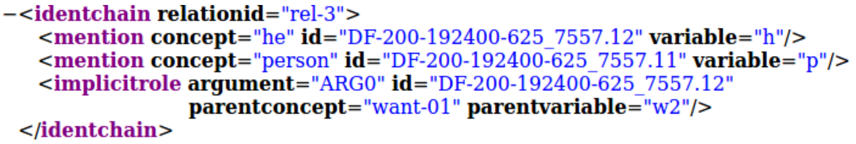
Abstract

This work addresses coreference resolution in Abstract Meaning Representation (AMR) graphs, a popular formalism for semantic parsing. We evaluate several current coreference resolution techniques on a recently published AMR coreference corpus, establishing baselines for future work. We also demonstrate that coreference resolution can improve the accuracy of a state-of-the-art semantic parser on this corpus.

1 Introduction

Abstract Meaning Representations (AMRs, Banarescu et al. (2013)) are a popular type of symbolic semantic representation for semantic parsing. AMRs are labeled directed graphs whose nodes represent entities, events, properties, and states; the edges represent semantic relations between the nodes. For instance, in the example AMRs of Fig. 2, the predicate node *c* describes a come-back relation between the ARG1 "I" and the ARG3 "this". AMR is designed to abstract over the way in which a certain piece of meaning was expressed in language; thus "the destruction of the room by the boy" and "the boy destroyed the room" are represented by the same graph. In the example AMR, the noun phrase "university offers" is decomposed into two nodes: the predicate node *o:offer-01* and the argument node *u:university*, describing an event in which the university offers something to "I".

An AMR graphbank annotates each sentence in the corpus with an AMR graph. Recently, O'Gorman et al. (2018) introduced the Multi-Sentence AMR (MS-AMR) corpus, which adds a layer of annotation on top of the AMR-2017 graphbank that represents coreference and implicit arguments beyond the sentence level. An example is shown in Fig. 1. Each <*identchain*> element collects mentions of the same entity; these mentions are not pieces of text as in other coreference annotation schemes, but nodes in the AMR graphs. The annotation also specifies what implicit roles of predicate nodes the entity fills.



```
-<identchain relationid="rel-3">
   <mention concept="he" id="DF-200-192400-625_7557.12" variable="h"/>
   <mention concept="person" id="DF-200-192400-625_7557.11" variable="p"/>
   <implicitrole argument="ARG0" id="DF-200-192400-625_7557.12"
                 parentconcept="want-01" parentvariable="w2"/>
 </identchain>
```

Figure 1: Coreference chain from MS-AMR.

In this paper, we make two contributions. First, we evaluate the performance of different coreference resolution tools on the MS-AMR annotations. We evaluate these on the token level (by projecting the coreference annotations from the nodes to the sentences) and on the node level (by projecting the tools' coreference predictions to the nodes of the graphs) and find that AllenNLP with SpanBERT embeddings (Joshi et al., 2020) generally performs best.

Second, we show for the first time how the output of a coreference system can be integrated into the predictions of a state-of-the-art AMR parser. We use the neural semantic parser of Lindemann et al.

Proceedings of the 3rd Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2020), pages 33–38,
Barcelona, Spain (online), December 12, 2020.

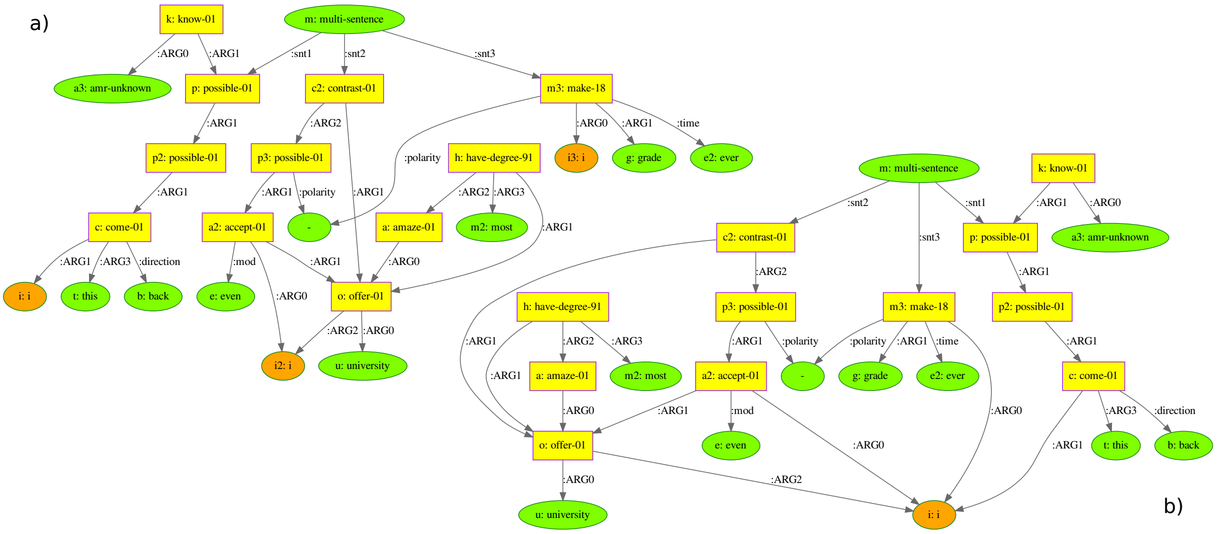


Figure 2: AMRs, (a) before and (b) after merge for *"Maybe **I** can come back from this, who knows. **I'**ve got the most amazing university offers, but **I** can't even accept them - **I'**ll never make the grades."*

(2019), which compositionally predicts a graph for the input sentence. We exploit this compositional structure to map coreferent input tokens to nodes in the predicted graph, and obtain an improvement of three points Smatch f-score over a coreference-unaware baseline.

2 Coreference in MS-AMR

Coreference resolution tools typically predict coreference between *tokens* in a text, but MS-AMR annotates coreference between *nodes* in the AMR graphs. To perform coreference resolution on MS-AMR, we therefore have to map between the token level and the node level. The MS-AMR corpus contains annotations which map between tokens and nodes, but this mapping is not always one-to-one. In the example shown in Fig. 2 (a), the two tokens "who knows" are aligned to the single node p. The nodes *a3:amr-unknown* and *h:have-degree-91* are left unaligned.

Furthermore, AMR graphs sometimes contain nodes that participate in the coreference chains but are not realized at the token level. For instance, in the sentence "speak to a doctor" the predicate *speak-01* has an ARG0 *you* which is a separate node in the graph even though it does not have any token alignment.

We evaluate coreference tools on MS-AMR in two different modes: *token-level*, where we project MS-AMR coreference annotations from nodes to tokens and compare them against the predicted token-level coreference annotations; and *node-level*, where we project token-level coreference predictions to MS-AMR nodes and compare them against the MS-AMR annotations. Because of the node–token mismatch explained above, we can project to the token level only coreference annotations between nodes that are aligned to tokens. We retained only coreference chains with at least two members. This reduces the 87 coreference chains between 425 mentions in the original MS-AMR test set to 69 coreference chains between 385 mentions. 35% of these chains consist only of two mentions although there are also some very long chains with more than 30 elements, mostly pronouns.

For the node-level evaluation and the Smatch-based evaluation (see below), we used the unmodified coreference annotations on the nodes.

3 Comparative Evaluation of Coreference Resolution Tools

We compared the output of the deterministic CoreNLP (Lee et al., 2013) and neural CoreNLP (Clark and Manning, 2016) coreference resolvers and tested two versions of the AllenNLP (Lee et al., 2017) coreference tool based on the GloVe (Pennington et al., 2014) and SpanBERT (Joshi et al., 2020) embeddings respectively. These tools were chosen due to their availability and their strong accuracy on English.

	MUC			B^3			$CEAF\ \phi_3$			$CEAF\ \phi_4$		
	P	R	F	P	R	F	P	R	F	P	R	F
AllenNLP (GloVe)	0.61	0.51	0.55	0.42	0.40	0.39	0.46	0.44	0.45	0.20	0.26	0.22
AllenNLP (SpanBERT)	0.60	0.56	0.58	0.44	0.44	**0.43**	0.50	0.49	**0.49**	0.24	0.28	**0.25**
CoreNLP (determin.)	0.45	0.50	0.47	0.35	0.35	0.32	0.35	0.39	0.37	0.14	0.27	0.18
CoreNLP (neural)	0.63	0.56	**0.59**	0.40	0.38	0.37	0.48	0.42	0.45	0.22	0.23	0.22

Table 1: Coreference evaluation at the token level for AllenNLP and CoreNLP.

	MUC			B^3			$CEAF\ \phi_3$			$CEAF\ \phi_4$		
	P	R	F	P	R	F	P	R	F	P	R	F
AllenNLP (GloVe)	0.62	0.37	0.45	0.31	0.29	0.28	0.51	0.34	0.39	0.32	0.23	0.26
AllenNLP (SpanBERT)	0.69	0.42	**0.50**	0.32	0.30	**0.30**	0.58	0.35	**0.43**	0.42	0.24	**0.30**
CoreNLP (determin.)	0.51	0.33	0.39	0.26	0.22	0.22	0.46	0.29	0.34	0.26	0.21	0.22
CoreNLP (neural)	0.64	0.37	0.46	0.31	0.28	0.27	0.51	0.33	0.39	0.30	0.22	0.24

Table 2: Coreference evaluation at the node level for AllenNLP and CoreNLP.

To evaluate the performance at the **token level**, the gold alignments were extracted and each coreference chain from the MS-AMR dataset was mapped to the corresponding span in the text. These annotations represent the gold standard to which we compared the system annotations. In order to annotate coreference chains, a separate text file was created for each document with the sentences representing the document AMRs. Then each document text was processed with different coreference resolution systems to generate the predictions. For the token-level evaluation we compared the system output directly to the coreferent tokens in the MS-AMR test set and for the node-level evaluation we first projected token annotations to the graph nodes using the gold alignments and then compared the node coreference chains.

Table 1 reports the token-level results on the MS-AMR test data using several metrics: MUC, B^3, mention-based $CEAF\ \phi_3$ and entity-based $CEAF\ \phi_4$. The evaluation shows that the neural version of CoreNLP achieves the best MUC f-score (0.59), followed by the SpanBERT version of AllenNLP (0.58). Neural CoreNLP and AllenNLP with GloVe show similar results in terms of B^3, $CEAF\ \phi_3$ and $CEAF\ \phi_4$. Overall, SpanBERT AllenNLP achieves the best performance and deterministic CoreNLP performs the worst in all metrics. The difference in scores is due to the way how metrics define the coreference: in terms of links (for MUC) or in terms of clusters (B^3 and $CEAF$).

Neural CoreNLP and AllenNLP are reasonable baselines for AMR coreference resolution, although the results seem to be worse than state-of-the-art performance reported on news and narrative texts. One problem might be that the MS-AMR corpus contains text snippets from blog data, including misspellings, jargon and incorrect grammar. Also the conversational style used in blogs poses challenges for the coreference tools since they do not distinguish between posts made by different authors.

The results of the **node-level** evaluation can be found in Table 2. They are based on mapping the predicted annotations to the nodes defined in the gold AMR graphs. The reason to perform both token and node-level evaluation is that coreference chains differ depending on whether their members are tokens or nodes. For example, there are four instances of token "I" in the text corresponding to the AMR in Fig. 2 (a) but the graph contains only three *i* nodes (*i*, *i2* and *i3*) because the predicates *a2:accept-01* and *o:offer-01* share the argument node *i2:i*. So, the number of mentions in each chain varies depending on whether the evaluation is done at the token or node level. Moreover, the node-level evaluation includes the full set of annotated nodes in the gold standard, not only those that can be aligned to tokens. At the node level, the SpanBERT version of AllenNLP achieves the best results in all metrics.

4 AMR parsing with coreference

Coreference is not an isolated task in MS-AMR parsing; in order to predict the gold annotations, coreference information needs to be incorporated into AMR graphs predicted by a semantic parser. We thus

	AMR parser			AMR parser + AllenNLP			AMR parser + oracle		
	P	R	F	P	R	F	P	R	F
macro-average:	0.57	0.52	0.54	0.61	0.54	0.57	0.63	0.56	0.59
micro-average:	0.57	0.50	0.53	0.60	0.53	0.56	0.63	0.55	0.58

Table 3: Smatch evaluation of document-level coreference annotations.

extended the AMR parser of Lindemann et al. (2019) with coreference information.

First, we prepared gold annotations at the document level. For this, we combined the individual AMRs from each document into a single graph to represent document-level annotations. The coreference chains were extracted from the gold annotations of the MS-AMR corpus, and coreferent nodes in the document graph were merged following the procedure described in (O'Gorman et al., 2018).

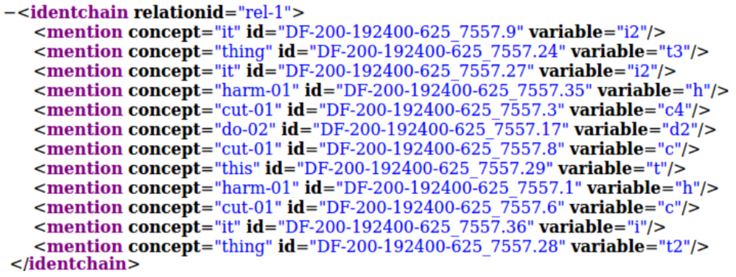
Second, we ran Lindemann's parser on each sentence separately and combined the predicted AMR graphs into a document-level graph. Then we ran SpanBERT AllenNLP (henceforth just AllenNLP) on each document text, and mapped each token-level prediction to the nodes the Lindemann parser predicted for those tokens. We collapsed the coreferent nodes by replacing all edges into a node for a coreferent token by edges into the first node of the coreference chain; see O'Gorman et al. (2018) for details. For example, in Fig. 2 (a) there are three coreferent nodes *i:i*, *i2:i* and *i3:i*. Since all three nodes represent the same entity the corresponding edges can be rearranged to point to the same node *i:i* as shown in Fig. 2 (b).

We evaluated the performance of Lindemann's parser, with and without the added coreference information, on the complete MS-AMR test data. To this end, we computed the Smatch score (Cai and Knight, 2013) for the predicted vs. gold document-level graphs. Table 3 shows the micro- and macro-average Smatch precision, recall and f-score for the documents from the test set. The left column indicates the scores obtained by comparing the gold AMRs with coreference to the ones generated by the parser without coreference. The middle column shows the scores for the gold MS-AMR graphs versus the parser output augmented with coreference predictions. The overall improvement in f-score is around three points Smatch f-score. The right column shows the scores obtained by augmenting Lindemann's parser output with the gold coreference chains extracted from the MS-AMR corpus (i.e. oracle predictions).

It is worth noting that the overall Smatch score is much lower than on other AMR graphbanks; for instance, Lindemann et al. (2019) report a Smatch f-score of 0.75 for their parser on the AMR-2017 test set. Even on the MS-AMR test corpus without coreference links (i.e. pure sentence-by-sentence parsing), the parser only gets a score of 0.61, indicating that this is a harder corpus than AMR-17. This then drops to 0.53 once nodes in the gold graphs are merged based on the coreference annotations.

5 Discussion

The coreference chains annotated in the MS-AMR corpus are quite heterogeneous. At the token level, mentions of the same chain can be expressed as verbs, nouns or pronouns and Fig. 3 illustrates one example where the chain includes different concepts at the node level: *it*, *thing*, *harm-01*, *cut-01*. Such chains are hard to predict for the AllenNLP coreference model because they are realized as different parts of speech and are semantically nontrivial (harm/cut). 35% of all coreference chains in the test set are heterogeneous, i.e. they include entities that are expressed with multiple different parts of speech.



```
–<identchain relationid="rel-1">
    <mention concept="it" id="DF-200-192400-625_7557.9" variable="i2"/>
    <mention concept="thing" id="DF-200-192400-625_7557.24" variable="t3"/>
    <mention concept="it" id="DF-200-192400-625_7557.27" variable="i2"/>
    <mention concept="harm-01" id="DF-200-192400-625_7557.35" variable="h"/>
    <mention concept="cut-01" id="DF-200-192400-625_7557.3" variable="c4"/>
    <mention concept="do-02" id="DF-200-192400-625_7557.17" variable="d2"/>
    <mention concept="cut-01" id="DF-200-192400-625_7557.8" variable="c"/>
    <mention concept="this" id="DF-200-192400-625_7557.29" variable="t"/>
    <mention concept="harm-01" id="DF-200-192400-625_7557.1" variable="h"/>
    <mention concept="cut-01" id="DF-200-192400-625_7557.6" variable="c"/>
    <mention concept="it" id="DF-200-192400-625_7557.36" variable="i"/>
    <mention concept="thing" id="DF-200-192400-625_7557.28" variable="t2"/>
 </identchain>
```

Figure 3: Heterogeneous coreference chain from MS-AMR.

On the one hand, AMR parsing already resolves some cases of coreference within the AMR graphs.

For instance, in Fig. 2 (a) a single node *o:offer-01* aligns to coreferent tokens “offers” and “them”. On the other hand, some AMR nodes can build coreference chains but do not have any token alignments. For example, a sentence like “speak to a doctor” has a separate node “you” as ARG0 of “speak-01” in the AMR graph. However, this node does not correspond to any token in the text. 9% of all coreferent mentions in the MS-AMR test set do not have any alignments and the token-based coreference resolvers are not able to handle them.

Incorrect (or incomplete) node-token alignments can hurt the performance. 10% of all coreferent nodes in the test set refer to generic concepts like *t:thing* or *p:person*. This becomes a problem when AllenNLP finds the coreference with more specific nodes such as *d:dad* in Fig. 4. Token “dad” is aligned to the node *d:dad* in the AMR graph whereas the more generic node *p:person* does not have an alignment. However, the gold coreference chain includes only *p:person* as a member which results in the wrong classification of *d:dad* as false positive although both nodes actually correspond to the same entity. This example illustrates the problem when the gold annotation includes generic concepts that are represented in the AMR graphs but not realized at the token level.

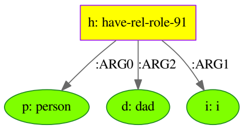


Figure 4: AMR for *“my dad”*.

We also found cases of incorrectly resolved personal pronouns because some texts were extracted from forums and the speaker could switch in the middle of the conversation, so that *I* and *you* would get a different meaning. For example, one document in the MS-AMR test set contains the following text: “Or should $[\text{I}]_1$ ... just keep an eye on the anxiety until it becomes a problem? Well $[\text{I}]_2$ woudn’t try to keep an eye on anxiety for a start because that will make $[\text{u}]_1$ tense.” The first sentence has the pronoun $[\text{I}]_1$ that refers to the same entity as $[\text{u}]_1$ in the second sentence and the $[\text{I}]_2$ pronoun in the second sentence corresponds to a different speaker. Since the input text for the coreference tool does not include any meta information about the speakers the tool resolves both occurrences of “I” as referring to the same entity. This issue affects 9% of the coreference chains from the MS-AMR test set.

6 Conclusion

In this paper, we evaluated two popular coreference resolution tools on the MS-AMR dataset, and found that the SpanBERT version of AllenNLP performs best in both a token-level and a node-level evaluation. We further extended a state-of-the-art AMR parser with predicted coreference information, and obtained a three-point improvement in Smatch score.

The coreference models we have used here were quite conservative, in that they relied only on textual information. In the future, it would be interesting to extend them with features based on the AMR graphs, which abstract over some surface details. It would also be interesting to predict bridging coreference relations and include those in the parser output too.

Acknowledgments. We thank Jonas Groschwitz and Matthias Lindemann for fruitful discussions and for their help with the Lindemann et al. parser. We also thank Tim O’Gorman for providing the MS-AMR corpus.

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, LAW-ID@ACL 2013, August 8-9, 2013, Sofia, Bulgaria*, pages 178–186.

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, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 748–752.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Heeyoung Lee, Angel X. Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 188–197.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4576–4585.

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, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3693–3702.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.