A Simple Method for Clarifying Sentences with Coordination Ambiguities

Michael White and **Manjuan Duan** and **David L. King**
Department of Linguistics
The Ohio State University
Columbus, OH 43210 USA
mwhite@ling.osu.edu, {duan.59,king.2138}@osu.edu

Abstract

We present a simple, broad coverage method for clarifying the meaning of sentences with coordination ambiguities, a frequent cause of parse errors. For each of the two most likely parses involving a coordination ambiguity, we produce a disambiguating paraphrase that splits the sentence in two, with one conjunct appearing in each half, so that the span of each conjunct becomes clearer. In a validation study, we show that the method enables meaning judgments to be crowd-sourced with good reliability, achieving 83% accuracy at 80% coverage.

1 Introduction

In principle, intelligent systems should be capable of explaining how they have interpreted unrestricted natural language sentences. Although some early dialogue systems such as SHRDLU (Winograd, 1973) could ask questions to clarify the meaning of certain structurally ambiguous sentences, little work has been done to date on the task of generating questions to clarify structural ambiguities in a broad coverage setting. Recently, Duan et al. (2016) have shown that generating unambiguous paraphrases from competing parses of structurally ambiguous sentences can serve as a useful method for asking to clarify their intended meaning; in particular, they showed that their method enables crowd-sourced meaning judgments to be collected in order to improve parser accuracy in new domains. Duan et al.'s study covered most of the major sources of common parser errors identified by Kummerfeld et al. (2012), with the exception of ambiguities involving the correct spans of conjuncts in coordinated phrases (unless they involve modifier attachment ambiguities). Also closely related is He et al.'s (2016) work on generating questions to identify semantic roles, though their work does not address coordination span ambiguities either.

In this paper, we present a novel method for generating disambiguating paraphrases for sentences with ambiguities involving two coordinated elements where the sentence is split in two, with one conjunct appearing in each half, so that the span of each conjunct becomes clearer. In a validation study, we show that the method enables meaning judgments to be crowd-sourced with good reliability. Following an error analysis that highlights problematic cases, we conclude with a discussion of ways in which the method could be improved.

2 Disambiguation Method

At a high level, our method for generating disambiguating paraphrases for sentences with coordination ambiguities is as follows:

1. Parse the sentence and determine whether the most likely parse (henceforth the 'top' parse) has a coordinated phrase with two conjuncts/disjuncts, recording its span in words.

2. Examine the subsequent parses in the n-best list (in order) to determine whether the parse (henceforth the 'next' parse) has a coordinated phrase with a different span.

3. If such 'top' and 'next' parses are found, generate paraphrases by

Proceedings of the 1st Workshop on Explainable Computational Intelligence, pages 6–10, Santiago de Compostela, Spain, September 4 2017. ©2017 Association for Computational Linguistics

```
(w1 / do [mood=dcl tense=pres]
 :Arg0 (w0 / they)
 :Arg1 (w2 / have
        :Arg0 w0
        :Arg1 (w5 / selection [num=sg]
               :Det (w3 / a)
               :Mod (w4 / good)
               :Mod (w6 / of
                     :Arg1 (w8 / and
                            :First (w7 / fabric
                                       [det=nil num=sg])
                            :Next (w9 / notion
                                      [det=nil num=pl]))))))
```

(a) Semantic dependency graph of 'top' parse

```
(w1 / do [mood=dcl tense=pres]
 :Arg0 (w0 / they)
 :Arg1 (w2 / have
        :Arg0 w0
        :Arg1 (w8 / and
               :First (w5 / selection [num=sg]
                        :Det (w3 / a)
                        :Mod (w4 / good)
                        :Mod (w6 / of
                               :Arg1 (w7 / fabric
                                          [det=nil num=sg])))
               :Next (w9 / notion [det=nil num=pl]))))
```

(b) Semantic dependency graph of 'next' parse

Figure 1: Most likely parses for (1)

(a) copying the words up to and including the first conjunct, followed by the words following the coordinated phrase;

(b) copying any sentence-final punctuation, then starting a new sentence by copying the conjunction; and

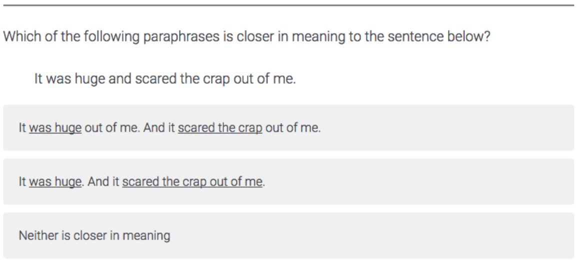
(c) again copying the words up to the first conjunct, then copying the second conjunct, again followed by the words following the coordinated phrase.

To illustrate, consider (1) below, a sentence from the English Web Treebank,[1] a corpus which is primarily out-of-domain for parsers trained on the original Penn Treebank. This sentence has a coordination ambiguity between *a good selection of [[fabric] and [notions]]* and *[[a good selection of fabric] and [notions]]*, which is not (conventionally) analyzed as a modifier attachment ambiguity.[2]

(1) They do have a good selection of fabric and notions.

a. They do have a good selection of fabric. And they do have a good selection of notions.

b. They do have a good selection of fabric. And they do have notions.



Which of the following paraphrases is closer in meaning to the sentence below?

It was huge and scared the crap out of me.

It was huge out of me. And it scared the crap out of me.

It was huge. And it scared the crap out of me.

Neither is closer in meaning

Figure 2: Sample survey question

The disambiguating paraphrase for the former, 'top' parse (correct according to the English Web Treebank) appears in (1-a), and the one for the latter, 'next' parse appears in (1-b), with underlining to highlight the differences between them.

To parse sentences, we use the Berkeley parser (Petrov et al., 2006) trained on OpenCCG[3] derivations (White, 2006; White et al., 2007; Boxwell and White, 2008) extracted from the CCGbank (Hockenmaier and Steedman, 2007). Derivations yield semantic dependency graphs represented using Hybrid Logic Dependency Semantics; the dependency graphs for (1) appear in Figure 1 using AMR-style notation (Banarescu et al., 2013). As shown in the figure, the `:First` and `:Next` relations can be used to identify coordinated phrases, and word identifiers allow spans to be extracted from subtrees.[4]

3 Crowd-Sourcing Judgments

We used Amazon Mechanical Turk (AMT) to crowd-source meaning judgments using our method of paraphrasing coordination ambiguities. Workers were given no training and very simple instructions, namely to select the paraphrase that is closer

[1] https://catalog.ldc.upenn.edu/ldc2012t13

[2] Note that sentences with ambiguities involving postmodifiers are dealt with symmetrically.

[3] http://openccg.sourceforge.net/

[4] Note that the conjunct spans can be accurately obtained even for shared argument coordination, e.g. VP-coordination or right node raising. Also note that as an alternative to the simple, surface-level algorithm employed here, we could have made changes to the dependency graphs and used OpenCCG to realize the modified graphs back as two sentences, which could avoid occasional errors with subject-verb agreement; the advantage of the present method is that it ensures that no undesired changes are made elsewhere in the sentence, as can happen with a broad coverage surface realizer.

Coverage	Accuracy			
	Majority		MACE	
	All	w/ Excl.	All	Filtered
25%	1.00	-	1.00	1.00
35%	-	0.98	0.97	1.00
50%	0.93	-	0.89	0.91
60%	-	0.88	0.87	0.90
80%	-	-	**0.83**	**0.84**
100%	0.75	0.75	0.74	-

Table 1: Coverage vs. accuracy highlights using majority vote (majority, strong majority, near unanimity) and MACE with all workers; majority vote with poorly performing workers excluded; and MACE with 'neither' responses filtered out

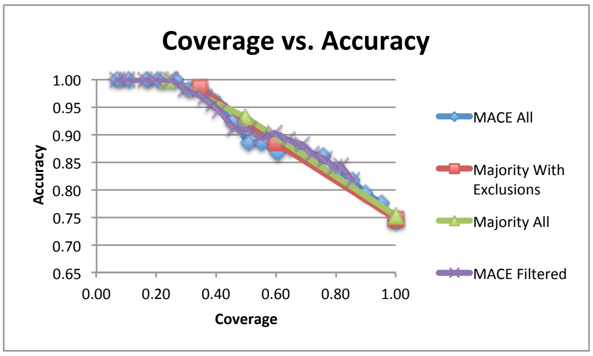


Figure 3: Coverage vs. accuracy using majority vote (majority, strong majority, near unanimity) and MACE across various confidence levels

in meaning to the original sentence, even it does not mean exactly the same thing, or 'neither' if neither sentence is closer in meaning. A screen shot showing a survey question appears in Figure 2.

For our validation experiment, we generated paraphrases for 172 items taken from the development section of the English Web Treebank. From these 172 items, twelve that were relatively short and clear were selected to be control items. The items were randomly distributed across eight surveys, with each survey containing 28 items, of which eight were control items, with four items per page.

We sought five workers to complete each survey. Workers were required to have a US IP address, be native speakers of English, and have achieved Masters status on AMT. Workers were told that they needed to get 75% of the control items correct. For five of the eight surveys, one worker failed to achieve 75% correct on the control items, so we sought an additional worker for each of these. All workers were paid $2 per survey, including the ones who failed to reach 75% on the control items, as they did not appear to be answering randomly. Each survey took 10-15 minutes to complete.

4 Results

4.1 Coverage vs. Accuracy

We measured the accuracy of the crowd-sourced judgments against our own expert judgments at various coverage levels. The results appear in Table 1 and Figure 3. Unlike the crowd-sourced judgments, our expert judgments were based on examining the 'top' and 'next' parses to see which one (if any) was more correct, consulting the structure annotated in the English Web Treebank in cases with any doubt.

One way to aggregate crowd-sourced judgments across multiple annotators is to simply take the majority judgment, breaking ties randomly. In this case, a consensus judgment is obtained for all items, so coverage is 100%. As shown in the table, accuracy at this coverage level is 75%, much higher than the chance level of 33.3%. A trade-off between coverage and accuracy can also be obtained by requiring a super-majority: for a strong majority, we required at least 75% agreement, and for (near) unanimity, we required at least 90% agreement. When all annotators are included—even those who performed poorly on the control items—requiring a strong majority reduces coverage to only 50%, but accuracy goes up to 93%; with the poorly performing annotators excluded, there are more strong majority cases, with 60% coverage, but accuracy is relatively lower, at 88%. Requiring (near) unanimity reduces coverage further, but raises accuracy to near 100%.

As an alternative to using majority judgments, MACE[5] (Hovy et al., 2013) can be used to make consensus predictions by weighting annotator judgments by their competence, where competence is estimated using expectation maximization. These consensus predictions can be assigned a confidence value according not only to agreement but also to estimated annotator competence. We ran MACE with thresholds to retain only the 5%, 10%, 15%,

[5] Multi-Annotator Competence Estimation

Error type	Count
preceding modifier scope ambiguity	14
following modifier scope ambiguity	4
apposition	4
miscellaneous	8
‘neither’ cases	14
total errors	44

Table 2: Distribution of errors

... 100% of the items with the highest model confidence, as shown in the figure; with MACE, it made little difference whether poorly performing annotators were excluded, so we only show the results with all annotators here. The accuracy of the MACE-derived consensus judgments was no better than with the majority judgments, but MACE did make it possible to identify a sweet spot where coverage is still high at 80% while accuracy is substantially higher at 83% than in the full-coverage case. Finally, the table and figure also show the coverage and accuracy when items where ‘neither’ was the consensus judgment are excluded, as these would be unhelpful for parser adaptation: here, a slightly higher accuracy of 84% is attained at the 80% coverage level.

4.2 Error Analysis

The distribution of errors using MACE at the 100% coverage level appears in Table 2. Out of 172 items, the annotator consensus differed from our judgment in 44 cases. Most of the errors were related to either modification or apposition. The miscellaneous errors were ones that only occurred once. There were also 7 items where neither parse was more correct, and 7 where the annotator consensus was erroneously ‘neither’, typically because parse errors led to hard-to-understand paraphrases.

Of the 30 remaining (non-‘neither’) errors, roughly half involved preceding modifiers with ambiguous scope. Although our paraphrasing method handles preceding modifiers within noun phrases reasonably well, adverbial scope proved more difficult to disambiguate. For example, consider (2):

(2) So go and get dancing!!!!!!!!!!!!!!!!!!!!!!!!!.

a. So go![...]. And so get dancing![...].

b. So go![...]. And get dancing![...].

Although without context, it is somewhat difficult to tell whether the scope of the discourse connective *so* applies to both imperative clauses or only the first, our crowd sourced annotators overwhelmingly preferred the paraphrase (3-b) of the ‘next’ parse, contrary to the English Web Treebank. One possible reason is that here, repeating *so* in (3-a) is quite awkward. Additionally, although *so* is only in the first sentence of paraphrase (3-b), it is easy to interpret it as also modifying the second clause.

Of the remaining errors, paraphrases from appositive constructions such as (3) stood out, as these do not have a straightforwardly distributive interpretation. Likewise, there were a couple errors involving collective readings for conjoined noun phrases.

(3) Shuttle veteran and longtime NASA executive Fred Gregory is temporarily at the helm of the 18,000-person agency.

a. Shuttle veteran is temporarily at the helm of the 18,000-person agency. And long time NASA executive Fred Gregory is temporarily at the helm of the 18,000-person agency.

b. Shuttle veteran is temporarily at the helm of the 18,000-person agency. And shuttle long time NASA executive Fred Gregory is temporarily at he helm of the 18,000-person agency.

5 Discussion

Our validation study has shown that our simple, broad coverage method for clarifying the meaning of sentences with coordination ambiguities enables meaning judgments to be crowd-sourced with good reliability, far above chance and at a level that can be expected to pay off for parser domain adaptation. Since the method is so simple, it should be possible to adapt to a variety of parsing frameworks.

Not surprisingly, an error analysis revealed that sentences whose interpretations are not straightforwardly distributive are problematic for our method, indicating that a more sophisticated way to handle such sentences is required. Less obviously, adverbial pre-modifiers turned out to work relatively poorly, suggesting that Duan et al.’s (2016) method for disambiguating these represents a better option.

Acknowledgments

We thank the OSU Clippers Group and the anonymous reviewers for helpful comments and discussion. This work was supported in part by NSF grant IIS-1319318.

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. Association for Computational Linguistics.

Stephen Boxwell and Michael White. 2008. Projecting Propbank roles onto the CCGbank. In *Proc. LREC-08*.

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. Association for Computational Linguistics.

Luheng He, Julian Michael, Mike Lewis, and Luke Zettlemoyer. 2016. Human-in-the-loop parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2337–2342. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130. Association for Computational Linguistics.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059, Jeju Island, South Korea, July. Association for Computational Linguistics.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL*.

Michael White, Rajakrishnan Rajkumar, and Scott Martin. 2007. Towards broad coverage surface realization with CCG. In *Proc. of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation (UCNLG+MT)*.

Michael White. 2006. Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. *Research on Language & Computation*, 4(1):39–75.

Terry Winograd. 1973. A procedural model of language understanding. In Roger Schank and Ken Colby, editors, *Computer Models of Thought and Language*, pages 152–186. W.H. Freeman. Reprinted in Grosz et al. (eds), Readings in Natural Language Processing. Los Altos CA: Morgan Kaufmann Publishers, 1986, pp.249-266.