

Generating automatic pseudo-entailments from AMR parses

Adam Poliak Benjamin Van Durme
Center for Language and Speech Processing
Johns Hopkins University
{azpoliak, vandurme}@cs.jhu.edu

Abstract

We explore how to generate a textual inference dataset from Abstract Meaning Representations (AMR) (Banarescu et al., 2013). Various aspects of AMR make it problematic to automatically derive inference patterns. Therefore, our generated dataset instead answers questions regarding the relation between entities and actions in sentences. We refer to these answers as pseudo-entailments. From this dataset, it is possible to then extract entailments from sentences.

1 Introduction

Abstract Meaning Representation (AMR) is a method to universally represent the meaning of entire sentences (Banarescu et al., 2013). AMR takes advantage of PropBank semantic roles (Palmer et al., 2005) to represent relations between concepts and entities in a sentence.¹ AMR parses are acyclic directed graphs rooted with the main idea of the sentence. Each entity can be represented as a vertex and the relation between entities can be represented as an edge. This makes parsing AMR relatively straight-forward.

2 Limitations of AMR

Despite the proliferation of AMR, the format has its drawbacks. Important information related to generating textual inferences from a sentence is often missing in an AMR parse. For instance, AMR does not represent articles, inflectional morphology or tense. Figure 1 represents the AMR parse of the sentence “I’m just taking a look.”

Generating textual inferences from the AMR in Figure 1 is difficult since the phrases “*I just took a*

¹An overview of PropBank can be found at <http://proppbank.github.io/>.

```
(l / look-01  
:ARG0 (i / i)  
:mod (j / just))
```

Figure 1: AMR parse for the sentence “I’m just taking a look”

look” and “*I will just take a look*” would share the same AMR as above. Generating textual inference solely from an AMR is a challenging and complex task.

3 Modified task: Pseudo-entailments

Therefore, instead of generating textual inferences from an AMR, we have decided to generate “pseudo-entailments” which are simple sentences that describe the different relations between entities in an AMR parse according to PropBank labels. Thus each pseudo-entailment we generate from an AMR will look as follows:

X is the Y of the Z

where *Z* is the present participle² of a PropBank Frame in the AMR, *X* is an entity associated with *Z*, and *Y* is the association or relation between *X* and *Z*. Even though there are roughly 100 relations used in AMR, we are only concerned with frame arguments that follow the PropBank conventions of :arg0 through :arg5. Since pseudo-entailments should be as close as possible to natural English sentences and numbered arguments in PropBank “correspond to slightly different semantic roles given the usage of each predicate”³, we substitute *Y* for the frame argument’s corresponding semantic role.

²We use the pattern module released by the CLiPS Research Center. Documentation can be found at <http://www.clips.ua.ac.be/pages/pattern>

³[http://clear.colorado.edu/compsem/documents/proppbank\\$_guidelines.pdf](http://clear.colorado.edu/compsem/documents/proppbank$_guidelines.pdf) Page 4

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*. Association for Computational Linguistics, Sofia, Bulgaria, pages 178–186. <http://www.aclweb.org/anthology/W13-2322>.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. [The proposition bank: An annotated corpus of semantic roles](#). *Computational Linguistics* 31(1):71–106. <https://doi.org/10.1162/0891201053630264>.