

An AMR parser for English, French, German, Spanish and Japanese and a new AMR-annotated corpus

Lucy Vanderwende, Arul Menezes, Chris Quirk

Microsoft Research
One Microsoft Way
Redmond, WA 98052
{lucyv, arulm, chrisq}@microsoft.com

Abstract

In this demonstration, we will present our online parser¹ that allows users to submit any sentence and obtain an analysis following the specification of AMR (Banarescu et al., 2014) to a large extent. This AMR analysis is generated by a small set of rules that convert a native Logical Form analysis provided by a pre-existing parser (see Vanderwende, 2015) into the AMR format. While we demonstrate the performance of our AMR parser on data sets annotated by the LDC, we will focus attention in the demo on the following two areas: 1) we will make available AMR annotations for the data sets that were used to develop our parser, to serve as a supplement to the LDC data sets, and 2) we will demonstrate AMR parsers for German, French, Spanish and Japanese that make use of the same small set of LF-to-AMR conversion rules.

1 Introduction

Abstract Meaning Representation (AMR) (Banarescu et al., 2014) is a semantic representation for which a large amount of manually-annotated data is being created, with the intent of constructing and evaluating parsers that generate this level of semantic representation for previously unseen text.

¹ Available at: <http://research.microsoft.com/msrsplat>

Already one method for training an AMR parser has appeared in (Flanigan et al., 2014), and we anticipate that more attempts to train parsers will follow. In this demonstration, we will present our AMR parser, which converts our existing semantic representation formalism, Logical Form (LF), into the AMR format. We do this with two goals: first, as our existing LF is close in design to AMR, we can now use the manually-annotated AMR datasets to measure the accuracy of our LF system, which may serve to provide a benchmark for parsers trained on the AMR corpus. We gratefully acknowledge the contributions made by Banarescu et al. (2014) towards defining a clear and interpretable semantic representation that enables this type of system comparison. Second, we wish to contribute new AMR data sets comprised of the AMR annotations by our AMR parser of the sentences we previously used to develop our LF system. These sentences were curated to cover a wide-range of syntactic-semantic phenomena, including those described in the AMR specification. We will also demonstrate the capabilities of our parser to generate AMR analyses for sentences in French, German, Spanish and Japanese, for which no manually-annotated AMR data is available at present.

2 Abstract Meaning Representation

Abstract Meaning Representation (AMR) is a semantic representation language which aims to assign the same representation to sentences that have

the same basic meaning (Banarescu et al., 2014). Some of the basic principles are to use a graph representation, to abstract away from syntactic idiosyncrasies (such as active/passive alternation), to introduce variables corresponding to entities, properties and events, and to ground nodes to OntoNotes (Pradhan et al., 2007) wherever possible.

As a semantic representation, AMR describes the analysis of an input sentence at both the conceptual and the predicative level, as AMR does not annotate individual words in a sentence (see annotation guidelines, introduction). AMR, for example, provides a single representation for the constructions that are typically thought of as alternations: “it is tough to please the teacher” and “the teacher is tough to please” have the same representation in AMR, as do actives and their passive variant, e.g., “a girl read the book” and “the book was read by a girl”. AMR also advocates the representation of nominative constructions in verbal form, so that “I read about the destruction of Rome by the Vandals” and “I read how the Vandals destroyed Rome” have the same representation in AMR, with the nominal “destruction” recognized as having the same basic meaning as the verbal “destroy”. Such decisions are part-conceptual and part-predicative, and rely on the OntoNotes lexicon having entries for the nominalized forms. AMR annotators also can reach in to OntoNotes to represent “the soldier was afraid of battle” and “the soldier feared battle”: linking “be afraid of” to “fear” depends on the OntoNotes frameset at annotation time.

3 Logical Form

The Logical Form (LF) which we convert to AMR via a small set of rules is one component in a broad-coverage grammar pipeline (see Vanderwende, 2015, for an overview). The goal of the LF is twofold: to compute the predicate-argument structure for each clause (“who did what to whom when where and how?”) and to normalize differing syntactic realizations of what can be considered the same “meaning”. In so doing, concepts that are possibly distant in the linear order of the sentence or distant in the constituent structure can be brought together, because the Logical Form is represented as a graph, where linear order is no longer primary. In addition to alternations and passive/active, other operations include: unbounded

dependencies, functional control, indirect object paraphrase, and assigning modifiers.

As in AMR, the Logical Form is a directed, labeled graph. The nodes in this graph have labels that are either morphologically or derivationally related to the input tokens, and the arcs are labeled with those relations that are defined to be semantic. Surface words that convey syntactic information only (e.g. *by* in a passive construction, *do*-support, singular/passive, or (in)definite articles) are not part of the graph, their meaning, however is preserved as annotations on the conceptual nodes (similar to the Prague T-layer, Hajič et al., 2003).

```

have1 (+Pres +Proposition)
  Dsub elephant1 (+Plur)
    Attrib African1 ()
      hunt1 (+Pass +Perf ...)
        Dsub X1
        Dobj elephant1
        for decade1 (+Plur)
  Dobj tusk1 (+Plur)
    Attrib large1 ()

```

Figure 1. The LF representation of "African elephants, which have been hunted for decades, have large tusks."

In Figure 1, we demonstrate that native LF uses reentrancy in graph notation, as does AMR, whenever an entity plays multiple roles in the graph. Note how the node *elephant1* is both the *Dsub* of *have1* and the *Dobj* of *hunt1*. The numerical identifiers on the leaf nodes are a unique label name, not a sense identification.

We also point out that LF attempts to interpret the syntactic relation as a general semantic relation to the degree possible, but when it lacks information for disambiguation, LF preserves the ambiguity. Thus, in Figure 1, the identified semantic relations are: *Dsub* (“deep subject”), *Attrib* (attributive), *Dobj* (“deep object”), but also the underspecified relation “for”.

The canonical LF graph display proceeds from the root node and follows a depth first exploration of the nodes. When queried, however, the graph can be viewed with integrity from the perspective of any node, by making use of relation inversions. Thus, a query for the node *elephant1* in Figure 1 returns *elephant1* as the *DsubOf* *have1* and also the *DobjOf* *hunt1*.

<i>System</i>	Test without word sense annotations				Test with word sense annotations			
	Proxy	DFA	Bolt	<i>Average</i>	Proxy	DFA	Bolt	<i>Average</i>
JAMR: proxy	64.4	40.4	44.2	<i>49.7</i>	63.3	38.1	42.6	<i>48.0</i>
JAMR: all	60.9	44.5	47.5	<i>51.0</i>	60.1	43.2	46.0	<i>49.8</i>
LF	59.0	50.7	52.6	54.1	55.2	46.9	49.2	50.4

Table 1. Evaluation results: balanced F-measure in percentage points. JAMR (proxy) is the system of Flanigan et al. (2014) trained on only the proxy corpus; JAMR (all) is the system trained on all data in LDC2014T12; and LF is the system described in this paper. We evaluate with and without sense annotations in three test corpora.

4 LF to AMR conversion

The description of LF in section 3 emphasized the close similarity of LF and AMR. Thus, conversion rules can be written to turn LF into AMR-similar output, thus creating an AMR parser. To convert the majority of the relations, only simple renaming is required; for example LF Dsub is typically AMR ARG0, LF Locn is AMR location, and so on.

We use simple representational transforms to convert named entities, dates, times, numbers and percentages, since the exact representation of these in AMR are slightly different from LF.

Some of the more interesting transforms to encourage similarity between LF and AMR are mapping modal verbs *can*, *may* and *must* to *possible* and *obligate* in AMR and adjusting how the copula is handled. In both AMR and LF the arguments of the copula are moved down to the object of the copula, but in LF the vestigial copula remains, whereas in AMR it is removed.

5 Evaluation

Using smatch (Cai and Knight, 2013), we compare the performance of our LF system to the JAMR system of Flanigan et al. (2014). Both systems rely on the Illinois Named Entity Tagger (Ratinov and Roth, 2009). LF strives to be a broad coverage parser without bias toward a particular domain. Therefore, we wanted to evaluate across a number of corpora. When trained on all available data, JAMR should be less domain dependent. However, the newswire data is both larger and important, so we also report numbers for JAMR trained on proxy data alone.

To explore the degree of domain dependence of these systems, we evaluate on several genres provided by the LDC: DFA (discussion forums data from English), Bolt (translated discussion forum data), and Proxy (newswire data). We did not ex-

periment on the consensus, mt09sdl, or Xinhua subsets because the data was pre-tokenized. This tokenization must be undone before our parser is applied.

We evaluate in two conditions: “without word sense annotations” indicates that the specific sense numbers were discarded in both the gold standard and the system output; “with word sense annotations” leaves the sense annotations intact.

The AMR specification requires that concepts, wherever possible, be annotated with a sense ID referencing the OntoNotes sense inventory. Recall that the LF system intentionally does not have a word sense disambiguation component due to the inherent difficulty of defining and agreeing upon task-independent sense inventories (Palmer et al. 2004, i.a.). In order to evaluate in the standard evaluation setup, we therefore construct a word-sense disambiguation component for LF lemmas. Our approach is quite simple: for each lemma, we find the predominant sense in the training set (breaking ties in favor of the lowest sense ID), and use that sense for all occurrences of the lemma in test data. For those lemmas that occur in the test but not in the training data, we attempt to find a verb frame in OntoNotes. If found, we use the lowest verb sense ID not marked with DO NOT TAG; otherwise, the lemma is left unannotated for sense. Such a simple system should perform well because 95% of sense-annotated tokens in the proxy training set use the predominant sense. An obvious extension would be sensitive to parts-of-speech.

As shown in Table 1, the LF system outperforms JAMR in broad-domain semantic parsing, as measured by macro-averaged F1 across domains. This is primarily due to its better performance on discussion forum data. JAMR, when trained on newswire data, is clearly the best system on newswire data. Adding training data from other sources leads to improvements on the discussion forum

data, but at the cost of accuracy on newswire. The lack of sophisticated sense disambiguation in LF causes a substantial degradation in performance on newswire.

6 Data Sets for LF development

The LF component was developed by authoring rules that access information from a rich lexicon consisting of several online dictionaries as well as information output by a rich grammar formalism. Authoring these LF rules is supported by a suite of tools that allow iterative development of an annotated test suite (Suzuki, 2002). We start by curating a sentence corpus that exemplifies the syntactic and semantic phenomena that the LF is designed to cover; one might view this sentence corpus as the LF specification. When, during development, the system outputs the desired representation, that LF is saved as “gold annotation”. In this way, the gold annotations are produced by the LF system itself, automatically, and thus with good system internal consistency. We note that this method of system development is quite different from SemBanking AMR, but is similar to the method described in Flickinger et al. (2014).

As part of this demonstration, we share with participants the gold annotations for the curated sentence corpora used during LF development, currently 550 sentences that are vetted to produce correct LF analyses. Note that the example in Figure 2 requires a parser to handle both the passive/active alternation as well as control verbs. We believe that there is value in curated targeted datasets to supplement annotating natural data; e.g., AMR clearly includes control phenomena in its spec (the first example is “the boy wants to go”) but in the data, there are only 3 instances of “persuade” in the amr-release-1.0-training-proxy, e.g., and no instances in the original AMR-bank.

7 AMR parsers for French, German, Spanish and Japanese

The demonstrated system includes not only a parser for English, but also parsers for French, German, Spanish and Japanese that produce analyses at the LF level. Thus, using the same set of conversion rules, we demonstrate AMR annotations gen-

```
# Pat was persuaded by Chris to eat the apple.
(p / persuade
  :ARG0 (p2 / person
    :name (c / name :op1 Chris))
  :ARG2 (e / eat
    :ARG0 (p4 / person
      :name (p3 / name :op1 Pat))
    :ARG1 (a / apple))
  :ARG1 p3)
```

Figure 2. LF-AMR for the input sentence “Pat was persuaded by Chris to eat the apple”, with both passive and control constructions.

erated by our parsers in these additional languages, for which there are currently no manually-annotated AMR SemBanks. Such annotations may be useful to the community as initial analyses that can be manually edited and corrected where their output does not conform to AMR-specifications already. Consider Figures 3-6 and the brief description of the type of alternation they are intended to demonstrate in each language.

```
Input: el reptil se volteó, quitándoselo de encima.
Gloss: the crocodile rolled over, throwing it off.
(v / voltear
  :ARG0 (r / reptil)
  :manner (q / quitar
    :ARG0 r
    :ARG1 (x / "él")
    :prep-de (e / encima)))
```

Figure 3 AMR in Spanish with clitic construction.

```
Input: Et j'ai vu un petit bonhomme tout à fait extraordinaire qui me considérait gravement.
Gloss: And I saw a small chap totally extraordinary who me looked seriously.
```

```
(e / et
  :op (v / voir
    :ARG0 (j / je)
    :ARG1 (b / bonhomme
      :ARG0-of (c / "considérer"
        :ARG1 j
        :mod (g / gravement))
      :mod (p / petit)
      :mod (e2 / extraordinaire
        :degree (t / "tout_à_fait")))))
```

Figure 4 AMR in French with re-entrant node “j”

Input: Die dem wirtschaftlichen Aufschwung zu verdankende sinkende Arbeitslosenquote führe zu höheren Steuereinnahmen.

Gloss: The the economic upturn to thank-for sinking unemployment rate led to higher tax-revenue

```
(f / "führen"  
  :ARG0 (a / Arbeitslosenquote  
    :ARG0-of (s / sinken)  
    :ARG0-of (v / verdanken  
      :ARG2 (a2 / Aufschwung  
        :mod (w / wirtschaftlich))  
      :degree (z / zu)))  
  :prep-zu (s2 / Steuereinnahme  
    :mod (h / hoch)))
```

Figure 5 AMR in German for complex participial construction

Input: 東国の諸藩主に勤王を誓わせた。

Gloss: eastern_lands various feudal_lords
serve_monarchy swear-CAUS-PAST

東国の諸藩主に勤王を誓わせた。

```
(x / "させる" cause  
  :ARG1 (x2 / "誓う" swear  
    :ARG0 (x3 / "藩主" feudal_lords  
      :mod (x4 / "諸") various  
      :prep-の (x5 / "東国")) Eastern lands  
    :ARG1 (x6 / "勤王")) serve monarchy
```

Figure 6. AMR in Japanese illustrating a causative construction

8 Conclusion

In the sections above, we have attempted to highlight those aspects of the system that will be demonstrated. To summarize, we show a system that:

- Produces AMR output that can be compared to the manually-annotated LDC resources. Available at: <http://research.microsoft.com/msrsplat>,
- Produces AMR output for a new data set comprised of the sentences selected for the development of our LF component. This curated data set was selected to represent a wide range of phenomena and representational challenges. These sentences and their AMR annotations are available at: <http://research.microsoft.com/nlpwin-amr>
- Produces AMR annotations for French, German, Spanish and Japanese input, which may be used to speed-up manual annotation/correction in these languages.

Acknowledgements

We are grateful to all our colleagues who worked on NLPwin. For this paper, we especially recognize Karen Jensen, Carmen Lozano, Jessie Pinkham, Michael Gamon and Hisami Suzuki for their work on Logical Form. We also acknowledge Jeffrey Flanigan and his co-authors for their contributions of making the JAMR models and code available.

References

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2014. Abstract Meaning Representation (AMR) 1.2.1 Specification. Available at <https://github.com/amrisi/amr-guidelines/blob/master/amr.md>
- Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In Proceedings of ACL.
- Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah Smith. 2014. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. In Proceedings of ACL 2014.
- Dan Flickinger, Emily M. Bender and Stephan Oepen. 2014. Towards an Encyclopedia of Compositional Semantics: Documenting the Interface of the English Resource Grammar. In Proceedings of LREC.
- Jan Hajič, Alena Böhmová, Hajičová, Eva, and Hladká, Barbara. (2003). The Prague Dependency Treebank: A Three Level Annotation Scenario. In Abeillé, Anne, editor, Treebanks: Building and Using Annotated Corpora. Kluwer Academic Publishers.
- Martha Palmer, Olga Babko-Malaya, Hoa Trang Dang. 2004. Different Sense Granularities for Different Applications. In *Proceedings of Workshop on Scalable Natural Language Understanding*.
- Sameer. S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. OntoNotes: A Unified Relational Semantic Representation. In Proceedings of the International Conference on Semantic Computing (ICSC '07).
- Hisami Suzuki. 2002. A development environment for large-scale multi-lingual parsing systems. In Proceedings of the 2002 workshop on Grammar engineering and evaluation - Volume 15, Pages 1-7.
- Lucy Vanderwende. 2015. NLPwin – an introduction. Microsoft Research tech report no. MSR-TR-2015-23, March 2015.