AMRICA: an AMR Inspector for Cross-language Alignments

Naomi Saphra
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21211, USA
nsaphra@jhu.edu

Adam Lopez
School of Informatics
University of Edinburgh
Edinburgh, United Kingdom
alopez@inf.ed.ac.uk

Abstract

Abstract Meaning Representation (AMR), an annotation scheme for natural language semantics, has drawn attention for its simplicity and representational power. Because AMR annotations are not designed for human readability, we present AMRICA, a visual aid for exploration of AMR annotations. AMRICA can visualize an AMR or the difference between two AMRs to help users diagnose interannotator disagreement or errors from an AMR parser. AMRICA can also automatically align and visualize the AMRs of a sentence and its translation in a parallel text. We believe AMRICA will simplify and streamline exploratory research on cross-lingual AMR corpora.

1 Introduction

Research in statistical machine translation has begun to turn to semantics. Effective semantics-based translation systems pose a crucial need for a practical cross-lingual semantic representation. One such schema, Abstract Meaning Representation (AMR; Banarescu et al., 2013), has attracted attention for its simplicity and expressive power. AMR represents the meaning of a sentence as a directed graph over *concepts* representing entities, events, and properties like names or quantities. Concepts are represented by nodes and are connected by edges representing *relations*—roles or attributes. Figure 1 shows an example of the AMR annotation format, which is optimized for text entry rather than human comprehension.

For human analysis, we believe it is easier to visualize the AMR graph. We present AMRICA, a system for visualizing AMRs in three conditions. First, AMRICA can display AMRs as in Figure 2. Second, AMRICA can visualize differences between aligned AMRs of a sentence, enabling users to diagnose differences in multiple annotations or between an annotation and an automatic AMR parse (Section 2). Finally, to aid researchers studying cross-lingual semantics, AMRICA can visualize differences between the AMR of a sentence and that of its translation (Section 3) using a novel cross-lingual extension to Smatch (Cai and Knight, 2013). The AMRICA code and a tutorial are publicly available.[1]

```
(b / be-located-at-91 :li 4
  :ARG1 (i / i)
  :ARG2 (c / country
    :name (n / name
      :op1 "New" :op2 "Zealand"))
  :time (w / week :quant 2
    :time (p / past)))
```

Figure 1: AMR for "I've been in New Zealand the past two weeks." (Linguistic Data Consortium, 2013)

2 Interannotator Agreement

AMR annotators and researchers are still exploring how to achieve high interannotator agreement (Cai and Knight, 2013). So it is useful to visualize a pair of AMRs in a way that highlights their disagreement, as in Figure 3. AMRICA shows in black those nodes and edges which are shared between the annotations. Elements that differ are red if they appear in one AMR and blue if they appear in the other. This feature can also be used to explore output from an

[1] *http://github.com/nsaphra/AMRICA*

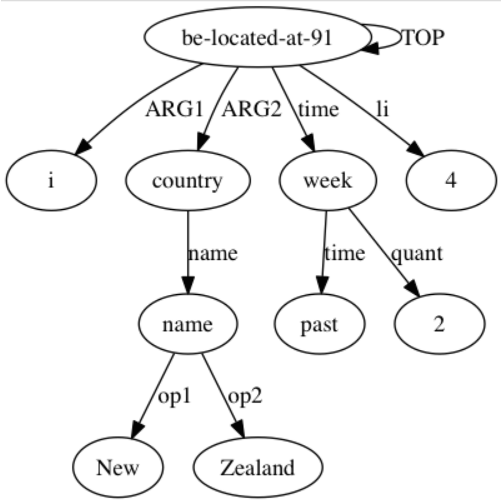

Figure 2: AMRICA visualization of AMR in Figure 1.

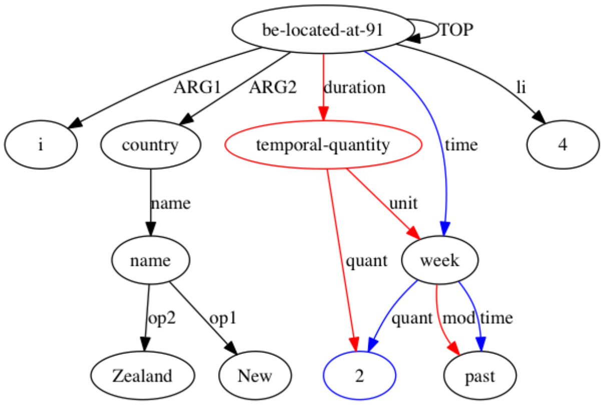

Figure 3: AMRICA visualization of the disagreement between two independent annotations of the sentence in Figure 1.

automatic AMR parser in order to diagnose errors.

To align AMRs, we use the public implementation of Smatch (Cai and Knight, 2013).[2] Since it also forms the basis for our cross-lingual visualization, we briefly review it here.

AMR distinguishes between *variable* and *constant* nodes. Variable nodes, like `i` in Figure 1, represent entities and events, and may have multiple incoming and outgoing edges. Constant nodes, like `2` in Figure 1, participate in exactly one relation, making them leaves of a single parent variable. Smatch compares a pair of AMRs that have each been decomposed into three kinds of relationships:

1. The set V of instance-of relations describe the conceptual class of each variable. In Figure 1, `(c / country)` specifies that `c` is an instance of a `country`. If node v is an instance of concept c, then $(v, c) \in V$.

2. The set E of variable-to-variable relations like `ARG2(b, c)` describe relationships between entities and/or events. If r is a relation from variable v_1 to variable v_2, then $(r, v_1, v_2) \in E$.

3. The set C of variable-to-constant relations like `quant(w, 2)` describe properties of entities or events. If r is a relation from variable v to constant x, then $(r, v, x) \in C$.

Smatch seeks the bijective alignment $\hat{b} : V \rightarrow V'$ between an AMR $G = (V, E, C)$ and a larger AMR $G' = (V', E', C')$ satisfying Equation 1, where $\mathbb{I}$ is an indicator function returning 1 if its argument is true, 0 otherwise.

$$\begin{aligned}\hat{b} = \arg\max_{b} & \sum_{(v,c)\in V} \mathbb{I}((b(v), c) \in V') + \\ & \sum_{(r,v_1,v_2)\in E} \mathbb{I}((r, b(v_1), b(v_2)) \in E') + \\ & \sum_{(r,v,c)\in C} \mathbb{I}((r, b(v), c) \in C')\end{aligned} \quad (1)$$

Cai and Knight (2013) conjecture that this optimization can be shown to be NP-complete by reduction to the subgraph isomorphism problem. Smatch approximates the solution with a hill-climbing algorithm. It first creates an alignment b_0 in which each node of G is aligned to a node in G' with the same concept if such a node exists, or else to a random node. It then iteratively produces an alignment b_i by greedily choosing the best alignment that can be obtained from b_{i-1} by swapping two alignments or aligning a node in G to an unaligned node, stopping when the objective no longer improves and returning the final alignment. It uses random restarts since the greedy algorithm may only find a local optimum.

3 Aligning Cross-Language AMRs

AMRICA offers the novel ability to align AMR annotations of bitext. This is useful for analyzing

[2] *http://amr.isi.edu/download/smatch-v2.0.tar.gz*

AMR annotation differences across languages, and for analyzing translation systems that use AMR as an intermediate representation. The alignment is more difficult than in the monolingual case, since nodes in AMRs are labeled in the language of the sentence they annotate. AMRICA extends the Smatch alignment algorithm to account for this difficulty.

AMRICA does not distinguish between constants and variables, since their labels tend to be grounded in the words of the sentence, which it uses for alignment. Instead, it treats all nodes as variables and computes the similarities of their node labels. Since node labels are in their language of origin, exact string match no longer works as a criterion for assigning credit to a pair of aligned nodes. Therefore AMRICA uses a function $L : V \times V \rightarrow \mathbb{R}$ indicating the likelihood that the nodes align. These changes yield the new objective shown in Equation 2 for AMRs $G = (V, E)$ and $G' = (V', E')$, where V and V' are now sets of nodes, and E and E' are defined as before.

$$\hat{b} = \arg\max_b \sum_{v \in V} L(v, b(v)) + \sum_{(r, v_1, v_2) \in E} \mathbb{I}((r, b(v_1), b(v_2)) \in E') \tag{2}$$

If the labels of nodes v and v' match, then $L(v, v') = 1$. If they do not match, then L decomposes over source-node-to-word alignment a_s, source-word-to-target-word alignment a, and target-word-to-node a_t, as illustrated in Figure 5. More precisely, if the source and target sentences contain n and n' words, respectively, then L is defined by Equation 3. AMRICA takes a parameter α to control how it weights these estimated likelihoods relative to exact matches of relation and concept labels.

$$L(v, v') = \alpha \sum_{i=1}^{n} \Pr(a_s(v) = i) \times \sum_{j=1}^{n'} \Pr(a_i = j) \cdot \Pr(a_t(v') = j) \tag{3}$$

Node-to-word probabilities $\Pr(a_s(v) = i)$ and $\Pr(a_s(v') = j)$ are computed as described in Section 3.1. Word-to-word probabilities $\Pr(a_i = j)$ are computed as described in Section 3.2. AMRICA uses the Smatch hill-climbing algorithm to yield alignments like that in Figure 4.

3.1 Node-to-word and word-to-node alignment

AMRICA can accept node-to-word alignments as output by the heuristic aligner of Flanigan et al. (2014).[3] In this case, the tokens in the aligned span receive uniform probabilities over all nodes in their aligned subgraph, while all other token-node alignments receive probability 0. If no such alignments are provided, AMRICA aligns concept nodes to tokens matching the node's label, if they exist. A token can align to multiple nodes, and a node to multiple tokens. Otherwise, alignment probability is uniformly distributed across unaligned nodes or tokens.

3.2 Word-to-word Alignment

AMRICA computes the posterior probability of the alignment between the ith word of the source and jth word of the target as an equal mixture between the posterior probabilities of source-to-target and target-to-source alignments from GIZA++ (Och and Ney, 2003).[4] To obtain an approximation of the posterior probability in each direction, it uses the m-best alignments $a^{(1)} \dots a^{(m)}$, where $a_i^{(k)} = j$ indicates that the ith source word aligns to the jth target word in the kth best alignment, and $\Pr(a^{(k)})$ is the probability of the kth best alignment according to GIZA++. We then approximate the posterior probability as follows.

$$\Pr(a_i = j) = \frac{\sum_{k=1}^{m} \Pr(a^{(k)}) \mathbb{I}[a_i^{(k)} = j]}{\sum_{k=1}^{m} \Pr(a^{(k)})}$$

4 Demonstration Script

AMRICA makes AMRs accessible for data exploration. We will demonstrate all three capabilities outlined above, allowing participants to visually explore AMRs using graphics much like those in Figures 2, 3, and 4, which were produced by AMRICA. We will then demonstrate how AMRICA can be used to generate a preliminary alignment for bitext

[3] Another option for aligning AMR graphs to sentences is the statistical aligner of Pourdamghani et al. (2014)

[4] In experiments, this method was more reliable than using either alignment alone.

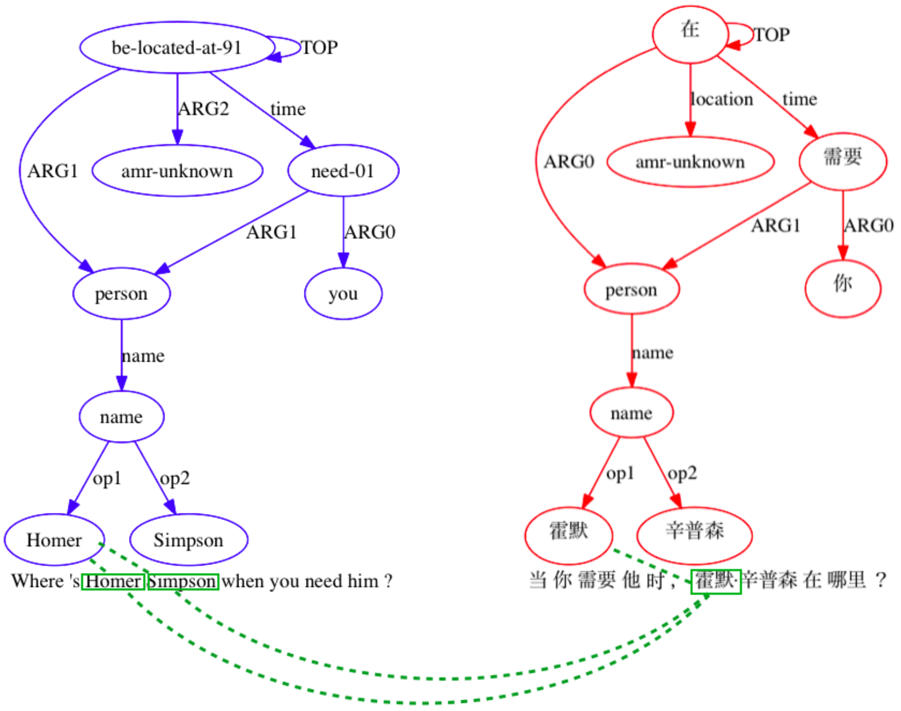

Figure 5: Cross-lingual AMR example from Nianwen Xue et al. (2014). The node-to-node alignment of the highlighted nodes is computed using the node-to-word, word-to-word, and node-to-word alignments indicated by green dashed lines.

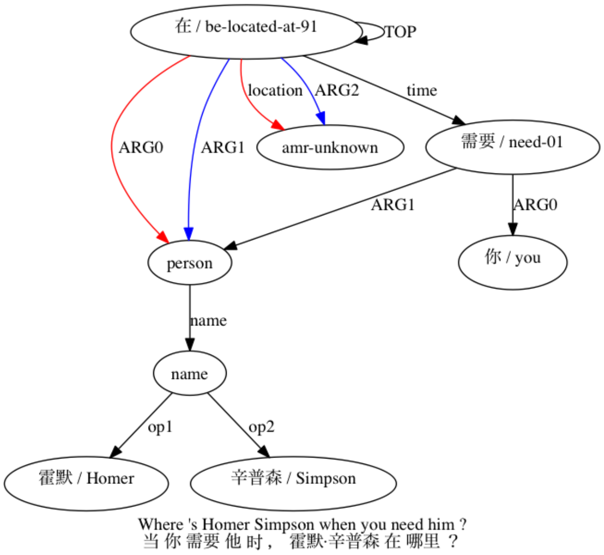

Figure 4: AMRICA visualization of the example in Figure 5. Chinese concept labels are first in shared nodes.

AMRs, which can be corrected by hand to provide training data or a gold standard alignment.

Information to get started with AMRICA is available in the README for our publicly available code.

Acknowledgments

This research was supported in part by the National Science Foundation (USA) under awards 1349902 and 0530118. We thank the organizers of the 2014 Frederick Jelinek Memorial Workshop and the members of the workshop team on Cross-Lingual Abstract Meaning Representations (CLAMR), who tested AMRICA and provided vital feedback.

References

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2013. Abstract meaning representation for sembanking. In *Proc. of the 7th Linguistic*

Annotation Workshop and Interoperability with Discourse.

S. Cai and K. Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proc. of ACL*.

J. Flanigan, S. Thomson, C. Dyer, J. Carbonell, and N. A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proc. of ACL*.

Nianwen Xue, Ondrej Bojar, Jan Hajic, Martha Palmer, Zdenka Uresova, and Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of English AMRs to Chinese and Czech. In *Proc. of LREC*.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, Mar.

N. Pourdamghani, Y. Gao, U. Hermjakob, and K. Knight. 2014. Aligning english strings with abstract meaning representation graphs.

Linguistic Data Consortium. 2013. DEFT phase 1 AMR annotation R3 LDC2013E117.