A Structured Syntax-Semantics Interface for English-AMR Alignment

Citation for published version:

Digital Object Identifier (DOI):
10.18653/v1/N18-1106

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Abstract

Abstract Meaning Representation (AMR) annotations are often assumed to closely mirror dependency syntax, but AMR explicitly does not require this, and the assumption has never been tested. To test it, we devise an expressive framework to align AMR graphs to dependency graphs, which we use to annotate 200 AMRs. Our annotation explains how 97% of AMR edges are evoked by words or syntax. Previously existing AMR alignment frameworks did not allow for mapping AMR onto syntax, and as a consequence they explained at most 23%. While we find that there are indeed many cases where AMR annotations closely mirror syntax, there are also pervasive differences. We use our annotations to test a baseline AMR-to-syntax aligner, finding that this task is more difficult than AMR-to-string alignment; and to pinpoint errors in an AMR parser. We make our data and code freely available for further research on AMR parsing and generation, and the relationship of AMR to syntax.

1 Introduction

Abstract Meaning Representation (AMR; Banarescu et al., 2013) is a popular framework for annotating whole sentence meaning. An AMR annotation is a directed acyclic graph in which nodes represent entities and events, and edges represent relations between them, as on the right in figure 1.

AMR annotations include no explicit mapping between elements of an AMR and the corresponding elements of the sentence that evoke them, and this presents a challenge to developers of machine learning systems that parse sentences to AMR or generate sentences from AMR, since they must first infer this mapping in the training data (e.g.

\footnote{For clarity of presentation, we have constructed the sentences and AMRs shown in figures—except for figure 3, which is a simplified version of a sentence in the corpus.}
of lies and not by any particular word.

Although it seems sensible to assume that all of the nodes and edges of an AMR are evoked by the words and syntax of a sentence, the existing alignment schemes do not allow for expressing that relationship. We therefore propose a framework expressive enough to align AMR to syntax (§2) and use it to align a corpus of 200 AMRs to dependency parses. We analyse our corpus and show that the addition of syntactic alignments allows us account for 97% of the AMR content.

Syntactic-semantic mappings are often assumed by AMR parsing models (e.g. Wang et al., 2015; Artzi et al., 2015; Damonte et al., 2017), which is understandable since these mappings are well-studied in linguistic theory. But AMR explicitly avoids theoretical commitment to a syntax-semantics mapping: Banarescu et al. (2013) state that “AMR is agnostic about how we might want to derive meanings from strings.” If we are going to build such an assumption into our models, we should test it empirically, which we can do by analysing our corpus. We observe some pervasive structural differences between AMR and dependency syntax (§3), despite the fact that majority of AMR edges map easily onto dependency edges.

Since syntactic alignment can largely explain AMRs, we also develop a baseline rule-based aligner for it, and show that this new task is much more difficult than lexical alignment (§4). We also show how our data can be used to analyze errors made by an AMR parser (§5). We make our annotated data and aligner freely available for further research.

2 Aligning AMR to dependency syntax

Our syntactic representation is dependency grammar, which represents the sentence as a rooted, directed graph where nodes are words and edges are grammatical relations between them (Kruijff, 2006). We use Universal Dependencies (UD), a cross-lingual dependency annotation scheme, as implemented in Stanford CoreNLP (Manning et al., 2014). Within the UD framework, we use enhanced dependencies (Schuster and Manning, 2016), in which dependents can have more than one head, resulting in dependency graphs (DGs).

Our alignment guidelines generalize ideas present in the existing frameworks. We want to allow many-to-many alignments, which we motivate by the observation that some phenomena cause AMR graph to have one structure expressing the same information as multiple DG structures, and vice versa. For instance, in figure 2 the AMR subgraph representing Cruella de Vil aligns to two subgraphs in the DG graph because of pronominal coreference. In the other direction, in figure 3 the capabilities node aligns to both capable nodes in the AMR, which is a result of the AMR treating conjoined adjectival modifiers as a case of ellipsis. The alignments we propose hold between subgraphs of any size. By aligning subgraphs we gain expressiveness needed to point out correspondences between semantic and syntactic structure. If AMR and DG were very similar in how they represent information, such correspondences would probably hold between subgraphs consisting of a single edge, as in figure 1 cat\text{nmod:poss} \sim my \sim cat\text{nmod:poss} \sim I. However, AMR by design abstracts away from syntax and it should not be assumed that all mappings will be so clean. For example, in the same figure lies\text{nmod-in} \sim sun\text{case} \sim lies\text{pos} \sim sun. Moreover, AMR represents the meaning of particular words or phrases with elaborate structures, the result of which might be that the same information is expressed by a single word and a whole AMR subgraph, as in figure 3 where AMR represents general as person\text{ARG0} \text{have-org-role} \text{general}.

2.1 Overview

An alignment is a link between subgraphs in an AMR and a DG which represent equivalent information. Given a sentence’s DG and AMR we define an alignment as a mapping between an AMR subgraph and a DG subgraph. Lexical alignments (§2.2) hold between pairs of nodes, and nodes from either graph may participate in multiple lexical alignments. Structural alignments (§2.3) hold between pairs of connected subgraphs where at least one of the subgraphs contains an edge.

In the following two sections we discuss the types of alignments that our framework allows. More detailed guidelines regarding how to align particular linguistic constructions can be found in appendix A.

3We chose UD because it emphasises shallow and semantically motivated annotation, by the virtue of which it can be expected to align relatively straightforwardly to a semantic annotation such as AMR. Aligning AMR with different versions of dependency grammar (e.g. Prague) or different syntactic frameworks (e.g. CCG, TAG) would be an interesting extension of our work.
Figure 2: “In the story, evildoer Cruella de Vil makes no attempt to conceal her glee and greed.” For legibility in this and following figures only a subset of the structural alignments are shown.

2.2 Lexical alignments

A lexical alignment should hold between a word and an AMR concept if the latter is judged to express the lexical meaning of the former. Node labels usually reflect their lexically aligned word or its lemma, including derivational morphology (e.g. thirsty ~ thirst-01). Thus, string similarity is a useful heuristic for lexical alignment.4

Most AMR nodes align lexically to a single word. Cases of one-to-many alignments include coreference, when an entity is mentioned multiple times in the sentence, and multiword expressions such as a verb-particle constructions (pay off ~ pay-off-02) or fixed grammatical expressions (instead of ~ instead-of-91). Occasionally an AMR node does not lexically align to any DG node. This is true for constants indicating sentence mood such as imperative, implicit uses of and to group list items, inferred concepts nodes such as entity types, name in named entities, and -91 frames like have-org-role-91.

Most words are lexically aligned to a single AMR node, if they are aligned at all. A word may align to multiple AMR nodes if it is duplicated in the AMR due to ellipsis or distributive coordination (capabilities aligns to c2 / capable and c3 / capable in figure 3), or if it is morphologically decomposed in the AMR (evildoer aligns to evil and do-02 in figure 2). Many words are not lexically aligned to any AMR node, including punctuation tokens, articles, copulas, nonmodal auxiliaries, expletive subjects, infinitival to, complementizer that, and relative pronouns.

2.3 Structural alignments

Structural alignments primarily reflect compositional grammatical constructions, be they syntactic or morphological. Note that the structural alignments build upon the lexical ones. Structural alignments hold between two subgraphs, at least one of which is larger than a single node. If a subgraph includes any edges, it automatically includes nodes adjacent to those edges. Structural alignments need not be disjoint: an edge can appear in two or more distinct alignments. Nodes and edges in both AMR and DG may be unaligned.

2.3.1 Constraints on structural alignments

The ability to align subgraphs to subgraphs gives considerable flexibility in how the annotation task can be interpreted. We establish the following principles to guide the specification of alignment:

Connectedness Principle. In an alignment $d \sim a$, $d$ must be a connected subgraph of the DG, and $a$ must be a connected subgraph of the AMR.

---

4Exceptions include: pronouns with noun antecedents in the sentence; the - indicating negative polarity, which lexically aligns to no, not, and negative prefixes; modal auxiliaries, e.g., can ~ possible; normalized dates and values such as February ~ 2 in a date-entity; and amr-unknown, which aligns to wh-words.
Minimality Principle. If two alignments, $d \sim a$ and $d' \sim a'$, have no dependency or AMR edges in common, then their union $a \cup d' \sim d \cup d'$ is redundant, even if it is valid. Individual alignments should be as small as possible; we believe compositionality is best captured by keeping structures minimal. Therefore, in figure 1 there is no alignment between subgraphs spanning $My$, $cat$, $lies$ and $i$, $cat$, $lie$. Such subgraphs do express equivalent information, but the alignment between them decomposes neatly into smaller alignments and we record only those.

Subsumption Principle. This principle expresses the fact that our alignments are hierarchical. Structural alignments need to be consistent with lexical alignments: for subgraph $a$ to be aligned to subgraph $d$, all nodes lexically aligned to nodes in $a$ must be included in $d$, and vice versa. Moreover, structural alignments need to be consistent with other structural alignments. A structural alignment $d \sim a$ is valid only if, for every connected AMR subgraph $a \subseteq a$ which is aligned to a DG subgraph, $a \sim a'$, we also have that $d'$ is a subgraph of $d$—and vice versa for every $d \subseteq d$.

Moreover, if $a$ contains a node $n$ which is not lexically aligned but which is part of a structurally aligned subgraph $a \subseteq a$ such that $a \sim a'$, it needs to be the case that $a \subseteq a$ and $d \subseteq d$. (And vice versa.) For example, $conceal_{nsubj-xsubj}$ Cruella $\sim$ conceal $\rightarrow$ Cruella is not a valid alignment, because the AMR side contains nodes person and name, which are not lexically aligned but which are both parts of a structural alignment marked in blue.

Coordination Principle. If an alignment contains a dependency edge between two conjuncts, or between a conjunct and a coordinating conjunction, then it must also include all conjuncts and the conjunction. This preserves the integrity of coordinate structures in alignments. For example, in figure 2 there is no alignment $glee\rightarrow and \sim$ and $\rightarrow glee$; only the larger structures which includes the greed nodes are aligned.

Named Entity Principle. Any structural alignment containing an AMR name node or any of the strings under it must contain the full subgraph rooted in the name plus the node above it specifying the entity type. This means that for example, in figure 2 there is no alignment $conceal_{nsubj-xsubj}$ Cruella $\sim$ Cruella $\rightarrow$ Cruella. Such an alignment would also be stopped by the Subsumption Principle provided that the blue alignment of the whole name was present. The Named Entity Principle is superfluous, but is provided to explicitly describe the treatment of such constructions.

2.3.2 Typology of structural alignments

The smallest structure which can participate in a structural alignment is a single node, provided that it is aligned to a subgraph containing at least one edge. A DG node may align to an AMR subgraph if the word is morphologically decomposed or otherwise analyzed in the AMR (e.g. in figure 2, evil-doer $\sim$ person $\rightarrow$ do-02 $\rightarrow$ thing $\rightarrow$ evil).
Examples of DG structures whose meaning is expressed in a single AMR node include light verb constructions, phrasal verbs, and various other multiword expressions (e.g. in figure 2, \textit{makes} \( \text{do} \) \textit{attempt} \( \sim \text{attempt-01} \)).

Conceptually the simplest case of structural alignment is one edge to one edge, as in the blue and green alignments in figure 1. For such an alignment to be possible, two requirements must be satisfied: nodes which are endpoints of those edges need to be aligned one-to-one; and the AMR relation and the syntactic dependency must map cleanly in terms of the relationship they express.

A one edge to multiple edges alignment arises when either of those requirements is not met. To see what happens in absence of one-to-one endpoint alignments let’s look at the relation between confident and general in figure 3. The DG \textit{general} node is aligned to an AMR subgraph: \textit{general} \( \sim \text{person} \) \( \text{ARG0-of} \) \( \text{ARG2} \) \( \sim \text{have-org-role} \) \( \sim \text{general} \). All alignments which involve the \textit{general} node on the DG side need to include its aligned subgraph on the AMR side. It necessarily follows that the AMR subgraphs in those alignments will contain more edges that the DG ones; in this case the yellow subgraph in DG has 1 edge, and in AMR 3 edges. As for the second requirement, it is possible for one graph to use multiple edges to express a relationship when the other graph needs only one. This is the case for \textit{lie} \( \sim \text{in mod-in} \) \( \sim \text{sun} \) \( \sim \text{case} \) \( \sim \text{lie in location} \) \( \sim \text{sun} \) in figure 1. An example which combines both the node- and edge-related issues is marked in red in figure 2.

Finally, we also allow for many edges to many edges alignments. This may seem counterintuitive considering the assumption that we want to capture a single relation, but the we need to treat a subgraph as an endpoint of the edge both in DG and AMR. For instance, let’s say we want to construct an alignment to state the fact that in figure 2 \textit{nsubj-xsubj} and \textit{ARG0} express the same relationship, one which holds between Cruella de Vil and concealing. One of the entities involved in that relationship, Cruella, is represented by a 2-edge subgraph in DG and a 4-edge one in AMR, and as a consequence the alignment we are after will be between subgraphs of consisting of 3 and 5 edges. A more difficult case of many edges to many edges alignment arises when relationships between nodes are expressed so differently in the DG and AMR that given an edge in one graph it is not possible to find in the other graph a subgraph that would convey the same information without also including some other information. The simplest example to illustrate such a case arises in conjunction constructions. In figure 2 the \textit{conj-and} dependency between \textit{glee} and \textit{greed} has no counterpart in the AMR. There is no edge between AMR nodes aligned to those words, and the smallest AMR subgraph which contains them also contains and, which is itself lexically aligned. We cannot align \textit{glee} \( \sim \text{op1} \) \( \sim \text{op2} \) \( \sim \text{and} \) \( \sim \text{greed} \) because of the rule that all lexically aligned nodes in one subgraph must be aligned to nodes in the other subgraph. Therefore we need to extend the DG side to \textit{and} \( \sim \text{cc} \) \( \sim \text{glee} \) \( \sim \text{greed} \).

### 3 Manually aligned corpus

We annotated a corpus of 200 AMR-sentence pairs (3813 aligned structures)\(^5\) using the guidelines of §2 and appendix A.

#### Data selection

To create the corpus we drew a total of 200 AMR-sentence pairs: 135 from the training split of the AMR Annotation Release 1.0 (Knight et al., 2014), 55 from the training split of The Little Prince Corpus v1.6,\(^6\) and 10 sentences from the Adam part of the CHILDES Brown corpus (Brown, 1973), for which AMRs were produced by an experienced annotator. Seventy items were selected to illustrate particular linguistic phenomena.\(^7\) The remaining 130 were selected at random.

#### Preprocessing

Dependency parses were obtained using Stanford CoreNLP neural network parser\(^8\) (Chen and Manning, 2014) and manually corrected. The final parses conform to the enhanced UD guidelines,\(^9\) except they lack enhance-

\(^5\)The other two available AMR-to-sentence alignment corpora (see §4.2) have the same sentence count

\(^6\)https://amr.isi.edu/download/amr-bank-struct-v1.6.txt

\(^7\)Namely: relative clauses, reflexive and non-reflexive pronominal anaphora, subject and object control, raising, exceptional case marking, coordination, wh-questions, do-support questions, ellipsis, expletives, modal verbs, light verbs, comparison constructions, and quantification.

\(^8\)The corpus is annotated with UD v1; a release of the dataset converted to UD v2 is planned for the future. We used the pretrained dependency parsing model provided in CoreNLP with depparse.extradependencies set to \textit{MAXIMAL}, and used collapsed CCprocessed dependencies.

\(^9\)http://universaldependencies.org/u/overview/enhanced-syntax.html
ments for ellipsis.

**Inter-annotator agreement.** The corpus was created by one annotator. To assess inter-annotator agreement, a second annotator deeply familiar with UD and AMR annotated a random sample of sentences accounting for 10% of alignments in the corpus. The overall inter-annotator $F_1$-score was 88%, with 96% agreement on lexical alignments and 80% on structural alignments. We take this as an indication that our richly structured alignment framework as laid out in §2 is reasonably well-defined for annotators.

### 3.1 Coverage

To assess our attempt to explain as much of the AMR as possible, we computed the proportion of AMR nodes and edges that participate in at least one alignment. Overall, 99.3% of nodes and 97.2% of edges in AMRs are aligned. We found that 81.5% of AMR graphs have full coverage, 18.5% have at least one unaligned edge, and 7.5% have one unaligned node (none had more than one; all unaligned nodes express mood or discourse-related information: interrogative, and, and say). We conclude that our framework can show that nearly all information in an AMR is evoked by lexical items or syntactic structure.

We expected coverage of DG to be lower because punctuation and many function words are unaligned in our guidelines (§2.2). Indeed, only 71.4% of words and 65.2% of dependency edges are aligned.

### 3.2 Syntactic-semantic similarity

The similarity of AMR to syntax in examples like figure 1 invites the assumption of a close mapping, which often seems to be made in AMR parsers (Wang et al., 2015; Artzi et al., 2015; Misra and Artzi, 2016; Damonte et al., 2017) and aligners (Chu and Kurohashi, 2016; Chen and Palmer, 2017). Such an attitude reflects decades of work in the syntax-semantics interface (Partee, 2014) and the utility of dependency syntax for other forms of semantics (e.g., Oepen et al., 2014; Reddy et al., 2016; Stanovsky et al., 2016; White et al., 2016; Zhang et al., 2017; Hershcovich et al., 2017). However, this assumption has not been empirically tested, and as Bender et al. (2015) observe, it is an assumption not guaranteed by the AMR annotation style. Having aligned a corpus of AMR-DG pairs, we are in a position to provide empirical evidence.

Are AMRs and dependency graphs structurally similar? We approach the question by analyzing the sizes of subgraphs used to align the two representations of the sentence.

We define the size of a subgraph as the number of edges it contains. If a structure consists of a single node, we say its size is 0. The **configuration** of an alignment is then the pair of sizes for its AMR and DG sides; for example, an alignment with 1 AMR edge and 2 DG edges has configuration 1:2. We call an alignment configuration simple if at least one of the subgraphs is a single edge, indicating that there is a single relation which the alignment captures. Complex configurations cover multiple relations. By principle of minimality we infer that some structural difference between the graphs prevented those relations from aligning individually.

One measure of similarity between AMR and DG graphs is the configuration of the most complex subgraph alignment between them. Configuration $a:b$ is higher than $c:d$ if $a+b > c+d$. However, all configurations involving 0 are lower than those which do not. A maximum of 1:1 means the graphs have only node-to-node, node-to-edges, and edge-to-edge alignments, rendering the graphs isomorphic (ignoring edge directions and unaligned nodes). In general if the maximum alignment configuration is a simple one, the graphs could be made isomorphic by collapsing the larger side of the alignment (e.g. figure 2 the AMR side of the alignment `*evildoer ~ person` could be collapsed into a node.

In contrast, complex configurations imply serious structural dissimilarity, as in figure 3, where the cyan alignment has configuration 4:4.

The numbers in table 1 show that ≈33% of the

![](image.png)

**Table 1**: Number of sentences whose highest alignment configurations is max config.

<table>
<thead>
<tr>
<th></th>
<th>max config.</th>
<th># avg. words</th>
<th>max config.</th>
<th># avg. words</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:1</td>
<td>18</td>
<td>8.7</td>
<td>2:2</td>
<td>21</td>
</tr>
<tr>
<td>1:2</td>
<td>16</td>
<td>13.1</td>
<td>2:3</td>
<td>14</td>
</tr>
<tr>
<td>3:1</td>
<td>12</td>
<td>13.4</td>
<td>3:2</td>
<td>13</td>
</tr>
<tr>
<td>2:0</td>
<td>6</td>
<td>5.8</td>
<td>3:4</td>
<td>12</td>
</tr>
<tr>
<td>1:3</td>
<td>5</td>
<td>13.2</td>
<td>3:3</td>
<td>10</td>
</tr>
<tr>
<td>other</td>
<td>9</td>
<td>15.2</td>
<td>other</td>
<td>64</td>
</tr>
<tr>
<td><strong>total:</strong></td>
<td><strong>66</strong></td>
<td><strong>11.6</strong></td>
<td><strong>134</strong></td>
<td><strong>18.0</strong></td>
</tr>
</tbody>
</table>

10 In particular, Chen and Palmer (2017) align dependency paths to AMR edges. However, their evaluation only considers node-to-node alignment, and their code and data are not available for comparison at the time of this writing.
sentences are simple.

Table 2 provides a detailed breakdown of alignment configurations in the corpus. Phenomena which often trigger complex configurations include coordination, named entities, semantically decomposed words, attachment of negation, and preposition-based concepts encoding location, time, and quantity.\footnote{An AMR concept evoked by a preposition usually dominates the structure \texttt{after date-entity decade nineties}, which is at odds with UD’s prepositions-as-case-markers policy \texttt{nineties \rightarrow after}.}

We observe, comparing tables 1 and 2, that while simple configurations are most frequent in the corpus, the majority of sentences have at least one alignment which is complex. It should not be assumed that AMR and DG representations of a sentence are, or could trivially be made to be, isomorphic. It is worth noting that our analysis suggests that DG and AMR could be made more similar by applying simple transformations targeting problematic constructions like coordination and named entities.

### 4 Evaluation of automatic aligners

We use our annotations to measure the accuracy of AMR aligners on specific phenomena that were inexpressible in previous annotation schemes. Our experiments evaluate the JAMR heuristic aligner (Flanigan et al., 2014), the ISI statistical aligner (Pourdamghani et al., 2014), and a heuristic rule-based aligner that we developed specifically for structural alignment.

#### 4.1 Rule-based aligner

Our aligner operates in two passes: one for lexical alignment and one for structural alignment.

**Lexical alignment algorithm.** AMR concepts are cognate with English words, so we align them by lexical similarity. This algorithm does not make use of the DG. Before alignment, we remove sense identifiers on AMR node labels, and lemmatize DG node labels. Then for every node $a$ in the AMR and $d$ in the DG we decide whether to align them using rules based on string similarity scores, morpheme overlap, a list of morphologically negative words and negations, and a list of light verbs. Further details of the algorithm are described in appendix B.

Note that if a word type is repeated in a sentence, each repetition is aligned to the same AMR nodes under the above rules.

**Structural alignment algorithm.** We align subgraphs using the procedure below, first from AMR to DG, then from DG to AMR. For clarity, the explanation refers to the first case.

**Local phase.** For every AMR edge \(e_a\) whose endpoints are lexically aligned nodes \(a_1\) (aligned to \(d_1\)) and \(a_2\) (aligned to \(d_2\)), we attempt to align minimal and connected AMR and dependency subgraphs, \(a’\) and \(d’\):

1. If there is a DG edge \(e_d\) whose endpoints are \(d_1\) and \(d_2\), then \(a’ \leftarrow e_a\) and \(d’ \leftarrow e_d\).

2. Otherwise, let \(\pi_d\) be the shortest undirected path between \(d_1\) and \(d_2\). If all lexically aligned nodes in \(\pi_d\) are aligned to \(a_1\) or \(a_2\), then \(a’ \leftarrow e_a\) and \(d’ \leftarrow e_d\).

3. Otherwise, let \(a’’\) be the smallest subgraph covering all AMR nodes that are lexically aligned to nodes in \(\pi_d\). If all the nodes in \(a’’\) are aligned only to nodes in \(\pi_d\), then \(a’ \leftarrow a’’\) and \(d’ \leftarrow \pi_d\).

4. Otherwise, the attempt is abandoned.

5. Finally, if the top node of \(a’\) has a parent node labeled with an entity type concepts, extend \(a’\) to include the parent. (This step is performed only in the AMR-to-DG step.)

**Global phase.** The local phase might produce alignments that violate the Subsumption Principle (§2.3.1), so we filter them out heuristically. For every pair of structural alignments, \(\pi_d \sim \pi_a\) and \(\pi_d’ \sim \pi_a’\) where \(\pi_a\) overlaps with \(\pi_d’\), or \(\pi_d\) with \(\pi_d’\), if the region of overlap is not itself an aligned subgraph, we prune both alignments.\footnote{This could be order-dependent since the removal of one alignment could trigger the removal of others, but our aligner}

### Table 2: Frequency of alignment configurations for named entities, coordination, semantically decomposed words, quantities and dates, other phenomena, and overall corpus frequencies.

<table>
<thead>
<tr>
<th>named entities</th>
<th>coordination</th>
<th>semantic decomposition</th>
<th>quantities &amp; dates</th>
<th>other</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:0</td>
<td>112</td>
<td>2:2</td>
<td>1:0</td>
<td>32</td>
<td>2:1</td>
</tr>
<tr>
<td>3:1</td>
<td>44</td>
<td>3:4</td>
<td>2:0</td>
<td>14</td>
<td>3:0</td>
</tr>
<tr>
<td>4:2</td>
<td>7</td>
<td>3:3</td>
<td>2:1</td>
<td>11</td>
<td>1:0</td>
</tr>
<tr>
<td>1:1</td>
<td>6</td>
<td>4:3</td>
<td>5:4</td>
<td>6</td>
<td>3:2</td>
</tr>
<tr>
<td>5:2</td>
<td>4</td>
<td>3:2</td>
<td>5:3</td>
<td>6</td>
<td>8:2</td>
</tr>
<tr>
<td>other</td>
<td>20</td>
<td>other</td>
<td>other</td>
<td>50</td>
<td>other</td>
</tr>
<tr>
<td>total</td>
<td>193</td>
<td>117</td>
<td>84</td>
<td>28</td>
<td>3385</td>
</tr>
</tbody>
</table>
We evaluate JAMR, ISI, and our aligner on two AMR nodes and 30.0% of AMR edges unaligned, words and occasional lack of similarity between Table 4.

Table 3: Lexical alignment (precision, recall, F1-score). Our lexical alignment algorithm does not use syntax.

<table>
<thead>
<tr>
<th>dataset</th>
<th>prec., rec., F1</th>
<th>prec., rec., F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMR</td>
<td>89 85 87 88 77 82 55 81 65</td>
<td>71 68 70 96 85 90 47 67 55</td>
</tr>
<tr>
<td>ISI</td>
<td>86 63 72 95 66 78 92 85 88</td>
<td>65 44 61 48 45 50</td>
</tr>
</tbody>
</table>

Table 4: Structural alignment (§4.1) scores, with different sources of input lexical alignments. Scores are shown for gold standard and automatic UD trees.

<table>
<thead>
<tr>
<th>UD structure</th>
<th>missed</th>
<th>mislabeled</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsubj</td>
<td>103 (40%)</td>
<td>14 (6%)</td>
</tr>
<tr>
<td>nmod + case</td>
<td>74 (44%)</td>
<td>26 (16%)</td>
</tr>
<tr>
<td>compound</td>
<td>55 (41%)</td>
<td>7 (5%)</td>
</tr>
<tr>
<td>amod</td>
<td>40 (26%)</td>
<td>9 (6%)</td>
</tr>
<tr>
<td>dobj</td>
<td>40 (33%)</td>
<td>6 (5%)</td>
</tr>
<tr>
<td>advmod</td>
<td>30 (39%)</td>
<td>7 (9%)</td>
</tr>
<tr>
<td>cc + conj</td>
<td>29 (57%)</td>
<td>4 (8%)</td>
</tr>
<tr>
<td>nmod</td>
<td>21 (60%)</td>
<td>1 (3%)</td>
</tr>
</tbody>
</table>

Table 5: Error analysis of the AMR parser of Damonte et al. (2017). Frequency of dependency structures aligned to AMR edges which the automatic AMR parser missed altogether or mislabeled; absolute count (% of all such aligned structures in the corpus).

F-score for the gold standard lexical alignments and DGs condition suggests that substantial improvements to our structural alignment algorithm are possible. The two most common reasons for low recall were missing one of the conjuncts in a coordinate structure and aligning structures that violate the principle of minimality. Our corpus gives alignments between AMRs and gold standard dependency parses. To see how much performance degrades when such parses are not available we also evaluate on automatic parses.15

5 Improving error analysis for AMR parsers

Our corpus of manually aligned AMRs can be used to identify linguistic constructions which cause problems for an AMR parser. We parsed the sentences from our corpus with the parser of Damonte et al. (2017).16 We map the nodes of the resulting automatic AMRs to the gold AMRs using the smatch evaluation tool (Cai and Knight, 2013), and on the basis of this mapping identify those nodes and edges of the gold AMRs which are missing or are mislabeled in the automatic AMRs.

We then measured the number and rate of erroneous AMR fragments associated with each UD relation or construction (table 5). The largest proportion of recall errors were for fragments associated with the subject relation, prepositional phrases, and nominal compounds. Focusing on the subject relation, we can further say that 69% of the missing or mislabeled edges have the gold label ARG0, 19% ARG1, and the rest are distributed amongst domain, ARG2, purpose and mod. Inspecting the errors we see that phenomena underlying them include pronominal coreference, sharing arguments...
between conjoined predicates, auxiliary verb constructions, and control and raising.\footnote{The missing edge counts include gold edges for which the parser failed to produce one or both endpoints.}

Our corpus facilitates fine-grained error analysis of AMR parsers with respect to individual syntactic constructions. We release the code for the above analysis in order to encourage syntactically-informed comparison and improvement of systems.

6 Conclusion

We have presented a new framework and corpus for aligning AMRs to dependency syntax. Our data and analysis show that the vast majority of the semantics in AMR graphs can be mapped to the lexical and syntactic structure of a sentence, though current alignment systems do not fully capture this correspondence. The syntax–semantics correspondences are often structurally divergent (non-isomorphic). Simple algorithms for lexical and structural alignment establish baselines for the new alignment task; we expect statistical models will be brought to bear on this task in future work. Our framework also facilitates syntactically-based analysis of AMR parsers. We will release our data and code for the benefit of the research community.

References


A Details of alignment guidelines

A.1 Lexical alignments

Names In proper names, individual strings denoting words in the name are lexically aligned, but the entity as a whole is structurally aligned.

Entity types If the entity type is based on a common noun which occurs in the sentence, it is lexically aligned: e.g., Jon, a clumsy man, has a cat would involve the alignment man ~ man. Most often, however an entity type is not explicitly mentioned in the sentence and is taken from AMR’s ontology of entity types: http://www.isi.edu/~ulf/amr/lib/ne-types.html, in which case it will not be lexically aligned.

Case marking and prepositions The possessive marker ‘s and many prepositions participate in structural but not lexical alignments because they are inherently relational. However, we align a preposition if it carries sufficient lexical content to be included as an AMR node (e.g., the AMR for The cat is under the table would include u/under :op1 t/table).

Wh-questions The special concept amr-unknown aligns lexically to the wh-word whose referent is questioned. For multiword wh-expressions like how much, the expression is aligned structurally (not lexically) to amr-unknown.

Sentence mood In AMR, non-wh questions are indicated by interrogative, imperatives by imperative, and exclamations/interjections by expressive. UD parses do not encode sentence mood, which can be conveyed by non-canonical word order (subject-auxiliary inversion for questions) or argument omission (subject omission for imperatives), rather than the presence of certain relations or words. Sometimes the sentence includes an appropriate alignment point, e.g. complementizers whether and if for interrogative, allowing for a lexical alignment. More often the parse has no obvious alignment point, and the constant interrogative, imperative, or expressive is left unaligned.18

18Among the UD community there has been discussion of possibly adding sentence-level marking of mood (https://github.com/UniversalDependencies/docs/issues/458), which could provide a convenient alignment point.
A.2 Structural alignments

Copula In UD, copulas are treated as modifiers of a predicate nominal or adjective, which is linked directly to the subject of the sentence via an nsubj dependency. We do not align copulas or the cop edge. Thus, in figure 3, there is a structural alignment between general — confident and the AMR subgraph connecting the lexically aligned nodes.

Control The subject of the control verb and the controlled predicate are connected by the nsubj—xsubj edge, which can be structurally aligned with the corresponding AMR argument relation, as in e.g. figure 2.

Relative clause In enhanced UD the noun governing a relative clause and the embedded predicate are linked by edges in both directions: a “surface syntax” acl-relcl edge headed by the noun, and a “deep syntax” edge such as nsubj, dobj, iobj, or nmod headed by the embedded predicate. Each participates in a structural alignment with the corresponding AMR subgraph. The relative pronoun is left unaligned.

Coordination Coordination does not naturally lend itself to analysis with dependencies, and different dependency grammar traditions offer different approaches (Nivre, 2005; Mareček et al., 2013). UD follows the Stanford style, where the first conjunct serves as the head of the remaining conjuncts, and the conjunction is a dependent of one of the conjuncts. In AMR the conjunction heads all the conjuncts (Prague style). In light of this mismatch, we use a subgraph alignment to group the conjunction with its conjuncts on each side. A simple example is illustrated in figure 2. A quirk of UD’s approach to coordination is that it does not distinguish modifiers of the first conjunct from modifiers of the coordinate structure as a whole. The basic UD parse of her glee and greed is therefore ambiguous. We rely on an extra edge in the enhanced parse between her and greed to establish an alignment for the AMR edge greed — person.

The coordination in figure 3 is more complex: the coordinated modifier defense and security distributes over capabilities (i.e., there are two kinds of capabilities). In the enhanced parse, defense and security are both attached as modifiers of capabilities. This is expressed semantically via duplicative AMR nodes labeled capable, each receiving different modifiers corresponding to different conjuncts. Independent of coordination, the two capable nodes also share a common argument, nation. The three syntactic modifiers give rise to three subgraph alignments, and the subgraph alignment covering the coordinate structure (cyan in the figure) enwraps two of these. Ellipsis constructions can also trigger node duplication in AMR, requiring similar structural alignments.

Named entities AMR annotates each named entity with a node representing the name, linked to the strings of the name and headed by an entity type. This full structure is aligned to the full name in the dependency parse.

Coreferent mentions Coreference often causes an AMR structure to align to multiple DG subgraphs. For example, in figure 2, both the pronoun her and the name align to the AMR subgraph representing the entity. This mechanism suffices to represent coreference between mentions in the sentence.

Light verbs Light verbs have no lexical alignment, but a subgraph alignment covers the light verb construction as a unit (e.g. makes — attempt ~ attempt-01 in figure 2). All subgraph alignments which involve the light verb or its complement have to involve to whole unit, as shown in the alignment highlighted in red in figure 2.

Multiword expressions In verb-particle constructions or fixed grammatical expressions the AMR node lexically aligns to all words in the expression, and additionally to the DG subgraph spanning the whole expression. (e.g. pay ∼ pay-off-02, off ∼ pay-off-02, and pay compound-prt ∼ pay-off-02).

Prepositional phrase PP modifiers typically involve an extra dependency edge for the preposition attachment, as with lies nmod-in case in ∼ lie-07 location sun.

Semantically decomposed words When one word has multiple lexical alignments because of morphological decomposition, there also exists a structural alignment between that word and an AMR subgraph representing the decomposition (e.g. in figure 2, evildoer ∼ person ARGO of do-02 ARGO thing mod evil), and in figure 3 general ∼ person ARGO of have-org-role-91 ARGO general) AMR decomposes certain words by convention.
which must always be structurally aligned, such as \( \text{ago} \sim \text{before} \) and \( \text{government} \sim \text{government-organization} \).

**Date, time, and value expressions** These expressions are aligned similarly to named entities, even though the normalized constants may not exactly match the words in the sentence. For example, the DG structure \( \text{nummod}^{\text{pm}} \rightarrow 9:00 \) would be represented in the AMR as \( \text{date-entity} \rightarrow 21:00 \); tokens \( 9:00 \) and \( \text{pm} \) are treated as a multiword expression: each is lexically aligned to “21:00”. Moreover, we also align \( 9:00^{\text{nummod}} \sim 21:00 \) and \( 9:00^{\text{nummod}} \sim \text{date-entity} \rightarrow 21:00 \).

**B Aligner algorithm**

we align an AMR node \( a \) and a DG node \( d \) if any of the following conditions holds:

1. The Levenshtein distance of \( a \) and \( d \) is 15% of the length of the longer word.\(^{20}\)
2. The label of \( a \) is the morphological negation of \( d \) (e.g. \( \text{prudent} \sim \text{imprudent} \)).\(^{21}\)
3. The label of \( a \) is – (AMR’s annotation of negation) and the parent of \( a \) aligns to \( d \) via rule 2.
4. The label of \( a \) is – and \( d \) is one of \( \text{no}, \text{none}, \text{not}, \text{or never} \).
5. The label of \( a \) consists of multiple words, and label of \( d \) matches any of them under rule 1. (e.g. \( \text{sit} \sim \text{sit-down}, \text{war-torn} \sim \text{war} \)).\(^{22}\)
6. Labels of \( a \) and \( d \) likely have the same root. We determine this by segmenting each word with Morfessor (Grönroos et al., 2014) trained on Wiki data and applying rule 1 to the first morpheme of each word.
7. The label of \( d \) is one of the light verbs \( \text{take}, \text{have}, \text{make}, \text{did}, \text{get}, \text{give}; a \) aligns to one of \( d \)’s children; and \( d \) is not aligned via any of the above rules to another node in the AMR graph.

\(^{20}\)We determined this threshold on the development set.
\(^{21}\)We use a list of morphologically negated words provided by Uli Hermjakob.
\(^{22}\)This rule misaligns some AMR-specific node types, e.g. \( \text{government} \sim \text{government-organization} \).