



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

A Structured Syntax-Semantics Interface for English-AMR Alignment

Citation for published version:

Szubert, I, Lopez, A & Nathan, S 2018, A Structured Syntax-Semantics Interface for English-AMR Alignment. in *The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, New Orleans, Louisiana , pp. 1169-1180, 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, United States, 1/06/18. <https://doi.org/10.18653/v1/N18-1106>

Digital Object Identifier (DOI):

[10.18653/v1/N18-1106](https://doi.org/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.



A Structured Syntax-Semantics Interface for English-AMR Alignment

Ida Szubert Adam Lopez

School of Informatics
University of Edinburgh
Edinburgh, Scotland, UK

{Ida.Szubert, alopez}@inf.ed.ac.uk

Nathan Schneider

Linguistics and Computer Science
Georgetown University
Washington, DC, USA

nathan.schneider@georgetown.edu

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 1.¹

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.

¹For 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.

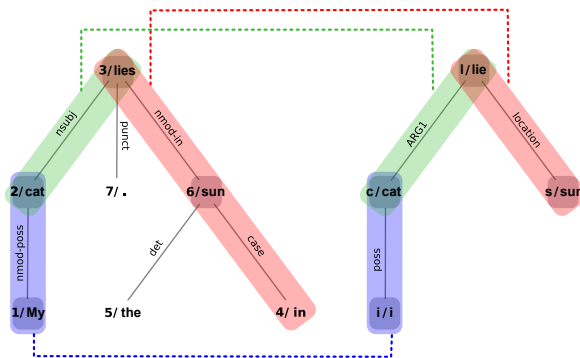


Figure 1: “My cat lies in the sun.” An alignment between the dependency parse (*left*) and AMR (*right*). Nodes participating in lexical alignments are marked with boxes, but the links between them are not displayed. Structural alignments are colour-coded and linked by dotted lines. Sense numbers for concepts that are PropBank frames are omitted for brevity.

[Flanigan et al., 2014](#); [Wang et al., 2015](#); [Artzi et al., 2015](#); [Flanigan et al., 2016](#); [Pourdamghani et al., 2016](#); [Misra and Artzi, 2016](#); [Damonte et al., 2017](#); [Peng et al., 2017](#), *inter alia*).²

This AMR alignment problem was first formalized by [Flanigan et al. \(2014\)](#), who mapped AMR nodes or connected subgraphs to words or sequences of words under the assumption of a one-to-one mapping—we call this **JAMR alignment**. [Pourdamghani et al. \(2014\)](#) then re-formalized it so that any AMR node or edge can map to any word without a one-to-one assumption—we call this **ISI alignment**. In ISI alignments, edges often align to syntactic function words: for example, `:location` aligns to *in* in figure 1. So edge alignments allow ISI to explain more of the AMR structure than JAMR, but in a limited way: only 23% of AMR edges are aligned in the ISI corpus. This may be because edges are often evoked by syntactic structure rather than words: for instance, the `:ARG1` edge in figure 1 is evoked by the fact that *cat* is the subject

²Some recent neural AMR systems require minimal or no explicit alignments ([Konstas et al., 2017](#); [van Noord and Bos, 2017](#)). But they implicitly learn them in the form of soft attention, and we believe that a clearer understanding of alignment will benefit modeling and error analysis even in these systems.

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-semantic 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).³

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

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 $cat \xrightarrow{nmod:poss} my \sim cat \xrightarrow{poss} I$. 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 $lies \xrightarrow{nmod-in} sun \xrightarrow{case} in \sim lies \xrightarrow{location} sun$. 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 $person \xrightarrow{ARG0-of} have-org-role \xrightarrow{ARG2} general$.

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.

frameworks (e.g. CCG, TAG) would be an interesting extension of our work.

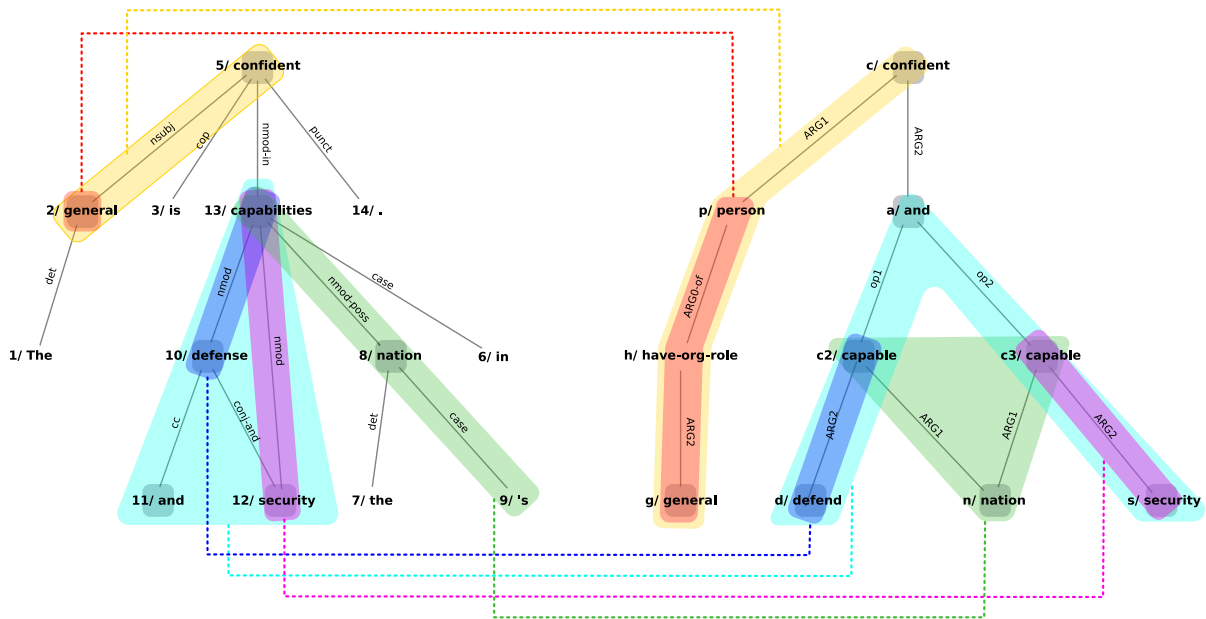


Figure 3: “The general is confident in the nation’s defense and security capabilities.”

Minimality Principle. If two alignments, $d \sim a$ and $d' \sim a'$, have no dependency or AMR edges in common, then their union $a \cup a' \sim d \cup d'$ 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 $a_< \subset a$ which is aligned to a DG subgraph, $a_< \sim d'$, we also have that d' is a subgraph of d —and vice versa for every $d_< \subset d$.

Moreover, if a contains a node n which is not lexically aligned but which is part of a structurally aligned subgraph $a_<$ such that $a_< \sim d'$, it needs to be the case that $a_< \subset a$ and $d' \subset d$. (And vice versa.) For example, $conceal \xrightarrow{nsbj-xsubj} Cruella \sim conceal \xrightarrow{ARG0} person \xrightarrow{name} name \xrightarrow{op1} Cruella$ 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 $glee \xrightarrow{cc} and \sim and \xrightarrow{op1} glee$; only the larger structures which includes the *greet* 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 $conceal \xrightarrow{nsbj-xsubj} Cruella \sim conceal \xrightarrow{ARG0} person \xrightarrow{name} name \xrightarrow{op1} Cruella$. 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, $evildoer \sim person \xrightarrow{ARG0-of} do-02 \xrightarrow{ARG1} thing \xrightarrow{mod} evil$).

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, $makes \xrightarrow{dobj} attempt \sim attempt-01$).

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 *general* node is aligned to an AMR subgraph: $general \sim person \xrightarrow{ARG0-of} have-org-role \xrightarrow{ARG2} general$. All alignments which involve the *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 than 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 $lie \xrightarrow{nmod-in} sun \xrightarrow{case} in \sim lie \xrightarrow{location} sun$ 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 mappings between relations expressed in DG and AMR, and that we want to align maximally small subgraphs. There are cases when an alignment is actually capturing a single relation, but 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 *nsubj-xsubj* and 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 *conj-and* dependency between *glee* and *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 $glee \xrightarrow{conj-and} greed \sim glee \xleftarrow{op1} and \xrightarrow{op2} greed$ 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 $and \xleftarrow{cc} glee \xrightarrow{conj-and} greed$.

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 parser⁸ (Chen and Manning, 2014) and manually corrected. The final parses conform to the enhanced UD guidelines,⁹ except they lack enhance-

⁵The 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>

⁷Namely: 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.

⁸The 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.extrdependencies` set to `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).¹⁰ 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; Hershovich et al., 2017). However, this assumption has not been empirically tested, and as Bender et al. (2015) observe, it is

¹⁰In 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.

<i>simple configurations</i>			<i>complex configurations</i>		
max config.	# sents	avg. words	max config.	# sents	avg. words
1:1	18	8.7	2:2	21	12.9
1:2	16	13.1	2:3	14	16.0
3:1	12	13.4	3:2	13	16.8
2:0	6	5.8	3:4	12	20.3
1:3	5	13.2	3:3	10	19.1
other	9	15.2	other	64	20.9
total:	66	11.6		134	18.0

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

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^{ARG0-of}→do^{ARG1}→thing^{mod}→evil 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 $\approx 33\%$ of the

named entities	coordination	semantic decomposition	quantities & dates	other	overall						
2:0	112	2:2	30	1:0	32	2:1	15	0:0	1946	0:0	1946
3:1	44	3:4	14	2:0	14	3:0	5	1:1	1002	1:1	1046
4:2	7	3:3	13	2:1	11	1:0	4	1:2	220	1:2	244
1:1	6	4:3	5	4:1	6	3:2	3	1:0	42	2:0	127
5:2	4	3:2	5	3:1	6	8:2	1	2:2	42	2:2	83
other	20	other	50	other	15	other	0	other	13	other	361
total:	193		117		84		28		3385		3807

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

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.¹¹

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

¹¹An AMR concept evoked by a preposition usually dominates the structure (after^{op1}→date-entity^{decade}→nineties), which is at odds with UD’s prepositions-as-case-markers policy (nineties^{case}→after).

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 π_d be the shortest undirected path between d_1 and d_2 . If all lexically aligned nodes in π_d are aligned to a_1 or a_2 , then $a' \leftarrow e_a$ and $d' \leftarrow \pi_d$.
3. Otherwise, let a'' be the smallest subgraph covering all AMR nodes that are lexically aligned to nodes in π_d . If all the nodes in a'' are aligned only to nodes in π_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 π_a overlaps with π'_a , or π_d with π'_d , if the region of overlap is not itself an aligned subgraph, we prune both alignments.¹²

¹²This could be order-dependent since the removal of one alignment could trigger the removal of others, but our aligner

aligner	dataset								
	our			ISI			JAMR		
our	89	85	87	88	77	82	55	81	65
ISI	71	68	70	96	85	90	47	67	55
JAMR	86	63	72	95	66	78	92	85	88

Table 3: Lexical alignment (precision, recall, F_1 -score). Our lexical alignment algorithm does not use syntax.

lexical alignments	prec., rec., F_1 using gold DGs			prec., rec., F_1 using automatic DGs		
	gold	79	73	76	70	63
our aligner	68	56	61	63	48	55
ISI	65	50	57	58	44	50
JAMR	71	41	52	61	34	44

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

4.2 Experiments

We evaluate JAMR, ISI, and our aligner on two distinct tasks.

Lexical alignment. Lexical alignment involves aligning AMR nodes to words, a task all three systems can perform. We evaluate against three datasets: our own, JAMR dataset (Flanigan et al., 2014), and ISI dataset (Pourdanghani et al., 2014).¹³ Results (table 3) suggest that this task is already well-addressed, but also that there exist marked differences between how lexical alignment is defined in each dataset and that aligners are fine-tuned to their dataset.

For our aligner, errors are due to faulty morphological analysis, duplicated words, and both accidental string similarity between AMR concepts and words and occasional lack of similarity between concepts and words that should be aligned.

Structural alignment. An important goal of our experiments is to establish baselines for the structural alignment task. While we cannot evaluate the JAMR and ISI aligners directly on this task, we can use the lexical alignments they output in place of the first pass of our aligner. The only dataset for this task is our own. The results (table 4) evaluate accuracy of structural alignments only and do not count lexical alignments.

The automatic alignments have lower coverage than the dataset: our best aligner leaves 13.3% of AMR nodes and 30.0% of AMR edges unaligned, compared to 0.07% and 2.8%¹⁴. The relatively low

does not account for this.

¹³We remove scope alignments in the JAMR dataset and edge alignments in the ISI dataset.

¹⁴The aligner also leaves 39.2% of DG nodes and 47.7% of DG edges unaligned, compared to 28.6% and 34.8% in the dataset.

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

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

¹⁵We use the CoreNLP dependency parser with settings as described in §3.

¹⁶The overall smatch score of the parser on this dataset was 0.65.

between conjoined predicates, auxiliary verb constructions, and control and raising.¹⁷

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

- Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proc. of EMNLP*.
- 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 *Proc. of 7th Linguistic Annotation Workshop and Interoperability with Discourse*.
- Emily M Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann A Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proc. of IWCS*.
- Roger Brown. 1973. *A first language: The early stages*. Harvard University Press.
- Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proc. of ACL*. Sofia, Bulgaria, pages 748–752.
- Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proc. of EMNLP*. Doha, Qatar, pages 740–750.
- Wei-Te Chen and Martha Palmer. 2017. Unsupervised AMR-dependency parse alignment. In *Proc. of EACL*.
- Chenhui Chu and Sadao Kurohashi. 2016. Supervised syntax-based alignment between english sentences and abstract meaning representation graphs. *arXiv preprint arXiv:1606.02126*.
- Marco Damonte, Shay B Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. *Proc. of EACL*.
- Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime G Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proc. of HLT-NAACL*.
- Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proc. of ACL*. Baltimore, Maryland, USA, pages 1426–1436.
- Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proc. of COLING*.
- Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proc. of ACL*. Vancouver, Canada, pages 1127–1138.
- Kevin Knight, Laura Baranescu, Claire Bonial, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, and Nathan Schneider. 2014. *Abstract Meaning Representation (AMR) Annotation Release 1.0*. Technical Report LDC2014T12, Linguistic Data Consortium, Philadelphia, Pennsylvania, USA. <https://catalog.ldc.upenn.edu/LDC2014T12>.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proc. of ACL*.
- Geert-Jan M Kruijff. 2006. Dependency grammar. In Keith Brown, editor, *Encyclopedia of Language and Linguistics (Second Edition)*, Elsevier, Oxford, pages 444–450.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proc. of ACL System Demonstrations*.
- David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Daniel Zeman, Zdeněk Žabokrtský, and Jan Hajič. 2013. Cross-language study on influence of coordination style on dependency parsing performance. Technical Report 49, ÚFAL MFF UK.

¹⁷The missing edge counts include gold edges for which the parser failed to produce one or both endpoints.

- Dipendra Kumar Misra and Yoav Artzi. 2016. Neural shift-reduce CCG semantic parsing. In *Proc. of EMNLP*.
- Joakim Nivre. 2005. Dependency grammar and dependency parsing. Technical Report MSI report 05133, Växjö University School of Mathematics and Systems Engineering, Växjö, Sweden.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing. In *Proc. of SemEval*. Dublin, Ireland, pages 63–72.
- Barbara H Partee. 2014. A brief history of the syntax-semantics interface in Western formal linguistics. *Semantics-Syntax Interface* 1:1–20.
- Xiaochang Peng, Chuan Wang, Daniel Gildea, and Nianwen Xue. 2017. Addressing the data sparsity issue in neural amr parsing. In *Proc. of EACL*.
- Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English strings with Abstract Meaning Representation graphs. In *Proc. of EMNLP*. Doha, Qatar, pages 425–429.
- Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from Abstract Meaning Representations. In *Proc. of INLG*. Edinburgh, UK, pages 21–25.
- Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics* 4:127–140.
- Sebastian Schuster and Christopher D Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proc. of LREC*.
- Gabriel Stanovsky, Jessica Fidler, Ido Dagan, and Yoav Goldberg. 2016. [Getting more out of syntax with PropS](https://arxiv.org/abs/1603.01648). *arXiv:1603.01648 [cs]* arXiv: 1603.01648. <http://arxiv.org/abs/1603.01648>.
- Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *arXiv preprint arXiv:1705.09980*.
- Chuan Wang, Nianwen Xue, Sameer Pradhan, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proc. of NAACL-HLT*.
- Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal Decompositional Semantics on Universal Dependencies. In *Proc. of EMNLP*. Austin, Texas, USA, pages 1713–1723.
- Sheng Zhang, Rachel Rudinger, and Benjamin Van Durme. 2017. An evaluation of PredPatt and Open IE via stage 1 semantic role labeling. In *Proc. of IWCS*. Montpellier, France.

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 $\xrightarrow{\text{mode}}$ interrogative, imperatives by $\xrightarrow{\text{mode}}$ imperative, and exclamations/interjections by $\xrightarrow{\text{mode}}$ 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* \xleftarrow{nsubj} *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*, *doj*, *ioj*, 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.¹⁹ 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* $\xrightarrow{ARG0}$ *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 **du-**

¹⁹In UD version 1, and therefore the examples in this paper, the conjunction attaches to the first conjunct, whereas in version 2 it attaches to the next successive conjunct (<http://universaldependencies.org/v2/summary.html>).

plicate 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) envelops 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* \xrightarrow{doj} *attempt* ~ *attempt-01* in figure 2). All subgraph alignments which involve the light verb or its complement have to involve the 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* $\xrightarrow{compound-prt}$ *off* ~ *pay-off-02*).

Prepositional phrase PP modifiers typically involve an extra dependency edge for the preposition attachment, as with *lies* $\xrightarrow{nmod-in}$ *sun* \xrightarrow{case} *in* ~ *lie-07* $\xrightarrow{location}$ *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* $\xrightarrow{ARG0-of}$ *do-02* $\xrightarrow{ARG1}$ *thing* \xrightarrow{mod} *evil*), and in figure 3 *general* ~ *person* $\xrightarrow{ARG0-of}$ *have-org-role-91* $\xrightarrow{ARG2}$ *general*) AMR decomposes certain words by convention

which must always be structurally aligned, such as $ago \sim before \xrightarrow{op1} now$ and $government \sim government-organization \xrightarrow{ARG0-of} govern-01$.

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 $pm \xrightarrow{nummod} 9:00$ would be represented in the AMR as $date-entity \xrightarrow{time} 21:00$; tokens $9:00$ and pm are treated as a multiword expression: each is lexically aligned to "21:00". Moreover, we also align $9:00 \xleftarrow{nummod} pm \sim 21:00$ and $9:00 \xleftarrow{nummod} pm \sim date-entity \xrightarrow{time} 21:00$.

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. $prudent \sim imprudent$).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 *no*, *none*, *not*, or *never*.
5. The label of a consists of multiple words, and label of d matches any of them under rule 1. (e.g. $sit \sim sit-down$, $war-torn \sim war$).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 *take*, *have*, *make*, *did*, *get*, *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.

²⁰We determined this threshold on the development set.

²¹We use a list of morphologically negated words provided by Ulf Hermjakob.

²²This rule misaligns some AMR-specific node types, e.g. $government \sim government-organization$.