Bridging Semantic Frameworks: mapping DRS onto AMR

Siyana Pavlova, Maxime Amblard, Bruno Guillaume

Bridging Semantic Frameworks: mapping DRS onto AMR

Siyana Pavlova, Maxime Amblard, Bruno Guillaume
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{firstname.lastname}@loria.fr

Abstract

A number of graph-based semantic representation frameworks have emerged in recent years, but there are few parallel annotated corpora across them. We want to explore the viability of transforming graphs from one framework into another to construct parallel datasets. In this work, we consider graph rewriting from Discourse Representation Structures (Parallel Meaning Bank (PMB) variant) to Abstract Meaning Representation (AMR). We first build a gold AMR corpus of 102 sentences from the PMB. We then construct a rule base, aided by a further 95 sentences. No benchmark for this task exists, so we compare our system's output to that of state-of-the-art AMR parsers, and explore the more challenging cases. Finally, we discuss where the two frameworks diverge in encoding semantic phenomena.

1 Introduction

Many semantic representation frameworks have emerged over the years (Kamp and Reyle, 1993; Copestake et al., 2005; Banarescu et al., 2013; Abend and Rappoport, 2013), at varying levels of abstraction in terms of encoding semantic phenomena. We want to be able to compare frameworks empirically across phenomena, with the goal to understand, unify and extend them. Unfortunately, this is difficult to do in a data-driven manner as there are few freely available parallel datasets. As manual annotation is laborious, it is important to develop automatic tools to create and expand datasets. One way to approach this is by transforming annotations across frameworks. In this work, we take a look at Abstract Meaning Representation (AMR) (Banarescu et al., 2013), and Discourse Representation Structures (DRS) (Kamp and Reyle, 1993), as expressed in the Parallel Meaning Bank (PMB) (Abzianidze et al., 2017), to see how much of the former can be constructed from the latter.

We show a significant portion of AMR can be constructed from DRS and provide a discussion on our insights as to where the process is not possible. To achieve this we build a graph rewriting system from DRS to AMR. As there is no parallel data between the two, we also annotate a small part of the PMB into AMR.

Our motivation for this work is twofold. Our first goal is to get more parallel annotated data between semantic formalisms in general, and between AMR and DRS for this particular study, in order to foster empirical cross-formalism comparison. A natural question to ask here is, since (as we will see in section 5) automatic parsers based on machine learning techniques seem to perform better than rule-based transformation systems on this task, why do we bother with such an experiment. We have a few reasons: (i) rule-based transformation systems may still perform quite well, especially for more closely-related formalisms (as we show in this study) and we do not know how well exactly until we test such a system; (ii) it is possible that the two approaches make different kinds of mistakes, which opens the possibility for hybrid solutions that combine their strengths; (iii) with a rule-based system, tracking the decision-making process is possible, rendering the method explainable.

Our second goal is to better understand the differences between formalisms with a view to extend and unify them. This is difficult to do in a non-data-driven manner as the formal definitions of formalisms are rarely complete. More importantly, within the community, it is not clear what a *complete* semantic representation should consist of. Thus, while not as direct as the first, an outcome we hope to get from this work is a deeper insight into what is needed in a semantic representation and what are the missing links between formalisms, as a step towards defining a unifying framework.

The rest of the paper is structured as follows:

in section 2, we present the two frameworks; in section 3 – our graph rewriting system (GRS); section 4 is about our annotation procedure for a small gold AMR dataset; in section 5, we present our experiments, discuss the results, and compare them to those of SoTA AMR parsers; in section 6, we provide a discussion and future work directions. Our code and data are publicly available[1].

2 Background

In this section we present the Parallel Meaning Bank as an instance of a large corpus of Discourse Representation Structures, and Abstract Meaning Representation.

2.1 DRS in the PMB

The Parallel Meaning Bank (PMB) is a semantically annotated corpus, with parallel annotations available for four languages – English, German, Italian and Dutch. The portion of the PMB that contains gold annotations for English is significantly larger than the other three: 10,715 sentences vs 2,844 (German), 1,686 (Italian) and 1,467 (Dutch). The formalism behind the PMB semantic representations is Discourse Representation Theory (DRT) (Kamp and Reyle, 1993) and in particular Projective DRT (PDRT) (Venhuizen, 2015), which differs from DRT in the way it accounts for presuppositions and conventional implicatures. DRT expressions are called Discourse Representation Structures (DRS). DRS are typically represented as boxes with variables defined at the top of the box and the entities and relations between them in the bottom. The boxes are used to label scopes and discourse units. Similar to (Muskens, 1996), the PMB "dialect" of DRS is compositional and it allows to embed boxes into one another, specifying the relations between them. Sentences from the PMB can be viewed on the PMB explorer[2]. There, DRS's can be seen in three kinds of notation: the traditional box notation (Figure 1a), clause notation (Figure 1b), and the recently proposed Simplified Box Notation (SBN) (Bos, 2021) (Figure 1d).

The PMB uses WordNet (Fellbaum, 1998)[3] to encode senses (e.g. `attack.v.04`, `shark.n.01`) and VerbNet/LIRICS (Bonial et al., 2011) for semantic roles (e.g. `Agent`, `Patient`, etc.).

For the purposes of our work, as the three notations available in the PMB are equivalent[4], we use SBN as a starting point, as it is simplest to process. We transform SBN representations into graphs for easier manipulation and visualisation (Figure 1c).

2.2 AMR

Abstract Meaning Representation (AMR) represents "who did what to whom" in a sentence. It is meant to be rather abstract in order to be easily-readable by humans and easier for annotators to work with. The simplification is achieved by not encoding phenomena such as tense, plurality or scope, though this can also be seen as a disadvantage.

AMR abstracts away from the surface representation, allowing multiple sentences with the same meaning to have the same representation. The AMR in Figure 2 is the representation of the sentence "*He was attacked by a shark.*", but also of "*A shark attacked him.*". Furthermore, as AMR does not encode various semantic phenomena, sentences with similar (but not the same) meanings can also get the same representation. The AMR in Figure 2 also represents the sentences "*The shark attacked him.*" and "*Sharks will attack him.*", among others.

AMR is centered around predicate-argument structure and, for English, makes extensive use of PropBank predicates (Palmer et al., 2005). Predicates are used to annotate verbs in a sentence, but also adjectives, and sometimes even nouns, if the appropriate PropBank frames exist. Each predicate has a set of arguments which are called *core roles* and appear as numbered arguments in AMRs (see `ARG0` and `ARG1` in Figure 2). Additionally, *non-core roles* such as `time`, `domain`, `duration` make up the rest of the AMR relations.

AMRs are directed acyclic graphs (DAGs) with a single root. Respecting both of these properties does not always come naturally. To preserve both, an AMR role can be inverted by changing its direction and adding `-of` to its label. Inverse roles are also useful for highlighting the *focus* of a sentence.

Unlike for DRS, the larger, more commonly used AMR datasets, are only available via a paid license from the Linguistic Data Consortium[5]. Still, a smaller portion of the so-called AMR Bank is freely available[6], namely the Little Prince corpus and the BioAMR corpus. However, for the purposes of our work, we need parallel data between DRS and

[1] https://gitlab.inria.fr/semagramme-public-projects/drs2amr

[2] https://pmb.let.rug.nl/explorer/explore.php; data freely available under ODC-BY 1.0

[3] https://wordnet.princeton.edu/

[4] with the exception of PRESUPPOSITION

[5] https://www.ldc.upenn.edu/

[6] https://amr.isi.edu/download.html

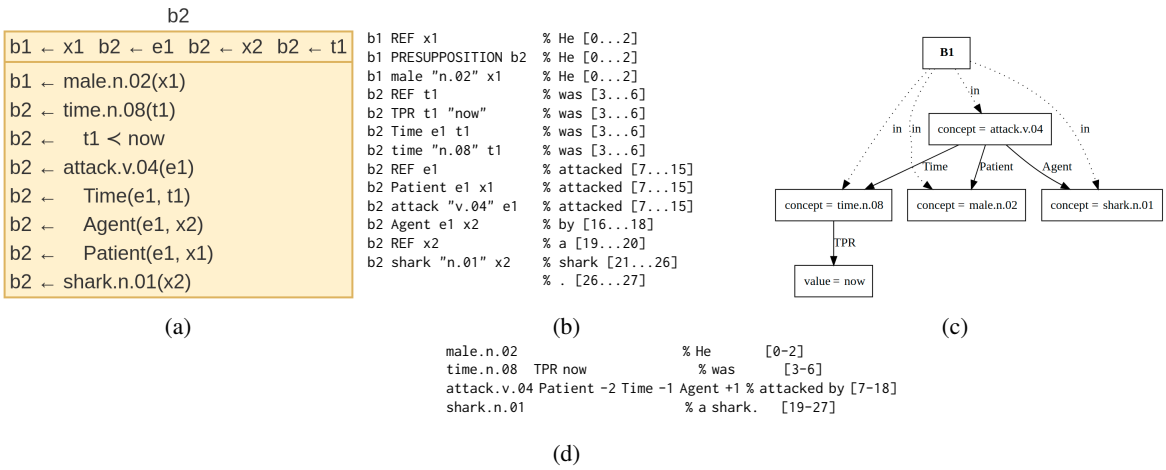

Figure 1: The sentence "He was attacked by a shark." in box notation (a), clause notation (b), as a graph (c), and in simplified box notation (SBN) (d).

```
(a / attack-01
   :ARG0 (s / shark)
   :ARG1 (h / he))
```

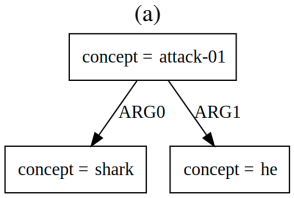

Figure 2: AMR annotation of the sentence "*He was attacked by a shark.*", among others, in (a) Penman notation and (b) as a graph.

AMR. Since that, to the best of our knowledge, does not exist, we chose to annotate a small portion of the PMB into AMR (section 4).

3 System

We use GREW[7] (Guillaume, 2021; Bonfante et al., 2018) to build a graph rewriting system (GRS) for rewriting SBN graphs into AMR ones. GREW is a tool that allows the user to define rules to match patterns in a graph and apply a set of commands that transform the matched part of the graph. GREW also allows for the use of lexicons, which lets us map sets of values and assign a value to a variable based on the value of another variable.

3.1 Lexicons

AMR and the PMB use different lexical resources, so our system relies extensively on lexicons to map them: WordNet verbs to PropBank predicates, and VerbNet semantic roles to PropBank arguments.

SemLink[8] (Palmer, 2009) is an existing effort aimed at linking English linguistic resources, among which PropBank and WordNet. We scraped the SemLink verb groupings[9] to collect mappings between the WordNet senses used in our dataset and the corresponding PropBank predicates. We note that this is a many-to-many mapping, as can be seen in the sample below.

```
wn_pred    pb_pred
%=================
try.v.01   try-01
try.v.01   try-04
play.v.01  play-01
play.v.07  play-01
```

We collected 133 such mappings in total for the 197 (95 from *dev* set + 102 from *test* set) sentences in our dataset[10]. These span across 110 WordNet senses and 119 PropBank predicates. The mappings do not cover all the predicates present in our dataset, as either the WordNet or PropBank predicate does not exist in its respective resource or the mapping between the two is not a part of SemLink. We found 22 such WordNet predicates, 13 of which correspond to phrasal verbs.

Next, for each PropBank predicate in our lexi-

[7] https://grew.fr/

[8] https://verbs.colorado.edu/semlink/

[9] https://verbs.colorado.edu/html_groupings/

[10] For this experiment, we wanted to simulate having a complete lexicon mapping. Constructing such a mapping is out of the scope of this work. Instead, we chose to have a lexicon that is "complete" at least for our dataset by pulling the predicate senses appearing in both the *dev* and *test* sets.

con, we manually[11] went over the corresponding PropBank entry and collected the VerbNet role for each argument where that was present. This way, we produced the first version of our lexicon, which we will refer to as *incomplete lexicon*. Out of the 119 PropBank predicates, 61 (around 51%) were missing a VerbNet role for some or all arguments.

Finally, we produce the final version of our lexicon, which we will refer to as *complete lexicon*. We do this by going over all the predicates again and deciding on a VerbNet role for each of the arguments that do not have one.

For example, in PropBank, for `try-01`, we have the following:

Arg0-PAG: *Agent/Entity Trying* (vnrole: 61.1-agent)
Arg1-PPT: *thing tried* (vnrole: 61.1-theme)

whereas for `try-04`, we have:

Arg0-PAG: *tryer*
Arg1-PPT: *thing tried (hand, patience)*
Arg2-PRD: *attribute of Arg 1*

As can be seen, for `try-01` the corresponding VerbNet roles are explicitly specified in the brackets, whereas for `try-04` they are not. Thus, while the *incomplete lexicon* contains entries for both `try-01` and `try-04`, it specifies the PropBank numbered arguments only for `try-01`.

```
wn_pred  pb_pred Agent Theme ...
%==============================
try.v.01 try-01  ARG0  ARG1  ...
try.v.01 try-04  -     -     ...
```

The *complete lexicon*, on the other hand, specifies the roles for `try-04` as well. As can be seen below, based on the descriptions from PropBank, we have decided to link `ARG0` to `Agent`, `ARG1` to `Theme` and `ARG2` to `Attribute`.

```
wn_pred  pb_pred Agent Theme Att.
%==============================
try.v.01 try-01  ARG0  ARG1  -
try.v.01 try-04  ARG0  ARG1  ARG2
```

The PMB typically uses WordNet's `measure.n.02` as a node when talking about quantities. In AMR, this is more fine-grained, with concepts such as `temporal-quantity` or `distance-quantity`. Many of these can be deduced based on the `:unit` of said quantity, e.g. if the `:unit` is `day`, then the concept should be `temporal-quantity`. To address this, we also produce and use a lexicon which maps unit types to quantity types.

3.2 Our Graph Rewriting System

Our Graph Rewriting System (GRS) includes a few groups of rules, centered around different types of roles or structures in both AMR and the PMB. We selected partition *00* of our split of the PMB (see section 4 for explanation on partitions) as the set used for constructing rules, referred to hereupon as our *dev* set. AMR annotations for it were produced by annotator D (see section 4). All our data comes from the English section of the PMB.

Core roles with lexicon. This set of rules encompasses a rule for picking a PropBank predicate for the WordNet verbs in the input SBN graph if a mapping for that WordNet verb is present in our lexicon, and rules for rewriting the VerbNet roles from the input graph into PropBank numbered arguments. This category contains 27 rules – one for sense picking and one each for the 26 VerbNet predicates in our lexicon.

Core roles without lexicon. Here, we include a set of rules that rewrite the most common VerbNet roles, `Agent`, `Patient`, `Theme`, `Stimulus` and `Experiencer`, into PropBank numbered arguments in case they were not present in the lexicon for the relevant PropBank entry. For each, we select the most common numbered argument that that role has in our lexicon. These are later referred to as our *fallback* rules. This category contains 5 rules.

Non-core roles. This set of rules covers rewriting of PMB roles, such as `Duration`, `Manner`, `Beneficiary`, etc., to their AMR counterparts (`:duration`, `:manner`, `:beneficiary`, etc.). This category contains 21 rules.

Structures. Another set of rules deals with what we call structures. As structures, we consider a set of nodes and edges (as opposed to just a single node or a single edge) that can be rewritten into another set of nodes and edges or an individual edge. One such example is the structure used by the PMB when we have `person.n.01 -EQU-> speaker`. This corresponds to using either the concept `I` or the concept `we` as a single node in place of the

[11] This seems like a lot of work for a small dataset, but it is a one-off effort. Once done for the entire sense bank for a given language, it can be used for all datasets for that language.

whole structure. Here we also include rules where a single node or edge is rewritten into a set of nodes and edges. An example of this is the rule we use for named entities that rewrites the edge `Name` from the SBN graph into a structure that encompasses the `name`, `wiki` and their corresponding values in the AMR graph. We have 25 rules in this category.

Special words. A small set of rules deals with special concepts and relations. One such example is the concept `be-03` which is most often used to refer to spatial location and therefore invokes the special AMR concept `be-located-at-91`. There are 12 rules in this category.

Boxes. As described in subsection 2.1, the PMB groups nodes in boxes. When there is a single box in the SBN representation of a sentence, this generally does not bring any new information for the AMR graph. However, when more than one box is present, for example to introduce phenomena such as negation or universal quantification, this can be informative for the AMR graph as well. For example `B1 -NEGATION-> B2` can introduce a `:polarity -` relation to AMR. Our final set of rules deals with the different types of relations between boxes when more than one box is present. A final rule removes all the boxes that are left at the end. This category contains 36 rules.

The different sets of rules presented here are applied in the following order: special words, core roles with lexicon, non-core roles, structures, core roles without lexicon, boxes, except the two rules dealing with the `AttributeOf` SBN role, which are applied after boxes. An additional rule for removing cycles with three nodes by inverting one of the relations in the cycle is applied at the end.

Some of the rules are combined into non-deterministic strategies. For example, since there is no way to tell from the SBN graph only (i.e. without referring to the text) whether `person.n.01 -EQU-> speaker` refers to `I` or `we`, both versions are produced by our GRS. Similarly, as we mentioned in subsection 3.1, the WordNet to PropBank predicate mapping is many-to-many. In case a WordNet predicate maps to multiple PropBank ones, all possible graphs are produced.

3.3 Post-processing

After applying the GRS to our data, we do some post-processing on the GREW graphs. For named entities, our `GRS` only produces an `:op1` property for the `name` of the entity even if the name consists of multiple words. This is addressed in the post-processing step by adding `:op2` to `:opN` accordingly. Additionally, for any remaining WordNet concepts (be it verbs, nouns or adjectives) we remove the trailing part starting from the first dot, i.e. `piano.n.01` becomes `piano`. Finally we produce the PENMAN notation for the output AMR graph (or graphs in the case of non-determinism).

4 Gold Data

To evaluate our system, we produced gold AMR annotations for 102 sentences of the English part of the PMB. In order to make sure that there were no specific phenomena concentrated in certain partition of the PMB data, instead of picking a random partition and risking having a non-representative sample, we applied an algorithm to "randomise" that[12]. We created 100 new partitions, by finding the sum of the part and document number of each sentence and applying modulo 100 to get a new partition number. This approach groups the data randomly, but is reproducible and as the PMB expands, the partitions should grow in a fairly uniform manner. Version 4.0.0 of the PMB contais 10,715 gold English sentences, so 107 sentences on average per partition.

We picked partition 25 (i.e. all the documents for which $(p + d)\%100$ is 25) to annotate manually. It contains 102 sentences. Our four annotators – **A**, **B**, **C** and **D** – annotated half of the sentences (51) each. Every sentence was annotated by two annotators. To ensure that each pair of annotators had the same number of overlapping sentences, we split the 102 sentences into six groups of 17 and distributed the groups among the six different pairings.

The annotators consulted the following resources during the annotation process:

- AMR Specifications[13] as the primary source for examples and explanations on how to annotate different phenomena

[12] As per (Abzianidze et al., 2017), the corpora used in the PMB are balanced across parts. It is difficult to verify whether the dataset is also balanced across various semantic phenomena. In case it is, our randomization step is not necessary. We do, however, want to point out that the English gold data in the PMB 4.0.0 is not distributed uniformly across parts. Those towards the beginning and those with a number divisible by 10 have significantly more sentences than the rest (see subsection C.1 in the Appendix).

[13] `https://github.com/amrisi/amr-guidelines/blob/master/amr.md` (points to version 1.2.6 of the specifications at the time of writing)

- AMR Annotation Dictionary[14] for additional annotation examples grouped by specific roles, concepts, words and constructions
- PropBank Searchable Frame Files[15] for PropBank predicates and their argument structures
- A full list of PropBank frames from the AMR website[16] to find "hidden" AMR frames (e.g. "strong-02" is hidden in `strengthen.html` in the Searchable Frame Files). PropBank Searchable Frame Files took precedence in case of conflict.
- GREW-MATCH[17] to search for examples of different concepts or structures, in graph format. For AMR, GREW-MATCH currently contains all the examples from the AMR Specifications, AMR Annotation Dictionary, The Little Prince corpus, and the BioAMR corpus.

We used Smatch (Cai and Knight, 2013) to compute the inter-annotator agreement (IAA). Smatch uses a hill-climbing algorithm to find the maximum number of triples between two graphs. There are three types of triples: instance, relation, and attribute. Instance triples match nodes in the graph, counting exact matches between the node concepts. Relation triples match edges in the graph. Attribute triples match properties of the nodes. Each type has equal weight in the overall score count.

The results of our IAA are reported in Table 1. Annotator **A** appears to have the lowest agreement with the other three annotators. One reason for this may be that annotator **A** correctly observed that named entities in AMR always get a `:wiki` property, even if they do not have an existing Wikipedia page[18] and added them accordingly. The other three annotators only added a `:wiki` property to Wikipedia named entities. We have adopted annotator **A**'s approach for the gold data.

To produce the final version of the gold data, the four annotators gathered in groups (two, three, or four) over the course of a few sessions. For each sentence, the two existing annotations were compared and after a discussion, one was chosen or a modification that combines elements of both annotations was selected. In a small number of cases, the annotators agreed on an entirely different annotation from the two proposed ones.

	A	**B**	**C**	**D**
A	–	0.76	0.82	0.81
B	0.76	–	0.83	0.86
C	0.82	0.83	–	0.84
D	0.81	0.86	0.84	–

Table 1: Inter-annotator agreement – Smatch f-score.

5 Evaluation

As with our IAA, we use Smatch to evaluate our system's output against the gold annotations. As mentioned in section 4, Smatch takes into account not only the graph structure, but also the exact match of concepts between graphs. Thus, we expected that the predicate lexicon and the second post-processing step (removing trailing part of WordNet concepts) would have a substantial impact on the final score. To evaluate this, we run the experiment with *no lexicon* (1), with the *incomplete lexicon* (2), and with the *complete lexicon* (3). Additionally, we also run a version with the *complete lexicon*, but without the second post-processing step (4). Finally, while adding a lexicon of senses increases the results for both the *test* and *dev* sets significantly, we see that the difference in results between the *incomplete lexicon* and the *complete lexicon* setting is very small. Our hypothesis is that this is due to the fallback rules for core roles that have not been rewritten. To verify this, we run the three different lexicon versions (*no lexicon* (5), *incomplete lexicon* (6), and *complete lexicon* (7)) also without the fallback rules. We run each of these experiments on both the *dev* and *test* sets.

The results from our experiments are reported in Table 2. As can be seen our hypothesis about the benefit of a lexicon and the post-processing step is justified: we get an increase of $6-7\%$ on both the *dev* and *test* sets for all scores. When we consider the fallback rules, we can compare experiments (1), (2) and (3) with experiments (5), (6) and (7). We see that the fallback rules do a lot of the groundwork, but more so when we have no lexicon or an incomplete one. They have less of an impact when working with a complete lexicon.

Due to non-deterministic rules, for some sen-

[14] https://www.isi.edu/~ulf/amr/lib/amr-dict.html

[15] http://verbs.colorado.edu/propbank/framesets-english-aliases/

[16] https://amr.isi.edu/doc/propbank-amr-frames-arg-descr.txt

[17] http://semantics.grew.fr/

[18] Indeed, we observe that all the named entities in the AMR annotated data, except from one sentence from the BioAMR corpus have a `:wiki` property.

tences we get more than one output graph. As we want to see what is the biggest part of the AMR structure that can be built from DRS, the scores we report take the graph with highest overlap (according to Smatch F1-score) with the gold graph[19].

Comparing the scores on the *dev* and *test* sets, we notice the big disparity in the scores between the two. This is due to our building the rules based on the *dev* set and thus missing out on structures that do not appear in it, but appear in the *test* set. This suggests that our rule set is incomplete and a larger *dev* set may be necessary to ensure broader structure coverage. One such example is that the structure used in the PMB for expressions such as *instead of* and *rather than* needs special treatment which requires either the duplication of a specific node or the introduction of the predicate *prefer-01*. However, since our rule base was built from the *dev* set and that does not include such an example, we do not have a rule to address it. We do, however, have such an example in the *test* set and it cannot be addressed properly.

5.1 Comparison to AMR parsers

As far as we are aware, transforming DRS graphs into AMR ones is a new task. There is, therefore, no benchmark against which we can compare our outputs. For the sake of argument, however, we got predictions from two SoTA AMR parsers – an ensemble one, and a single-model one.

MBSE. Maximum Bayes Smatch Ensemble (MBSE) (Lee et al., 2022) is an ensemble distillation model that combines knowledge from a number of models to produce a single prediction. MBSE is currently the SoTA AMR parser.

AMRBART. AMRBART (Bai et al., 2022) uses graph-to-graph pre-training to improve pre-trained language models' awareness of the graph structure of AMRs. It is currently the best single-model and fifth best overall parser on the AMR2.0 and AMR3.0 datasets.

We sent our dataset to the MBSE authors and obtained the predictions from the Ensemble-5 MBSE model back from them. As for AMRBART, we ran the fine-tuned on AMR parsing AMRBART-large (AMR2.0)[20] on our dataset. The granular Smatch scores[21] (Damonte et al., 2017) for these two as well as for our system on both our *test* and *dev* sets are in Table 3. MBSE predictions have not been wikified. We expect that after wikification, MBSE's score will be on par with AMRBART's.

As can be seen in the table, the AMR parsers perform better overall compared to our system. We believe there are two main reasons for this. Firstly, the AMR parsers have been trained on a lot more data: tens of thousands of sentences versus 95 for our system. Secondly, we are limited by the information that is present in the DRS and parts of the AMR structure simply cannot be predicted from it (see section 6 for further discussion).

A closer look at the granular scores indicates that the areas where our system performs particularly poorly is when dealing with negations and reentrancies, both of which are the hardest areas for the parsers as well. For negation, we owe this to the fact that in DRS, when negation is morphological, but there is a corresponding WordNet concept, as is the case with `unhealthy.a.01` in "I knew it was unhealthy" (*26/2674*), this is expressed in one node, whereas in AMR, we have a node for `healthy-01` and a node that negates that[22].

In some cases, our system performs better than the parsers. For example, in sentences that use comparison (e.g. "This car is bigger than that one" (*67/2333*)). However, this is likely because our rule for handling these cases was built following the AMR guidelines, as was our gold dataset. The data that the two parsers have been trained on uses a different than in the guidelines structure, leading them to learn that instead. To their credit, our hypothesis is that if they were trained on the same structure, they would be more likely to predict it correctly.

5.2 Error Analysis

As discussed earlier, some of our errors are due to certain structures not being present in our *dev* set. These do not, however, account for the errors on the *dev* set itself. There are a number of other aspects which come at play here.

Missing predicates from lexicon. A number of the predicates in our sentences, while present in both WordNet and PropBank, do not appear in Semlink. Therefore we have not been able to add them to our lexicon. This leads to a non-overlap

[19] For completeness, if we take the worst graph instead, the F1-score for experiment (3) is 0.71 for *dev* and 0.65 for *test*.

[20] https://github.com/goodbai-nlp/AMRBART

[21] https://github.com/mdtux89/amr-evaluation

[22] It would be possible to address this via a rule that captures such words. However, only words where the negative particle is indeed a morpheme, need to follow this rule (it would not apply to *"uniform"*, for example). This would require the construction of a lexicon of words with negative morphemes. This is ultimately a task that requires morphological analysis and, as such, is out of the scope of this work.

	Dev set			Test set		
	Precision	Recall	F1-score	Precision	Recall	F1-score
(1) No lexicon	0.72	0.71	0.72	0.66	0.62	0.64
(2) Incomplete lexicon	0.79	0.77	0.78	0.72	0.68	0.70
(3) Complete lexicon	**0.79**	**0.78**	**0.78**	**0.73**	**0.68**	**0.70**
(4) Complete lexicon, no concept fix	0.65	0.63	0.64	0.61	0.57	0.59
(5) No lexicon, no fallback	0.61	0.60	0.60	0.55	0.51	0.53
(6) Incomplete lexicon, no fallback	0.75	0.74	0.75	0.69	0.65	0.67
(7) Complete lexicon, no fallback	0.77	0.75	0.76	0.70	0.66	0.68
MBSE – no wiki	0.84	0.83	0.83	0.85	0.80	0.82
AMRBART	**0.85**	**0.83**	**0.84**	**0.86**	**0.86**	**0.86**

Table 2: Smatch scores. Where "no fallback" is not specified, it means that the fallback rules have been applied. Where "no concept fix" is not specified, it means that the post-processing concept addition has been applied.

		Smatch	Unlabeled	No WSD	Concepts	NE	Neg.	Wiki	Reent.	SRL
Dev	MBSE	0.83	0.87	0.84	0.88	0.92	0.55	–	0.66	0.84
	AMRBART	0.86	0.89	0.87	0.87	0.94	0.70	0.85	0.66	0.83
	Our system	0.78	0.84	0.79	0.78	0.91	0.40	0.68	0.53	0.75
Test	MBSE	0.82	0.86	0.83	0.86	0.94	0.60	–	0.65	0.81
	AMRBART	0.84	0.87	0.84	0.86	0.94	0.55	0.88	0.59	0.84
	Our system	0.70	0.78	0.71	0.70	0.82	0.48	0.73	0.37	0.65

Table 3: Granular Smatch scores.

between instance nodes for those predicates as well as a wrong argument structure. This is especially true in the case of adjectives since many are PropBank predicates. However, there are no adjectives in the Semlink groupings so we have not been able to add them to our lexicon.

Divergence between AMR and DRS. AMR and DRS differ in the way in which they encode certain semantic phenomena, notably scope. There are specific AMR structures for which it is not possible to decide on the correct structure, given only the DRS. We discuss some of these cases in more detail in section 6.

Inconsistencies in the PMB data. Finally, while a much smaller number, some errors are propagated from wrong annotations in the PMB dataset. An example of this can be seen in subsection C.2.

6 Discussion

Our goal with this work was to see what portion of AMR can be constructed from DRS and where that is not possible, to understand why. While constructing our rule base, we observed that the way the two frameworks encode predicate-argument structure is very similar, differing mostly in semantic role labels, where DRS relies on VerbNet roles and AMR on PropBank predicates. With an exhaustive lexicon that contains a mapping between all senses and their arguments in the two lexical resources, it will be possible to rewrite these correctly.

The most notable difference between the two frameworks is the lack of scope in AMR, whereas that is present in DRS. Some phenomena linked to scope are encoded differently in the two. E.g., universal quantification is typically encoded in the PMB in the same way as generics: the sentences "All the seats are booked." (*50/2764*) and "A cat has two ears." (*60/0913*) have a similar structure, despite the different phenomena. In AMR the two are encoded differently as the quantifier "all" is present on the surface in one and not in the other. Similarly, the quantifiers "the" and "this" are expressed in the same way in DRS: by neither being present in the representation, while in AMR "this" is expressed as a separate node and "the" is not.

DRS, as the name suggests, is centered around discourse (as opposed to dialogue) and is not meant to encode questions very well. We observe that in the PMB, wh-questions can be derived from the representation. However, this is not the case for yes-no questions, which, in the PMB are encoded exactly as their declarative counterparts. This is not the case in AMR, thus preventing us from distinguishing between the two without referring to

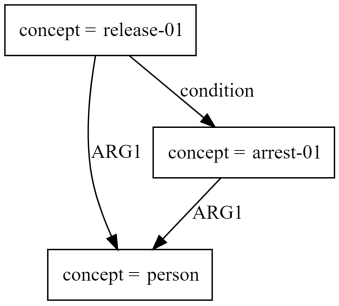

Figure 3: AMR annotation of the sentence "All who were arrested have been released." *(99/1243)*, the way it would look like if we were to follow the "logical" reading as in DRS.

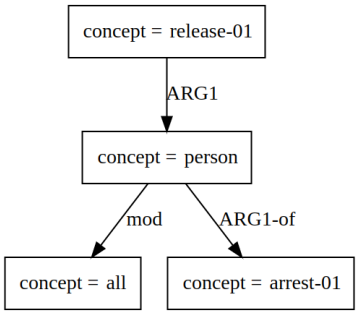

Figure 4: AMR annotation of the sentence "All who were arrested have been released." *(99/1243)*, the way a human annotator using the AMR guidelines and datasets as examples would likely annotate it.

the text of the sentence.

That being said, for a number of our rules in the **Boxes** category, we do make use of GREW's ability to check for specific regular expressions in the original sentence. This works for short sentences where there is a small or no risk of having a specific structure appear more than once. However, it is not a valid solution for longer texts. A future version of this system may benefit from using the SBN notations comments (part after % in Figure 1d).

Finally, we want to discuss a broader issue in relation to universal quantification in DRS. In the PMB, sentences such as "All who were arrested have been released." (*99/1243*) have a structure which corresponds to the reading "if a person has been arrested, they have been released". This is the way to express the semantics of universal quantification in logic. It is achieved in the PMB by using a combination of a `CONDITION-CONSEQUENCE` box embedding. This can be rewritten into AMR by making use of the non-core role `condition`, obtaining the graph in Figure 3. This is a correct reading of the sentence. However, if an annotator was to follow the AMR guidelines, we would get the graph in Figure 4. We believe this is also a correct representation of the sentence. While logically the two may be equivalent, the graph representations are structurally different. This raises the question of whether we can have more than one correct AMR per sentence. If so, then this opens the door for future considerations on how to take that into account in our evaluation metrics.

There are a number of other improvements to our system that are worth exploring in the future. Expanding the rule base can happen in two main ways (1) by expanding the *dev* set so that more varying structures are present and (2) thoroughly going though the different expressions in the AMR guidelines and AMR dictionary and designing rules for each of them. Ideally, a combination of the two should be considered. Furthermore, as we have seen with our experiments in section 5, having a lexicon that maps WordNet senses to PropBank predicates improves the score significantly. Our lexicon is still incomplete and can be further improved by adding adjectives, for instance. It would also be interesting to explore how our system performs on other languages (see Appendix A).

Our effort in trying to transform frameworks is not unique for the semantic representations community. In an exploration to better understand what linguistic semantic phenomena formalisms encode, Hershcovich et al. (2020) propose a rule-based conversion system from syntax and lexical semantics into Universal Conceptual Cognitive Annotation. Closer to our work in terms of formalisms used, Bos (2020) proposes AMR+ (an AMR extension to deal with scope) and a formal procedure to convert AMR+ into DRS. As a future work, we are interested in seeing how much AMRs obtained by applying the reverse procedure (from DRS to AMR+), then dropping the scope information, would differ from what we obtained with our system.

7 Conclusion

The goal of this work was to build a graph rewriting system from DRS (as in the PMB) to AMR to discover what portion of the latter can be constructed from the former. To do so, we first constructed a small AMR dataset from PMB sentences and built a lexicon mapping WordNet senses to PropBank predicates and arguments. We showed a significant part of the AMR structure is contained in DRS. Finally, we discussed their divergences.

Acknowledgements

We would like to thank: the anonymous reviewers for their feedback and suggestions, we have tried to incorporate all their remarks in the current version of the paper; the MBSE team, and Ramón Astudillo in particular, for kindly running MBSE on our dataset and providing us with the predictions; Guy Perrier for being our fourth annotator; Priyansh Trivedi for helping us run AMRBART. Part of this work has been funded by *Agence Nationale de la Recherche* (ANR, fr: National Research Agency), grant number ANR-20-THIA-0010-01.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. 2018. *Application of Graph Rewriting to Natural Language Processing*. Wiley Online Library.

Claire Bonial, William Corvey, Martha Palmer, Volha V. Petukhova, and Harry Bunt. 2011. A hierarchical unification of lirics and verbnet semantic roles. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 483–489. IEEE.

Johan Bos. 2020. Separating argument structure from logical structure in AMR. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 13–20, Barcelona Spain (online). Association for Computational Linguistics.

Johan Bos. 2021. Variable-free discourse representation structures.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Bruno Guillaume. 2021. Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–175, Online. Association for Computational Linguistics.

Daniel Hershcovich, Nathan Schneider, Dotan Dvir, Jakob Prange, Miryam de Lhoneux, and Omri Abend. 2020. Comparison by conversion: Reverse-engineering UCCA from syntax and lexical semantics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2947–2966, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Dordrecht. Kluwer.

Young-Suk Lee, Ramón Astudillo, Hoang Thanh Lam, Tahira Naseem, Radu Florian, and Salim Roukos. 2022. Maximum Bayes Smatch ensemble distillation for AMR parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5379–5392, Seattle, United States. Association for Computational Linguistics.

Reinhard Muskens. 1996. Combining montague semantics and discourse representation. *Linguistics and philosophy*, pages 143–186.

Martha Palmer. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*, pages 9–15. GenLex-09, Pisa, Italy.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Noortje Venhuizen. 2015. *Projection in Discourse: A data-driven formal semantic analysis*. Ph.D. thesis, Rijksuniversiteit Groningen.

A Limitations

There are a number of limitations of our work that we address in this section.

We work with semantics, and it can be argued that the meaning representation of a sentence should be identical regardless of the language. However, empirical experiments are necessary to verify that this is indeed the case when we work with real-world data and that our system still works for languages which are structurally very different from English.

That being said, reproducing this experiment for another language is not as straightforward as simply running our system on a dataset in a different language. For our system we rely heavily on lexical resources in English. The same are not as well-developed for most other languages.

Furthermore, as there is no parallel data between DRS and AMR, to run an evaluation on such a system for another language, requires the construction of a corpus in one or both frameworks. This comes at the cost of either training or having access to a skilled annotator who is also a speaker of the language for which the system is to be constructed.

Finally, relating to subsection 3.1, the missing VerbNet arguments for the PropBank predicates were decided on by one of the authors, after carefully reading descriptions for each numbered argument of the given predicate in PropBank. However, as none of the authors is an expert in semantic role labeling, we have to note that the decisions may not have always been what an expert in this field may have chosen.

B Ethical considerations

Our system is entirely rule-based: it does not rely on heavy computational power and takes a few seconds to run on a standard computer.

Our code and data are freely available and it is not necessary to obtain any paid resources to be able to reproduce our experiments.

C PMB data

C.1 Source distribution for English gold

Figure 5 shows that the sources where data comes from in gold English section of the PMB 4.0.0 is balanced across parts. The total number of sentences per part, however, is not evenly distributed, with parts towards the beginning and those with a sequence number divisible by 10 having more sentences than the rest.

C.2 Inconsistencies in PMB data

Though not very frequent, there are errors in the PMB annotations, which, in turn, propagate to the AMR annotations produced by our system. One such example is for the sentence "Since I didn't receive a reply, I wrote to her again" (*75/3043*). Its PMB annotation, in graph format, can be seen in Figure 6. This is incorrect, as this is the DRS for the sentence "I didn't receive a reply because I wrote to her". For the correct version of this sentence, the `NEGATION` and `EXPLANATION` labels have to be reversed, like they are in Figure 7 for the sentence "I am hungry because I did not eat lunch" (*86/1591*).

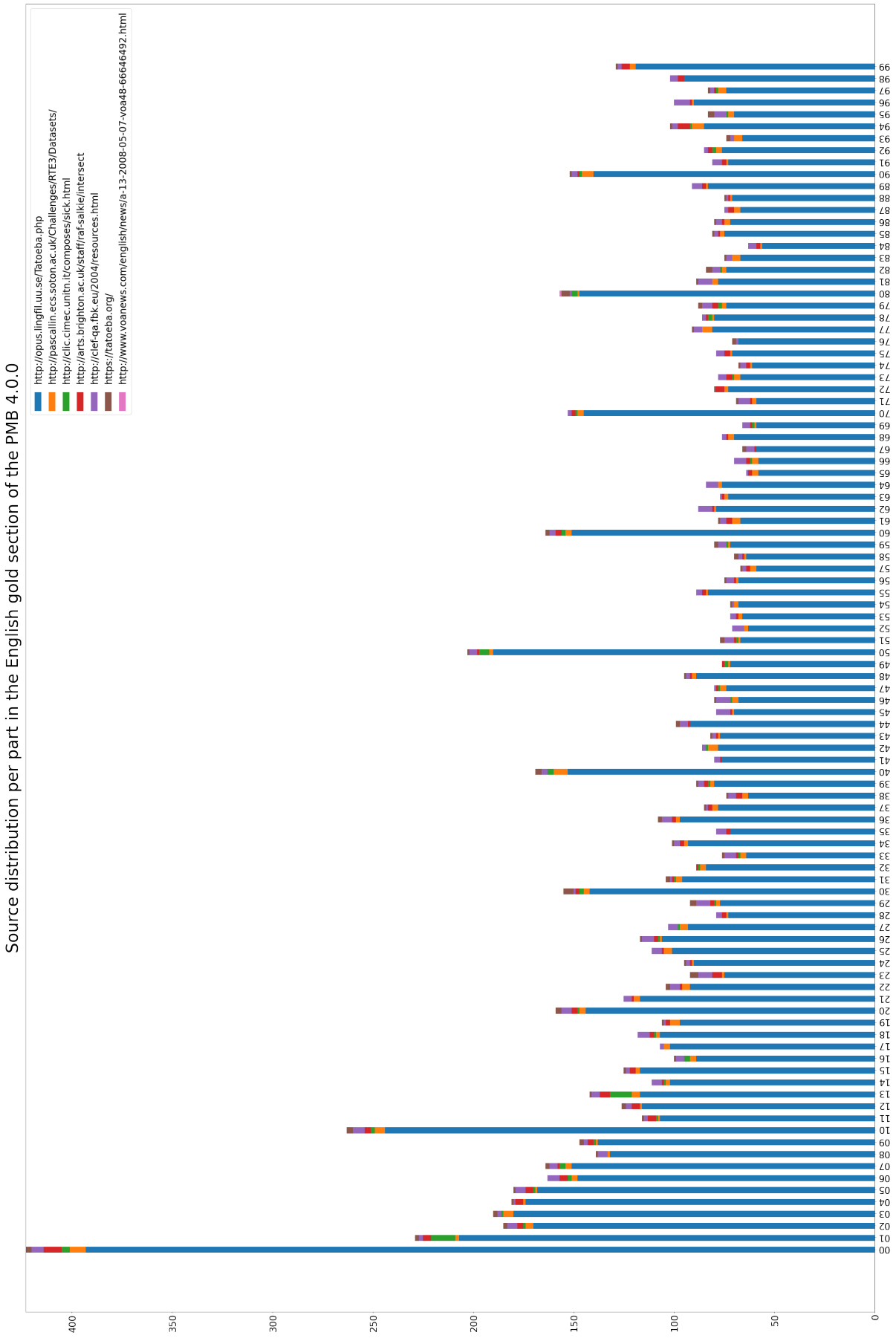

Figure 5: Distribution of the sources across the English gold part of the PMB, release 4.0.0.

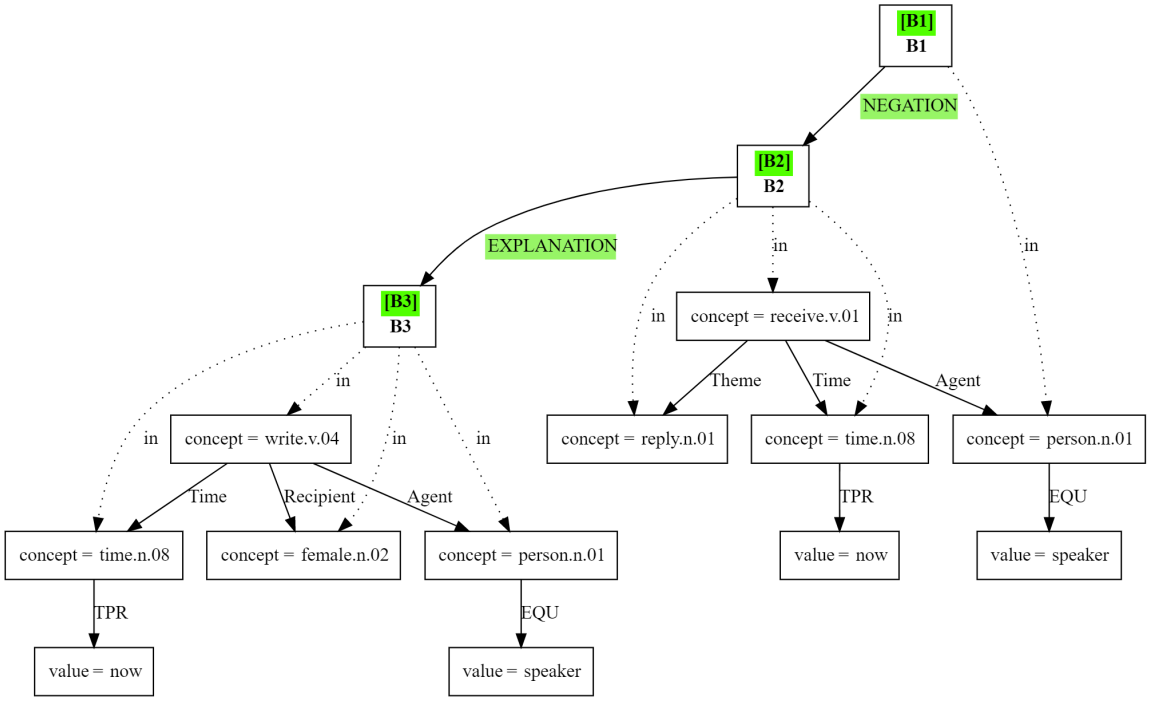

Figure 6: PMB annotation of the sentence “Since I didn’t receive a reply, I wrote to her again.” *(75/3043)*. The `NEGATION` and `EXPLANATION` labels should be reversed.

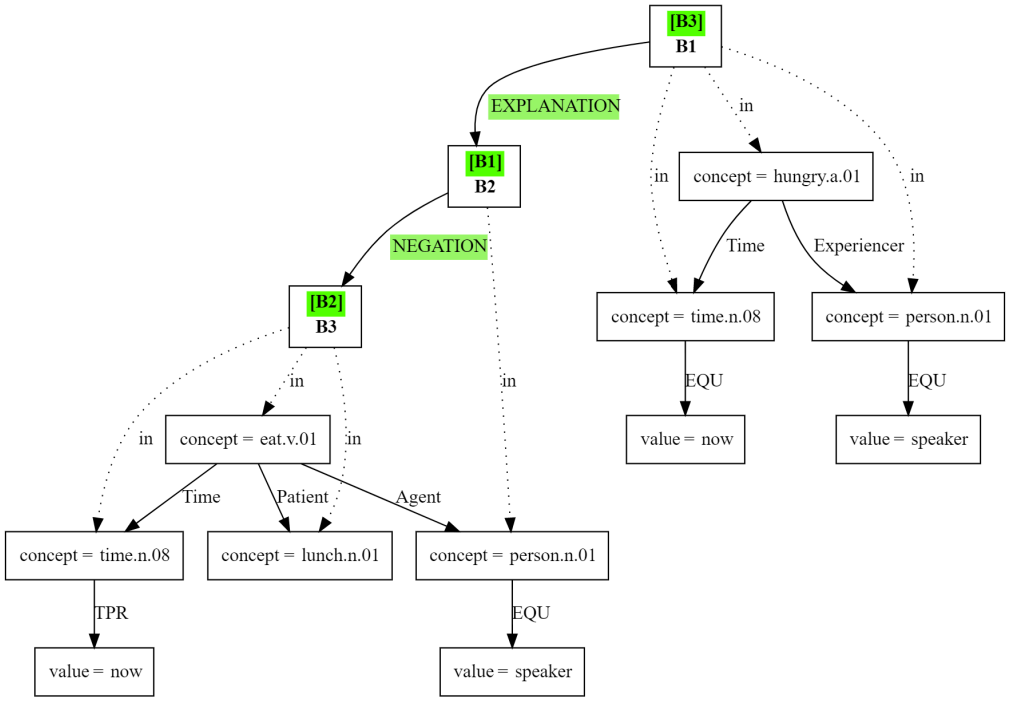

Figure 7: PMB annotation of the sentence “I am hungry because I did not eat lunch.” *(86/1591)*.