Graph Querying for Semantic Annotations

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

Abstract

This paper presents how the online tool GREW-MATCH can be used to make queries and visualise data from existing semantically annotated corpora. A dedicated syntax is available to construct simple to complex queries and execute them against a corpus. Such queries give transverse views of the annotated data, these views can help for checking the consistency of annotations in one corpus or across several corpora. GREW-MATCH can then be seen as an error mining tool: when inconsistencies are detected, it helps finding the sentences which should be fixed. Finally, GREW-MATCH can also be used as a side tool to assist annotation tasks helping to find annotation examples in existing corpora to be compared to the data to be annotated.

Keywords: Graph matching, Semantic annotations, Error mining, Abstract Meaning Representation, Parallel Meaning Bank

1. Semantic annotations as graphs

There are a huge number of proposals in the literature to describe the formal representations of the semantics of natural language texts. This diversity can be due to several factors; the main one being different linguistic theories used in the modeling. We also observe differences in terms of levels of annotations or with a specific focus on some level.

Most of these representations use the notions of objects as *entities* and *events*. They describe semantic relations between these objects. Of course, many propositions go further and propose other mechanisms to deal with temporal aspects or to describe the scope or the restriction linked to the logical interpretation of determiners as quantifiers; but we can consider that semantic relations between entities and/or events are a kind of minimal common denominator of the these proposals.

The mathematical notion of graphs is well-adapted to describe such kind of objects and we propose here to consider insofar as possible semantic annotations as graphs. In our context, we consider labeled graphs, where nodes are decorated with flat features structures and edges are associated with specific labels.

In this paper, three semantic annotation frameworks are considered: Abstract Meaning Representation (AMR), Discourse Representation Structure (DRS), as they are used in the Parallel Meaning Bank (PMB), and QuantML. The freely available annotated data for these three frameworks are now available in the GREW-MATCH[1] tool. In the following, we briefly review these frameworks and illustrate how the tool facilitates annotation while making it more consistent.

Apart from tools specific to the different formalisms, we can cite Cohen et al. (2021) which also proposed a generic framework based on graph visualisation adapted to several semantic frameworks. However, the tool is more focused on single graph visualisation and with manipulation features. It does not propose complex queries with negative application patterns or the clustering feature we describe here for GREW-MATCH.

1.1. AMR

The Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a proposal whose focus is the predicate argument structure, using PropBank (Palmer et al., 2005) as an inventory of semantic concepts.

As shown in Figure 1, we interpret an AMR annotation as a graph in the following way:

- each concept (like *fox* or *know-02* is a node with a feature name `concept`;

- each value (like *1*) is a node with a feature name `value`;

- each semantic relation, with prefix ":" in Penman notation is an edge, typed with the relation name.

In GREW-MATCH, two freely available AMR English corpora can be queried: The Little Prince Corpus version 3.0[2] (1,562 sentences) and BioAMR Corpus version 3.0[3] (6,952 sentences).

1.2. DRS in the PMB

There are several presentations of the DRS structures. In this paper we focus on the one used in the Parallel Meaning Bank (PMB) (Abzianidze et al., 2017)[4], version 4.0.0, released in October 2021. As in the AMR case, the predicate-argument structure is described with typed entities and typed semantic relations which can be converted into a graph representation. In addition, the box notation is used to describe the discourse relations and other constructions for which a notion of

---

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

[2] https://amr.isi.edu/download/amr-bank-struct-v3.0.txt
[3] https://amr.isi.edu/download/2018-01-25/amr-release-bio-v3.0.txt
[4] http://pmb.let.rug.nl/

```
(r / resemble-01
  :ARG1 (y / you)
  :ARG2 (f / fox
    :poss (i / i))
  :time (k / know-02
    :ARG0 i
    :ARG1 f
    :ord (o / ordinal-entity :value 1)))
```
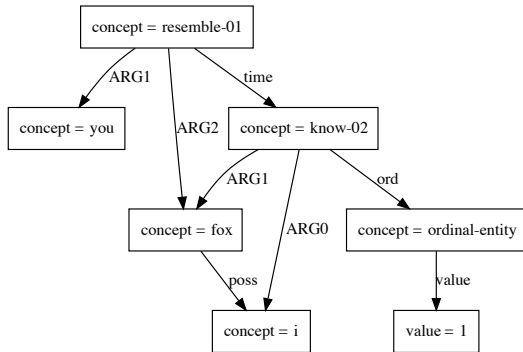
Figure 1: AMR annotation in Penman notation and its interpretation as graph of the sentence [lpp_1943.1161] *You are like my fox when I first knew him.*

scope is needed (like quantifiers or negation). The box notation requires a specific encoding into the graph structure. Following the Bos' proposal (Bos, 2021b; Bos, 2021a), each box is drawn as a new node. Moreover, the embedding of a semantic node in a box is marked with a link which is drawn with a dotted line and labelled with the relation name **in** in the figures.

```
                    NEGATION -1
be.v.01             Theme 15 Co-Theme +1
prime_number.n.01
```
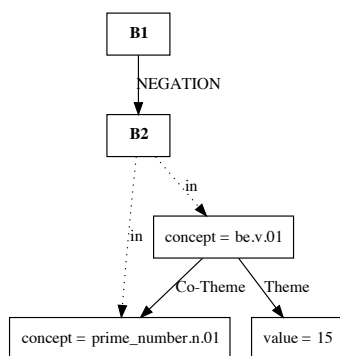
Figure 2: PMB annotation in SBN (Simplified Box Notation) and its interpretation as graph of the sentence [p52/d2324] *Fifteen is not a prime number.*

Figure 2 shows an example of the a DRS annotation from the PMB and its representation as a graph. In GREW-MATCH, the gold data of the PMB is available (10,715 sentences in English, 2,844 in German, 1,686 in Italian and 1,467 in Dutch).

1.3. QuantML

QuantML (Bunt et al., 2018; Bunt, 2020) is another semantic annotation with a focus on quantification. There is currently no annotated corpus in QuantML but a few annotations are proposed in the Guidelines part of the technical report (Bunt, 2020). Again, the so-called *Concrete syntax* of examples from the guidelines are converted into graphs.

An example of the graph associated to the concrete level of a QuantML annotation is shown on Figure 3. These graphs are richer: they use a skeleton with predicate/argument structure, but information about definiteness, distributivity or scope constraint is also given. Features structures are used to describe different semantic aspects both on nodes and edges. Scoping constraint between different arguments of the same predicate can be expressed (red edge **equal** in the figure).
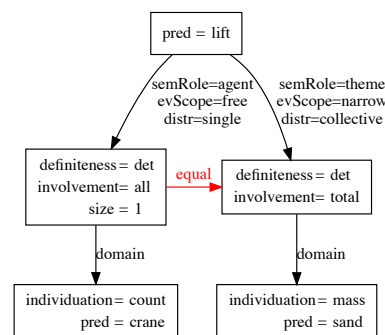
Figure 3: QuantML graph of the sentence [A10] *The crane lifted all the sand*

In Amblard et al. (2021), we have participated in the ISA-17 shared task and proposed such annotations for 7 English sentences (with two alternative annotations for an ambiguous sentence). In GREW-MATCH, these annotations are available together with the 11 English examples[5] of the guidelines in Bunt (2020).

2. The GREW-MATCH tool

In some previous works (Bonfante et al., 2018), we have proposed to consider the representation of linguistic structures as graphs and to promote the well-studied computational model of Graph Rewriting to describe transformations of these structures. In this framework, complex transformations can be encoded as a sequence of elementary and local transformations. The local steps are graph rewriting rules, composed of a pattern (describing the part of the graph to be modified) and a sequence of commands (describing in which way the graph is changed). The GREW (Guillaume, 2021) tool is an implementation of this framework.

A mechanism of graph matching is used in GREW to detect when a rule can be applied to a graph. But

---

[5]for five of them, we have detected annotation errors (see an example in section 4 and we give both the original version and the fixed version.

we have observed that this mechanism can be used on its own as a way to query a graph or a collection of graphs. This querying aspect give birth to a new tool, called GREW-MATCH, a web-based interface to express queries on annotated corpora and to visualise the occurrences returned by the query.

A screenshot of the tool applied to AMR example is shown in Figure 4. The visualisation of semantic structures uses the Graphviz tool[6].

Theoretically, graph matching is an NP-complete problem but in the present context, matching is done on a set of small graphs (one graph per sentence) and then the complexity is not an issue and most of the graph requests can be executed very quickly. We have made a few experiments on larger graphs (around 20,000 nodes) and then the complexity strongly relies of the shape of the pattern. If a pattern has a tree structure, the matching is easing and linear in the size of the graph; for general patterns, there is no generic efficient algorithm and some heuristics will be needed.

2.1. The query language

We briefly describe here the main aspects of queries in GREW-MATCH, we let the reader go to the GREW documentation pages for more details. The main part of a query is introduced by the keyword `pattern` which describes the set of nodes and edges that should be matched in the host graph. For nodes and edges, several constraints can be expressed. More general constraints can also be expressed, for example the fact different nodes share the same feature value. Given a corpus and a basic request (introduced by the keyword `pattern`), we can further refine the query by adding negative application patterns (introduced by the keyword `without`). Each negative application pattern is a constraint that filters out the occurrences returned by the basic pattern according to additional constraints.

As an example, the following pattern with one basic pattern (first line) and two negative application patterns (last two lines). It shows the syntax to express: find all the concept nodes in the corpus where the concept is `say-01` but such that there are no outgoing edges labelled **ARG0** from this node (note that the identifier `N` is used to refer to the same node) and such that there are no incoming edges labelled **ARG0-of** on this node. An example of a graph returned by this pattern is given in section 3 below.

```
pattern { N [concept = "say-01"] }
without { N -[ARG0]-> A0 }
without { A0 -[ARG0-of]-> N }
```

Another feature which has been proved useful in GREW-MATCH is the ability to cluster the results given by a complex request. The user can chose a clustering key (like the feature of one of the nodes of the basic pattern), the set of occurrences is clustered according

to value of this feature (see example in the next section).

The clustering can also be done following a sub-pattern: considering a pattern P and a sub-pattern P', all occurrences of P are clustered in two subsets P_{yes} and P_{no} depending on whether P' is also satisfied by the considered occurrence. For instance, we can observe how coordination is annotated with the pattern P: `pattern { N [concept = "and"] }` and the sub-pattern `N -[op1]-> X`, to see if the concept `and` appears with or without an **op1** outgoing edge. On The Little Prince, there are 215 occurrences in P_{yes} and 127 in P_{no}. With the same P and the sub-pattern `N -[op2]-> X`, the occurrences are 240 in P_{yes} and 102 in P_{no}. This shows that "unary" coordination (sentence beginning with the word *and* are not consistently annotated: the unique conjoint is sometimes annotated **op1** and sometimes **op2**.

3. Linguistic observations on semantic annotations

We list here a few examples of requests which can be used to make observations on the annotated corpora.

Concepts linked to a given verb. With the following request and a clustering on `N.concept`

```
pattern { N [concept = re"make-.*"]; }
```

we obtained the distribution of the usage of the concepts. On The Little Prince Corpus, the concepts returned are `make-02` (18), `make-01` (17), `make-05` (1), `make-06` (1) and `make-up-07` (1).

Realisation of an argument of a predicate. In The Little Prince Corpus, the most frequent predicate is `say-01` (234 occurrences). According to PropBank, this predicate has 4 core arguments: **ARG0** (*Sayer*), **ARG1** (*Utterance*), **ARG2** (*Hearer*) and **ARG3** (*Attributive*). With a few requests on GREW-MATCH, we can observe how often the different arguments are realised of not. For **ARG0**, the following request gives the 6 occurrences of the predicate without the *Sayer*. Note that we have to take care both of the **ARG0** outgoing edges (line 2) but also to the **ARG0-of** incoming edges (line 3) (without the last line, 9 occurrences would be wrongly reported).

```
pattern { N [concept = "say-01"] }
without { N -[ARG0]-> A0 }
without { A0 -[ARG0-of]-> N }
```

An example of one of the six occurrences is shown in Figure 5.

Observation of distributions in the data. Graph querying is also available through scripts which produces statistics about the number of occurrences in corpora. With the following pattern, and a clustering following the label of the edge named `e`, we can observe the distribution of relations between two "concept" nodes (see Figure 6).
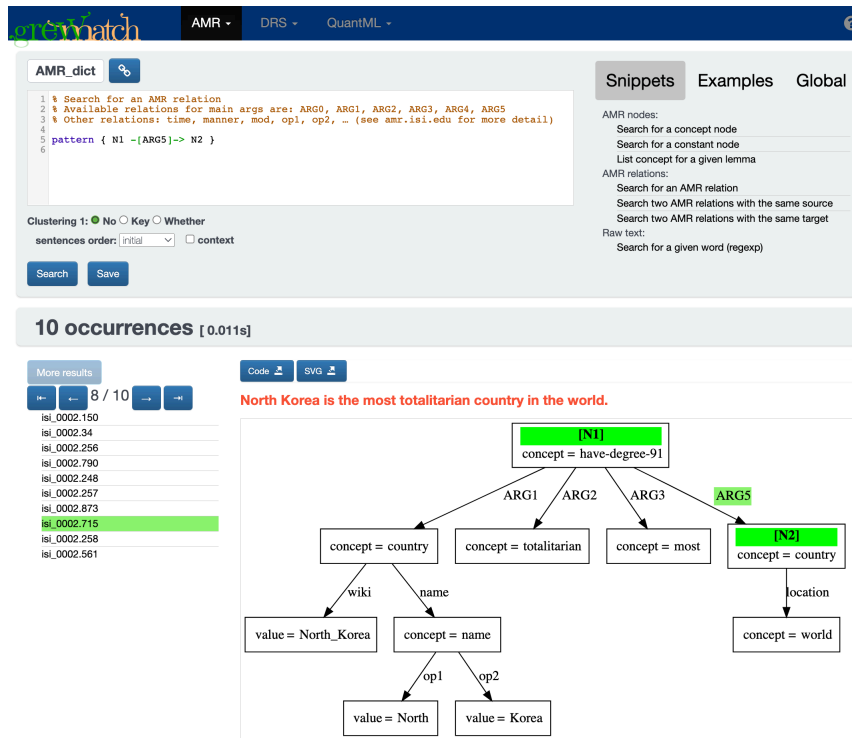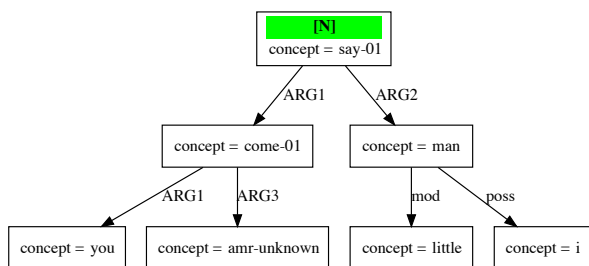
Figure 4: The GREW-MATCH tool

Figure 5: AMR annotation of the sentence *"My little man, where do you come from?*.

```
pattern {
  M [concept]; N [concept];
  e: M -> N
}
```

With the encoding of boxes we have described above, it is possible to request for specific configurations of boxes. The next pattern corresponds to two nested negations.

```
pattern {
  B1 -[NEGATION]-> B2;
  B2 -[NEGATION]-> B3
}
```

With this query, in Figure 7, we can observe a perfect illustration of the encoding of universal quantification through a double negation.[7]

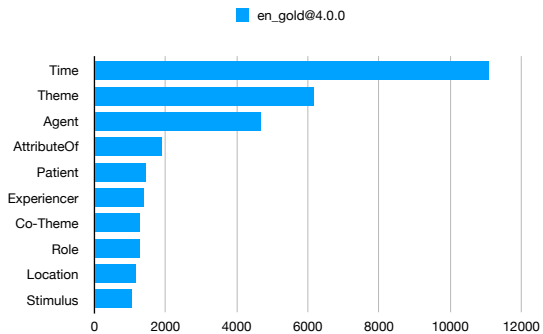
[7]In PMB, TPR stands for *temporal precedence.*

Figure 6: distribution of the ten most frequent semantic relations between two "concept" nodes

Global graph structure. More general queries about the graph structure allow for instance to check for cyclic structures.

```
global { is_cyclic }
```

The AMR guidelines say "Approximately 0.3% of AMRs are legitimately cyclic"[8]. But, with the query above, we can report the ratio of cyclic structures in AMR corpora: more than 3% in The Little Prince Corpus and almost 6.9% in the BioAMR corpus.

On the gold data of the PMB, cyclic structure are rare: there are 34 cyclic structures in English (among 10,715 sentences) and 1 in German (in 2,844 sentences). There are no examples in Italian or Dutch data.
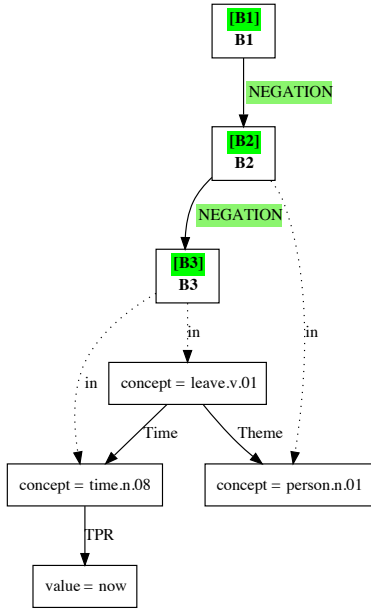
---

[8]https://github.com/amrisi/
amr-guidelines/blob/master/amr.md

Figure 7: Double negation encoding for the sentence [p18/d1454] *Everybody left*.

4. Error mining

GREW-MATCH can be used to detect inconsistencies in the annotations. A query is designed to express a constraint which should be respected by all the annotated structures. Such a query is supposed to return an empty set of occurrences. If it is not the case, we can observe the exceptions given. These can be annotation errors or if the annotation is legitimate, the query should be refined to take into account these cases. It also helps to find missing information in the guidelines where some cases are not recorded.

We give below a few examples of such usage of GREW-MATCH for inconsistencies detection.

In AMR structures, according to the guidelines, each named entity, is annotated with a node whose concept expresses the kind of entity (*Person*, *City*. . .) and with two outgoing edges labeled **name** and **wiki**. With the following pattern, we can search for nodes with an outgoing edge **name** and without an **wiki** edge, and spot inconsistent annotations.

```
pattern { M -[name]-> N }
without { M -[wiki]-> * }
```

This pattern returns one occurrence in the data from the AMR Annotation Dictionary[9] where the city name *New Orleans* is not associated with its wikipedia page. We can also report that the BioAMR Corpus is not consistently annotated in this respect: 95% of **name** edges appear without a **wiki** edge.

On the PMB, we can use the following pattern to observe structures where the same entity (node E) is both the **Agent** and the **Patient** of the same predicate P.

9[https://www.isi.edu/~ulf/amr/lib/amr-dict.html](https://www.isi.edu/~ulf/amr/lib/), consulted on 2022/03/31

```
pattern {
  P -[Agent]-> E;
  P -[Patient]-> E;
}
```

On the English gold data (10,715 sentences), 20 occurrences are returned. In 15 cases, the pattern is legitimate (sentences with *himself, herself. . .*) but the 5 remaining cases are annotation errors: for instance, [p60/d0784] *Betty killed her mother.* or [p62/d1397] *He was seduced by Tom.* (see Figure 8 for this last sentence).
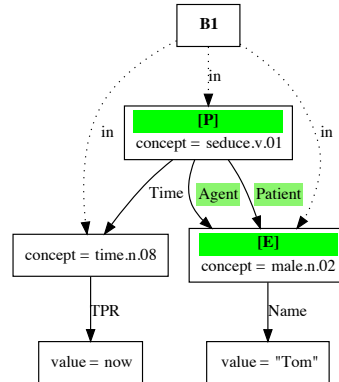

Figure 8: PMB annotation of the sentence [p62/d1397] *He was seduced by Tom*.

With the pattern already given in section 3 for double negation, we retrieve also the example given in Figure 9 where the structure is not the one expected. The two internal boxes should be at the same level and not embedded as in the figure. In fact, in the clause notation of the PMB (the original notation from which the SBN notation is extracted), the sentence is correctly annotated. We have indeed found a bug in the conversion process for the SBN notation which has been reported to the PMB maintainers.

On QuantML, the number of available annotated sentences is really tiny: 11 sentences in the TiCC report and 7 sentences in Amblard et al. (2021). Hence, sentences can be checked one by one without using queries; nevertheless, having a graph visualisation of this annotation was useful. When working on the ISA-17 shared task, we started producing the graphs for the examples in the guidelines and we discovered some inconsistencies. In Figure 10, we present the wrong graph of one example and the corrected version of the same annotation.

5. Conclusion

Semantic structures are often complex and represent several different levels of information in the same structure. It is then very useful to provide graphical visualisation in order to assist the humans who have to work with these data, either as annotators or as users. With the GREW-MATCH tool, we propose to use the mathematical model of graphs as a common way to
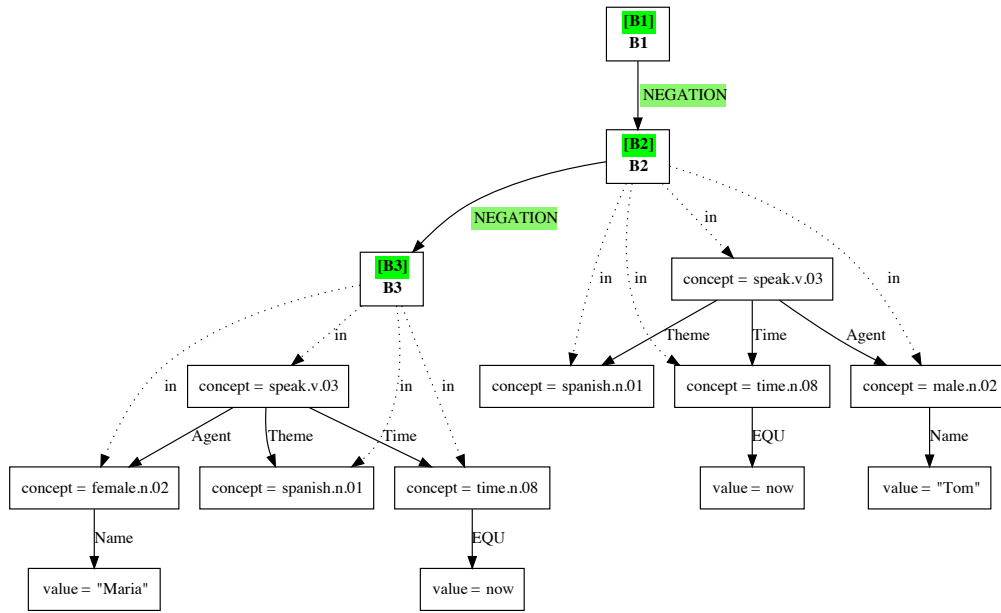
Figure 9: SBN annotation in the PMB of the German sentence `[p38/d2263]` *Weder Tom noch Maria sprechen Spanisch* ('*Neither Tom nor Mary speaks Spanish.*').

represent the semantically annotated data in various frameworks. Doing this, we have the possibility to visualise the annotations but also to use the graph query languages provided to make request on corpora of annotated sentences.

Querying graphs with GREW-MATCH has been useful to make linguistic observations on the data or to check the consistency of the data and the conformity with the guidelines. When inconsistencies are reported, it helps finding how the data or the guidelines (or some other tool) should be improved. GREW-MATCH can also be used as a side-tool when doing annotation, which helps finding similar examples in the already annotated data and thus helps annotators to take consistent decisions for similar constructions. We would like to recommend to use the methodology presented in this paper, based on graph visualisation and graph querying as a non regression evaluation tool for any framework.

In future work, we plan to consider other semantic annotations frameworks like UCCA (Abend and Rappoport, 2013) or DMRS (Copestake, 2009) for instance for which a graph based visualisation and querying would probably be useful as well.

6. Acknowledgements

7. Bibliographical References

Abend, O. and Rappoport, A. (2013). Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association*
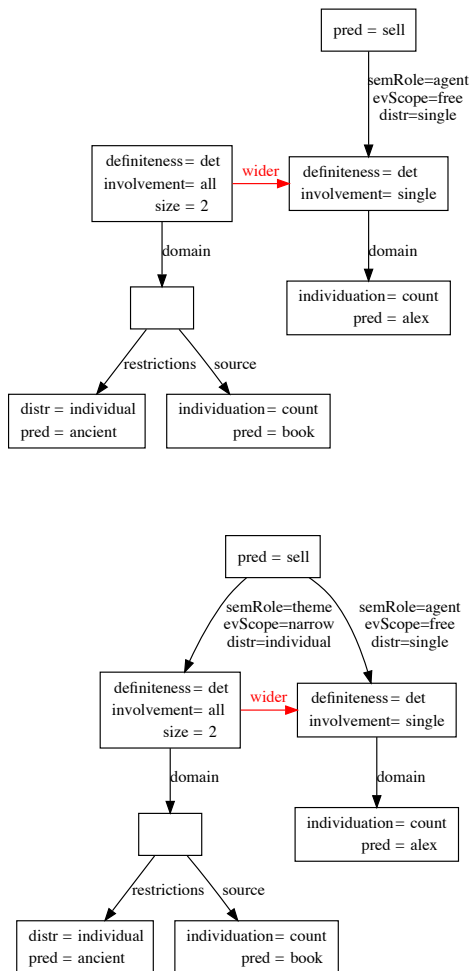

Figure 10: QuantML annotation of the sentence `[A7]` *Alex sold the two ancient books* in the TiCC report (top) and the corrected annotation (bottom).

for Computational Linguistics (Volume 1: Long Papers), pages 228–238, Sofia, Bulgaria, August. Association for Computational Linguistics.

Abzianidze, L., Bjerva, J., Evang, K., Haagsma, H., van Noord, R., Ludmann, P., Nguyen, D.-D., and Bos, J. (2017). The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations. In *15th Conference of the European Chapter of the Association for Computational Linguistics, Volume 2, Short Papers*, pages 242 – 247, Valencia, Spain, April.

Amblard, M., Fort, K., Guillaume, B., de Groote, P., Li, C., Ludmann, P., Musiol, M., Pavlova, S., Perrier, G., and Pogodalla, S. (2021). The Annotators Did Not Agree on Some of the Guidelines Examples. In *ISA 2021 - 17th Workshop on Interoperable Semantic Annotation*, Groningen/Virtuel, Netherlands, June.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Bonfante, G., Guillaume, B., and Perrier, G. (2018). *Application of Graph Rewriting to Natural Language Processing*, volume 1 of *Logic, Linguistics and Computer Science Set*. ISTE Wiley, April.

Bos, J. (2021a). Quantification Annotation in Discourse Representation Theory. In *ISA 2021 - 17th Workshop on Interoperable Semantic Annotation*, Groningen/Virtuel, Netherlands, June.

Bos, J. (2021b). Variable-free discourse representation structures. `https://semanticsarchive.net/Archive/jQzMzJlY/compact.pdf`.

Bunt, H., Pustejovsky, J., and Lee, K. (2018). Towards an ISO standard for the annotation of quantification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Bunt, H. (2020). Semantic annotation of quantification in natural language. Technical report, TiCC Technical Report 2020-2. Tilburg Center for Cognition and Communication and Department of Cognitive Science and Artificial Intelligence, Tilburg University.

Cohen, J., Cohen, R., Toledo, E., and Buys, J. (2021). RepGraph: Visualising and analysing meaning representation graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–86, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Copestake, A. (2009). Invited talk: Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceed-ings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 1–9, Athens, Greece, March. Association for Computational Linguistics.

Guillaume, B. (2021). Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics*, Kiev/Online, Ukraine, April.

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