Graph Querying for Semantic Annotations
Maxime Amblard, Bruno Guillaume, Siyana Pavlova, Guy Perrier

To cite this version:
Maxime Amblard, Bruno Guillaume, Siyana Pavlova, Guy Perrier. Graph Querying for Semantic Annotations. ISA 2022 : 18th Joint ACL - ISO Workshop on Interoperable Semantic Annotation, Jun 2022, Marseille, France. hal-03724068

HAL Id: hal-03724068
https://inria.hal.science/hal-03724068
Submitted on 22 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 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 (1,562 sentences) and BioAMR Corpus version 3.0 (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) 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

http://semantics.grew.fr

https://amr.isi.edu/download/amr-bank-struct-v3.0.txt
http://pmb.let.rug.nl/
You are like my fox when I first knew him.

Figure 1: AMR annotation in Penman notation and its interpretation as graph of the sentence.

Figure 2: PMB annotation in SBN (Simplified Box Notation) and its interpretation as graph of the sentence.

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.

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 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 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.

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).
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. 

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 \( P \): 

\[
\text{pattern } \{ \text{N [concept = "and"]} \}
\]

and the sub-pattern \( \text{N -[op1]-> X} \), 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 \( \text{N -[op2]-> X} \), 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 \( N . \text{concept} \)

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

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”. 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.

In PMB, TPR stands for temporal precedence.

---

https://github.com/amrisi/amr-guidelines/blob/master/amr.md
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 a 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 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.

```
{ 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, "Betty killed her mother. or 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 \([p38/d2263]\) \textit{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 G\textsc{rew-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. G\textsc{rew-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 Rap\textsc{poort, 2013}] or DMRS [Copestake, 2009] for instance for which a graph based visualisation and querying would probably be useful as well.

6. Acknowledgements

This work was partially supported by the ANR fund (ANR-20-THIA-0010-01). We would like to thanks Clara Serruau for her preliminary work on the subject and the reviewers for their comments and suggestions.

7. Bibliographical References


