metAMoRphosED: a graphical editor for Abstract Meaning Representation

Johannes Heinecke
Orange Innovation
2 avenue Pierre Marzin
F-22300 Lannion
johannes.heinecke@orange.com

Abstract

This paper presents a graphical editor for directed graphs, serialised in the PENMAN format, as used for annotations in Abstract Meaning Representation (AMR). The tool supports creation and modification of AMR graphs and other directed graphs, addition and deletion of instances, edges and literals, renaming of concepts, relations and literals, setting a "top node" and validation of the edited graph.

1 Introduction

Abstract Meaning Representation (AMR) is a semantic representation language designed to formalise the meaning of sentences or a set of sentences (Banarescu et al., 2013)[1]. Its motivation is to annotate semantic information like named entities, coreferences, word senses, semantic relations etc. However, it does not annotate meaning of natural language at the same degree as more complex frameworks such as the Discourse Representation Theory (Kamp and Reyle, 1993), as it does not mark number, semantic time, mode etc. Even though AMR has been explicitely devised for English and must not be considered as an interlingua, AMR is increasingly used to annotate sentences in languages other than English (Damonte and Cohen, 2018; Blloshmi et al., 2020; Uhrig et al., 2021; Heinecke and Shimorina, 2022). AMR graphs are directed graphs which contain concepts, instances, literals and labelled edges between instances and literals.

AMR uses concepts from PropBank (Kingsbury and Palmer, 2002; Palmer et al., 2005) where available (mainly verbal concepts), e.g., *bear-02* in figure 1, PropBank sense 2 for *bear*. Instances are indicated by a following "/", e.g., *p* being an instance of the concept *person*. `:ARG1` etc. mark relations. Literals (strings and numbers) lack an preceeding instance and "/" (c.f., *"Queen"* and *1926* in the example in figure 1).

```
(b / bear-02
    :ARG1 (p / person
        :name (n / name
            :op1 "Queen"
            :op2 "Elizabeth"))
    :time (d / date-entity
        :year 1926))
```

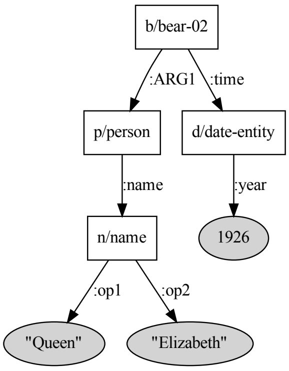


Figure 1: AMR graph for "Queen Elizabeth was born in 1926" in PENMAN format (above) and graphical visualisation

AMR data is available at the Linguistic Data Consortium (LDC) for English 2 :

- LDC2020T02: LDC general release AMR 3.0 (2020), with 59,255 sentences;
- LDC2017T10: LDC general release AMR 2.0 (2017), with 39,260 sentences.

The sentences of the test corpus of AMR 2.0 were translated by human translators into four languages (LDC2020T07: AMR 2.0, four translations of AMR 2.0 test set into Italian, Spanish, German, Chinese, 1371 sentences per language).[2]

[1] See also the project web site https://amr.isi.edu

[2] Corpora available at https://amr.isi.edu/download.html

Proceedings of the 19th Joint ACL - ISO Workshop on Interoperable Semantic Annotation, pages 27–32
June 20, 2023. ©2023 Association for Computational Linguistics

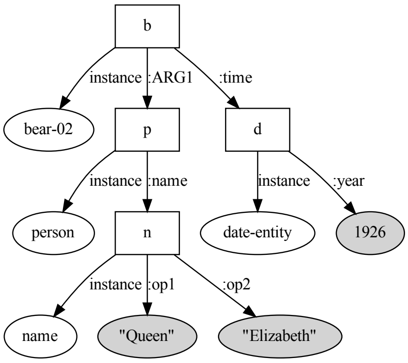


Figure 2: AMR graph of figure 1 with instances explicitely visualised

```
# ::id lpp_1943.293 ::date 2012-11-18...
# ::snt I answered , "eats anything ...
# ::save-date Thu Apr 18, 2013 ...
(a / answer-01
   :ARG0 (i / i)
   :ARG1 (e / eat-01
      :ARG1 (a2 / anything
         :ARG1-of (f / find-01
            :ARG0 (i2 / it)
            :location (r / reach-03
               :ARG0 i2)))))

# ::id lpp_1943.294 ::date 2012-11-18...
# ::snt "Even flowers that have thorns ?"
# ::save-date Thu Oct 29, 2015 ...
(f / flower :mode interrogative
   :mod (e / even)
   :ARG0-of (h / have-03
      :ARG1 (t / thorn)))
```

Figure 3: Example of two sentences (slightly truncated for place reasons) in an AMR file (taken from *The Little Prince* corpus, available at the AMR project website)

Nearly all available annotated AMR corpora use the PENMAN graph serialisation format (Kaspar, 1989; shown in figure 1 together with a graphical representation where instances and concepts are shown in one rectangle for better readability, the full visualisation of figure 1 would be the visualisation in figure 2).

In addition to the PENMAN serialisation, typical AMR files contain some metadata too: the sentence itself, translations, a unique sentence identification, annotator identification, saving date, named entities, etc, e.g., figure 3.

Since the AMR graph is not anchored, i.e., there is no obvious link between words of the sentences and concepts, instances and relations in the graph, annotation of a corpus using a simple text editor is not possible. Apart from the parentheses it would be very difficult to check manually whether the concepts are correctly chosen and the arguments (notably `:ARG0` to `:ARG9`) defined for the chose concepts. The corpora mentioned in the AMR project website have mostly been annotated and validated using the AMR editor (Hermjakob, 2013) at https://amr.isi.edu/editor.html. Since this tool is not available for download, and we wanted to annotate a specialised evaluation corpus, we started developping metAMoRphosED. Our aim was a utility easy to use for annotators without any profound knowledge of semantic graphs, PENMAN format or triplets and providing as much assistance to the annotators as possible.

2 Architecture

metAMoRphosED is a webserver (implemented in python), the graphical user interface (GUI; implemented using html, css and javascript) is accessible with an internet browser. The server will handle the AMR file and optionally additional validation information like PropBank-data, a list of valid relations or a file which defines the relations an instance of which concepts can have. Start the server with

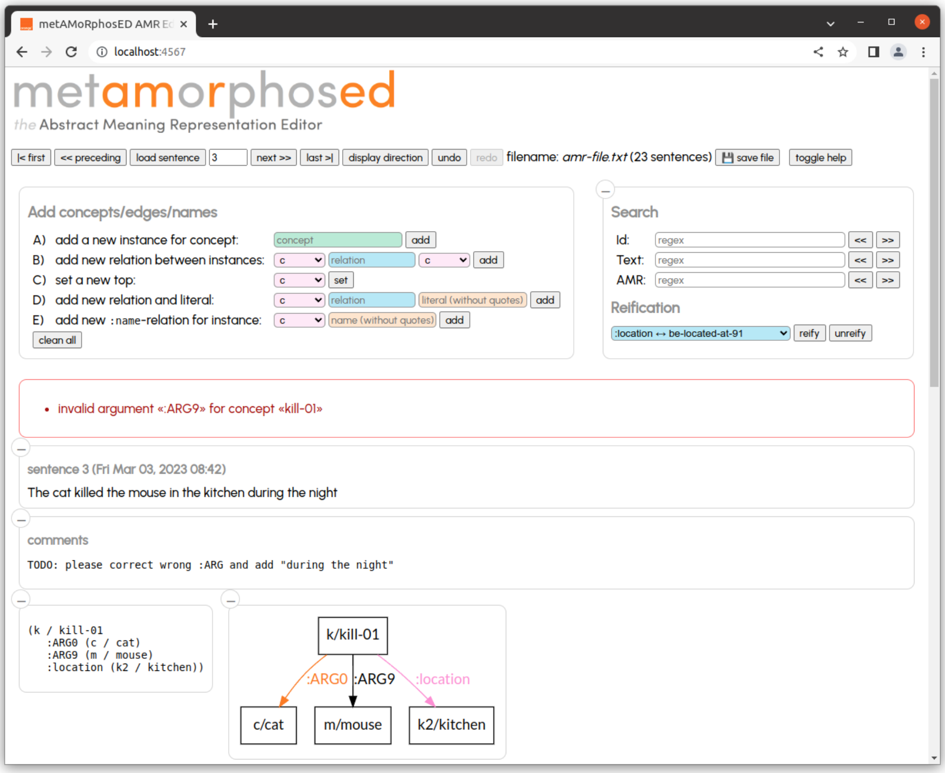
```
server.py --file amr-file.txt \
     --pbframes propbank-frames/frames \
     --reification reification-table.txt \
     --relations amr-relation-list.txt \
     --concepts amr-concept-list.txt \
     --constraints constraints.yml
```

and point your internet browser to `http://localhost:4567`. Once you have clicked on load sentence, a view similar to the one in figure 4 appears. In addition to the sentence, the PENMAN serialisation and the visualisation, the graphical user interface shows the PropBank documentation of all the verbs found in the graph, and possible errors.

2.1 Graph validation

In order to help the annotators metAMoRphosED can load AMR-related data to find potential annotation errors. metAMoRphosED will not modify a graph on its own without user approval, but it will issue warnings.

- concept definitions: since metAMoRphosED has been primarily developed to edit AMR graphs, it can load PropBank data to validate `:ARGn` relations of verbal concepts. In order to do so the option `--pbframes <propbank frames directory>` can be used



load sentence: 3

filename: *amr-file.txt* (23 sentences)

Reification: :location ↔ be-located-at-91

- invalid argument «:ARG9» for concept «kill-01»

sentence 3 (Fri Mar 03, 2023 08:42)

The cat killed the mouse in the kitchen during the night

comments

```
TODO: please correct wrong :ARG and add "during the night"
```

```
(k / kill-01
   :ARG0 (c / cat)
   :ARG9 (m / mouse)
   :location (k2 / kitchen))
```

Figure 4: Initial screen with a sentence loaded, comments and an error message

to point to the `frames/` directory PropBank (available at https://github.com/propbank/propbank-frames). In addition to validation, metAMoRphosED will show all senses of all verbal concepts in the current graph (cf. figure 5).

- valid relations: the option `--relations <filename>` accepts a simple text file which contains a list of all valid relations (including inverted `-of` relations. If a graph contains a relation not in the list, a warning is given. Duplicated relations between two instances (e.g., two `:ARG0` relations) are also indicated as an error.

  The editor verifies that instances with outgoing `:op`*n* or `:snt`*n* relations, metAMoRphosED have a correct sequence of `:op1` to `:op`*n* without any missing number.

- relation constraints: a more specific way of limiting the possible range values of relations comes with the option `--constraints <constraints.yml>`, for instance:

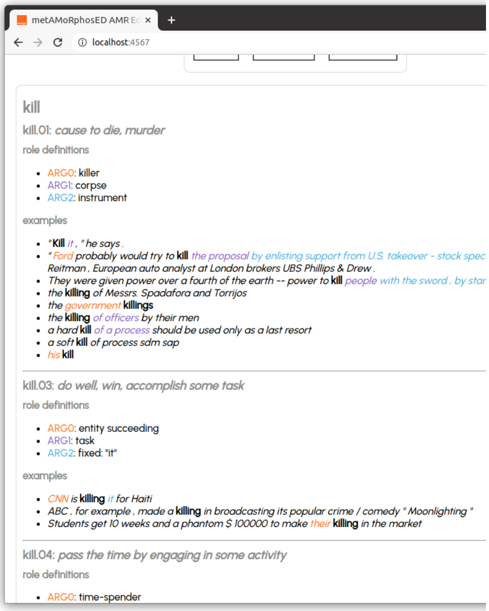
```
# constraints for domain/relation/range
subjects:
# name-instances can only have :opn relations,
# which in turn have quoted strings as ranges
# (an initial _ indicates that the predicate
# or object is a regex)
name:
  _:op\d:
    - _".*"

# date-entity instances must only have
# :month, :day and :year predicates with
# integer values or :dayperiod with any value
date-entity:
  :month:
    - _[01]\d?
  :day:
    - _[0-3]\d?
  :year:
    - _\d\d\d\d
  :dayperiod:
```

- reification: the AMR documentation lists a set of relations which can be reified, metAMoRphosED proposes a function for this (cf. figure 6), which can be activated using the option `--reifications <table>`[3].

2.2 Non-AMR data

metAMoRphosED is not confined to AMR-data only. As long as the data to be edited can be represented

[3] See https://github.com/amrisi/amr-guidelines/blob/master/amr.md for more details on reification in AMR.



metAMoRphosED AMR Ed ×

localhost:4567

kill

kill.01: *cause to die, murder*

role definitions

- ARG0: killer
- ARG1: corpse
- ARG2: instrument

examples

- *"***Kill** *it , " he says .*
- *" Ford probably would try to* **kill** *the proposal by enlisting support from U.S. takeover - stock spec Reitman , European auto analyst at London brokers UBS Phillips & Drew .*
- *They were given power over a fourth of the earth -- power to* **kill** *people with the sword , by star*
- *the* **killing** *of Messrs. Spadafora and Torrijos*
- *the government* **killings**
- *the* **killing** *of officers by their men*
- *a hard* **kill** *of a process should be used only as a last resort*
- *a soft* **kill** *of process sdm sap*
- *his* **kill**

kill.03: *do well, win, accomplish some task*

role definitions

- ARG0: entity succeeding
- ARG1: task
- ARG2: fixed: "it"

examples

- *CNN is* **killing** *it for Haiti*
- *ABC , for example , made a* **killing** *in broadcasting its popular crime / comedy " Moonlighting "*
- *Students get 10 weeks and a phantom $ 100000 to make their* **killing** *in the market*

kill.04: *pass the time by engaging in some activity*

role definitions

- ARG0: time-spender

Figure 5: PropBank documentation (clipped)

using the PENMAN format the tool is able to process it. I.e. the data must contain concepts, instances, attributes (literals) and directed relations between them. This means that pure RDF (in contrast to RDFS) can not be annotated, neither can data taken from wikidata be edited directly due to the qualifiers, i.e. triples with an property in subject position.

However data like the MultiWOZ corpus can be transformed into PENMAN and than be edited by metAMoRphosED (Abrougui et al., 2023).

3 Editing functions

metAMoRphosED can create new graphs from scratch (in this case, the AMR-file must contain at least `()`) or can be used to modify existing graphs (possibly generated by an AMR parser). Apart from graphical operations a direct modification of the PENMEN serialisation is possible. After every modification the current version of the graph is visualised. Navigation within the current file is possible by giving the sentence number, navigation buttons (first, last, next, preceding) or search functions (sentence id, sentence text, PENMAN, cf. figure 6) with highlighted results.

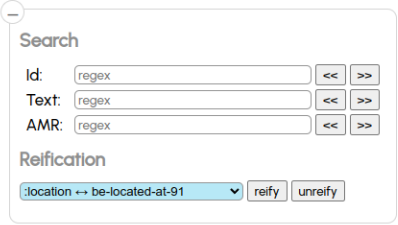
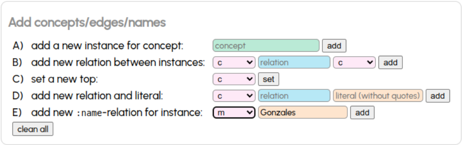


Figure 6: search and reification/dereification

In order to add new instances or literals and new relations between new and existing instances, an input form is provided in the GUI (figure 7). Existing data can be modified or deleted by clicking on instances or relations in the graph visualisation (cf. figures 8 and 9). In case of an unwanted modification an *undo* function exists to revert the graph to the preceding version.



Add concepts/edges/names

A) add a new instance for concept: concept add

B) add new relation between instances: c relation c add

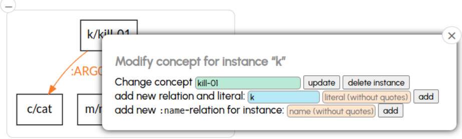
C) set a new top: c set

D) add new relation and literal: c relation literal (without quotes) add

E) add new `:name`-relation for instance: m Gonzales add

clean all

Figure 7: add menu



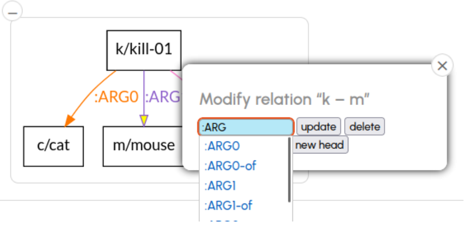
Modify concept for instance "k"

Change concept kill-01 update delete instance

add new relation and literal: k literal (without quotes) add

add new `:name`-relation for instance: name (without quotes) add

Figure 8: modify a concept



Modify relation "k – m"

:ARG update delete

new head

Figure 9: modify a relation

In AMR the entry point of a graph contains topic information, in order to change the entry point a *set top* function is provided.

As mentioned above, if metAMoRphosED was started with the option `--reification`, all relations listed correctly in the loaded reification table can be automatically reified (and unreified if no additional relation exists). So for instance the graph shown in figure 4 is transformed into the graph of figure 10.

If the loaded AMR file is under git version control, clicking the save file button also performs a `git add`/`git commit`.

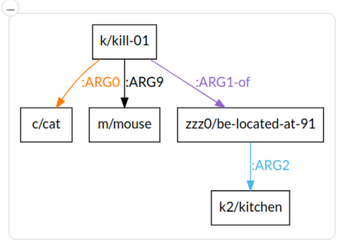


Figure 10: relation `:location` reeified

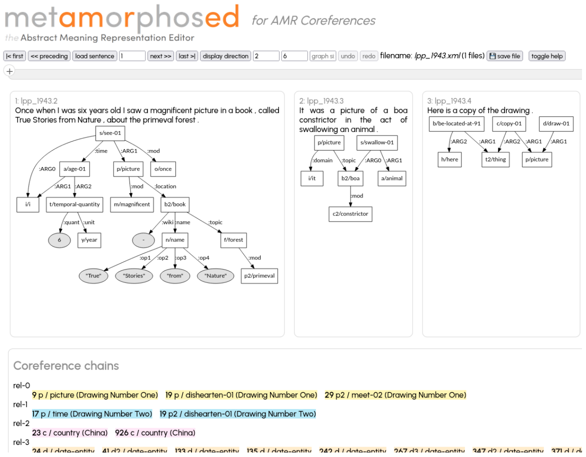
4 Comparison

Since the existing AMR editor (Hermjakob, 2013) is not available for download, we were not able to compare the anotation speed and ergonomics of metAMoRphosED and ISI's editor. Since AMR data is stored in PENMAN format, some might find it more difficult to "understand" than a graphical representation. In general, annotation speed depends mainly of the competence and experience of the annotators and much less on the tool. However the graphical representaion which metAMoRphosED proposes, makes it easier for annotators new to AMR.

5 Conclusion and prospectives

We presented a novel graph editor, suitable to create or modify and validate Abstract Meaning Representation graphs in a visual mode. All modifications are git-version controlable. The code is actively maintained and available at https://github.com/Orange-Opensource/metamorphosed. metAMoRphosED is currently used to annotate a test corpus containing 400 questions and turned out to be stable.

Apart from an interface to annotate AMR coreferences, which has been implemented and is being tested (cf. figure 11) we plan several future developments, notably a multi-user system, where multiple users can annotate the same file with integrated calculation of annotator agreement. Another improvement could be an automatic search of similar sentences in a reference corpus (such as AMR 3.0) to ensure that similar constructions and sentences are annotated homogenously. Other ideas include providing a way to integrate plugins which could run queries to external systems to facilitate the annotation as much as possible. We will consider comments and issues posted by users to decide which feature is the most urgent to be implemented first.



1 | 2 | 6

filename: lpp_1943.xml (1 files)

1: lpp_1943.2
Once when I was six years old I saw a magnificent picture in a book , called True Stories from Nature , about the primeval forest .

2: lpp_1943.3
It was a picture of a boa constrictor in the act of swallowing an animal .

3: lpp_1943.4
Here is a copy of the drawing .

Coreference chains

rel-0
9 p / picture (Drawing Number One) 19 p / dishearten-01 (Drawing Number One) 29 p2 / meet-02 (Drawing Number One)
rel-1
17 p / time (Drawing Number Two) 19 p2 / dishearten-01 (Drawing Number Two)
rel-2
23 c / country (China) 926 c / country (China)
rel-3
24 d / date-entity 41 d2 / date-entity 133 d / date-entity 135 d / date-entity 242 d / date-entity 267 d3 / date-entity 347 d2 / date-entity 371 d / d

Figure 11: AMR coreference editor interface (clipped)

Acknowledgments

We'd like to thank the anonymous reviewers for comments and ideas to improve this paper.

References

Rim Abrougui, Géraldine Damnati, Johannes Heinecke, and Frédéric Béchet. 2023. Abstract Representation for Multi-Intent Spoken Language Understanding. In *International Conference on Acoustics, Speech, and Signal Processing*, Rhodes Island, Greece. IEEE.

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, Sofia, Bulgaria. Association for Computational Linguistics.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling Cross-Lingual AMR Parsing with Transfer Learning Techniques. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 2487–2500, Online. Association for Computational Linguistics.

Marco Damonte and Shay B. Cohen. 2018. Cross-lingual Abstract Meaning Representation Parsing. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1146–1155, New Orleans, Lousiana, USA. Association for Computational Linguistics.

Johannes Heinecke and Anastasia Shimorina. 2022. Multilingual Abstract Meaning Representation for Celtic Languages. In *Proceedings of the 4th Celtic Language Technology Workshop within LREC2022*, pages 1–6, Marseille. ELRA.

Ulf Hermjakob. 2013. AMR Editor: A Tool to Build Abstract Meaning Representations. https://amr.isi.edu/papers/amr-editor-ulf2013a.pdf.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Studies in Linguistics and Philosophy 42. Kluwer, Dordrecht.

Robert T. Kaspar. 1989. A Flexible Interface for Linking Applications to Penman's Sentence Generator. In *Proceedings of the Workshop on Speech and Natural Language*, pages 153–158, Philadelphia, PA. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From Tree-Bank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, Las Palmas, Canary Islands - Spain. European Language Resources Association.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Sarah Uhrig, Yoalli Rezepka García, Juri Opits, and Anette Frank. 2021. Translate, then Parse! A strong baseline for Cross-Lingual AMR Parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 58–64, Online. Association for Computational Linguistics.