Querying Multi-word Expressions Annotation with CQL

Natalia Klyueva
The Hong Kong Polytechnic University
Hong Kong
natalia.klyueva@polyu.edu.hk

Anna Vernerová
Charles University
Prague, Czech Republic
vernerova@ufal.mff.cuni.cz

Behrang QasemiZadeh
Heinrich-Heine University
Düsseldorf, Germany
zadeh@phil.hhu.de

Abstract

This paper demonstrates a solution for querying corpora with multi-word expression (MWE) annotation using a concordance system. Namely, the PARSEME multilingual corpora, which contain manually annotated verbal multi-word expression (VMWE) in 18 languages, are converted to a suitable *vertical format* so that they can be explored using the Corpus Query Language (CQL). VMWEs encompass a range of categories such as idioms, light verb constructions, verb-particle constructions, and so on. Although these corpora were mainly developed for the purpose of developing automatic methods for the identification of VMWEs, we believe they are a valuable source of information for corpus based studies. The solution proposed in this paper is an attempt to provide a linguist/non-tech-savvy friendly method for exploring these corpora. We show how CQL-enabled concordancers such as NoSke or KonText can be exploited for this purpose. Despite several limitations, such as problems related to discontinuous and coordinated MWEs, CQL still is an enabling tool for basic analysis of MWE-annotated data in corpus-based studies.

1 Introduction

Multi-word expressions (MWEs) are structures that cross word boundaries and thus present challenges to a variety of NLP tasks such as syntactic analysis and machine translation, etc. The PARSEME (PARSing and Multi-word Expressions) EU COST action (Savary et al., 2015) addressed these problems by organizing a shared task on automatic identification of verbal MWEs (Savary et al., 2017)[1]. Verbal MWEs (VMWEs) were annotated in corpora in 18 languages according to common guidelines (Candito et al., 2017). Several categories of VMWEs were defined: idioms (ID), light verb constructions (LVC), inherently reflexive verbs (IReflV), verb-particle constructions (VPC), and OTH (other).

The data for each of the languages is distributed in two files: **.conllu** and **.parsemetsv**. The first one contains morphosyntactic annotation in the CoNLL-U format[2] and the second one contains the MWE annotation in the parseme-tsv format (example follows). These representation are mainly optimized for machine readability and particularly for training predictive models. The data in these two files is not presented in an intuitive and suitable form for search and retrieval scenarios involving human users. We present one of the approaches that can be used for this purpose (i.e., as a query system for MWE annotated corpora). For example, one may easily retrieve frequency lists of MWEs and study them as key words in context.

The problem we face is not well-studied, though we can find some works related to the topic. Klyueva and Straňák (2016) introduce a mechanism for basic queries over syntactic trees by copying selected attributes of a node's parent to attributes of the child node itself, e.g. `p_form`, `p_lemma`, `p_tag`. A corpus with terminology (above all, multi-word terminology) annotation was represented in a vertical format with structural attributes used for encoding MWEs (QasemiZadeh and Schumann, 2016)[3]. Another web service that allows to query for MWEs is based on CQPWeb[4], but the texts as well as the manual are only

[1] http://multiword.sourceforge.net/sharedtask2017
[2] http://universaldependencies.org/format.html
[3] Online search at http://lindat.mff.cuni.cz/services/kontext/first_form?corpname=aclrd20_en_a.
[4] http://yeda.cs.technion.ac.il/HebrewCqpWeb/

Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16), pages 73–79,
Prague, Czech Republic, January 23–24, 2018.

Keywords: verbal multiword expressions, Sketch Engine, corpus query

in Hebrew which did not let us make a full study of the functionality. A survey of MWEs in treebanks is presented by Rosén et al. (2015), but the paper does not contain any directions on how to access and query MWEs.

This paper is structured as follows. Section 2 is devoted to corpus query systems suitable to querying corpora with annotation of MWEs. Section 3 introduces the data format in which the data is distributed and the necessary conversion to the vertical format behind CQL. Example queries can be found in Section 4.

Our conversion scripts[5] are intended for a subgroup of 15 PARSEME corpora that also feature syntactic annotation, thus excluding Bulgarian, Hebrew and Lithuanian which are distributed without a **.conllu** file.

2 Corpus Query Systems

Tools to search corpora—corpus query systems—present a powerful and popular concept for digital humanities. We can distinguish several types of engines depending on their functionalities. The first group treats text in a linear manner (as a string of annotated words), e.g. SketchEngine[6] (Rychlý, 2007) or IMS Corpus Workbench[7] (Evert and Hardie, 2011); the second group sees text as a group of trees, e.g. PML-TQ[8] (Štěpánek and Pajas, 2010), INESS[9] (Rosén et al., 2012) or Tündra[10] (Martens, 2013). While the first group of tools is easier to maintain, and from the user's point of view the query language (CQL/CQP) is simpler than that of the second group, the treebank query languages of the tools in the second group have much greater expressive power.

Concerning data preparation and compilation, treebank query systems use much more complex data formats, e.g. the native format of PML-TQ is an XML-based format called PML (Štěpánek and Pajas, 2010); for the web search, the data is indexed using a relational database and high level queries are internally translated to SQL. In contrast, the vertical format required for concordance systems such as Sketch Engine or IMS almost corresponds to the original format of our corpora and requires less effort to make them available for search and retrieval.

Multi-word units pose problems for both categories of corpus querying tools since in both paradigms the basic unit that carries annotation is a token. In this paper, we work with the open source corpus management system Manatee that applies the linear paradigm; the suggested representation of data can then be exploited through either of the two open-source front-ends for Manatee, i.e. either through the NoSke[11] (a free edition of the Sketch Engine) or through KonText[12] (a front-end developed by the Institute of the Czech National Corpus based on NoSke); the PARSEME corpora in this paper are available via both platforms.

3 Vertical Encoding of the Data

As mentioned earlier, the PARSEME corpora come in two files in two different formats for each language, one with morphosyntactic annotations (**.conllu**) and another with MWE annotation (**.parsemetsv**). Both formats represent a challenge to the 'linear'-based corpus management tools as they contain hierarchal or graph annotations (e.g., syntactic dependencies in CoNLL-U and discontinuous structures in parseme-tsv). In order to provide unified querying over both morphosyntactic and MWE annotations, we combine these two resources into a single file.

The following is a sentence fragment in the **.parsemetsv** format:[13]

[5] https://github.com/natalink/mwe_noske
[6] http://sketchengine.co.uk
[7] http://cwb.sourceforge.net
[8] http://hdl.handle.net/11858/00-097C-0000-0022-C7F6-3
[9] http://iness.uib.no
[10] http://weblicht.sfs.uni-tuebingen.de/Tundra
[11] https://nlp.fi.muni.cz/trac/noske, Parseme data http://corpora.phil.hhu.de/parseme
[12] https://github.com/czcorpus/kontext, Parseme data http://lindat.mff.cuni.cz/services/kontext
[13] The four columns contain the word id, the word form, information whether the token is followed by a space, and the MWE annotation.

```
1   Delegates  _  _
2   are        _  1:LVC
3   in         _  1
4   little     _  _
5   doubt      _  1
6   that       _  _
7   the        _  _
8   shadow     _  2:ID
9   cast       _  2
10  over       _  _
11  the        _  _
12  city       _  _
...
```

The fourth column encodes the MWE annotation of a token as follows. Tokens belonging to the same MWE are labeled with the same numerical identifier so that they can be distinguished as independent MWE unit in a sentence. The first token in a particular MWE is additionally labeled the category of the MWE. In case that a token belongs to several MWEs, the respective tags are separated by a semi-colon (e.g. `1:VPC;2:VPC`).

In our previous work (QasemiZadeh and Schumann, 2016), we have been able to encode MWEs by structural[14] attributes. In doing so, we were relying on annotation that was based on the largest span policy—there data did not annotate MWEs that are part of other MWEs, nor overlapping MWEs. However, in case of VMWEs, modeling overlapping structures is inevitable, and the use of structural attributes leads to complexities which can be avoided by the use of positional attributes. In our proposed format, the CoNLL-U attributes and the MWE annotations are both encoded as positional attributes (columns themselves).

We use the following attributes to encode MWEs:

- **mwe** specifies the type of the MWE, e.g. `LVC` for a light verb construction or `IReflV` for an inherently reflexive verb;
- **mwe_order** has two possible values, `first` for the first word in the MWE and `cont` for all remaining ("continuation") words;
- **mwe_id** gives the consecutive number of the MWE within the sentence; we shall show later how this attribute helps to distinguish overlapping MWEs;
- **mwe_lemma** is just a concatenation of the lemmas of all words that are part of the MWE, in the order in which they appear in the sentence, e.g. `be in doubt`.

In case one token is annotated as part of multiple MWEs, the MWE annotations attached to it are treated as *multivalue*. For instance, the sentence above will be represented as

```
1 Delegates Delegates NOUN  NNS Number=Plur                     5 nsubj _ _  _   _     _ _
2 are       be        AUX   VBP Mood=Ind|Tense=Pres|VerbForm=Fin 5 cop   _ _  LVC first 1 be in doubt
3 in        in        ADP   IN  _                                5 case  _ _  LVC cont  1 be in doubt
4 little    little    ADJ   JJ  Degree=Pos                       5 amod  _ _  _   _     _ _
5 doubt     doubt     NOUN  NN  Number=Sing                     0 root  _ _  LVC cont  1 be in doubt
6 that      that      SCONJ IN  _                                9 mark  _ _  _   _     _ _
```

The attributes to search for are named exactly as in the CoNLL-U scheme (e.g. **upostag**) with an exception for the word-form, which is called **word** in our concordance system instead of **form** as in CoNLL-U; they can be queried using the standard CQL syntax (see the screenshot from the UI on Figure 1).

4 Example Queries

In this section we provide basic examples showing how to query the PARSEME corpora using CQL queries. We concentrate on a few examples that we believe can be most helpful in several scenarios.

[14]Definitions of structural and positional attributes can be found at https://www.sketchengine.co.uk/corpus-configuration-file-all-features/.

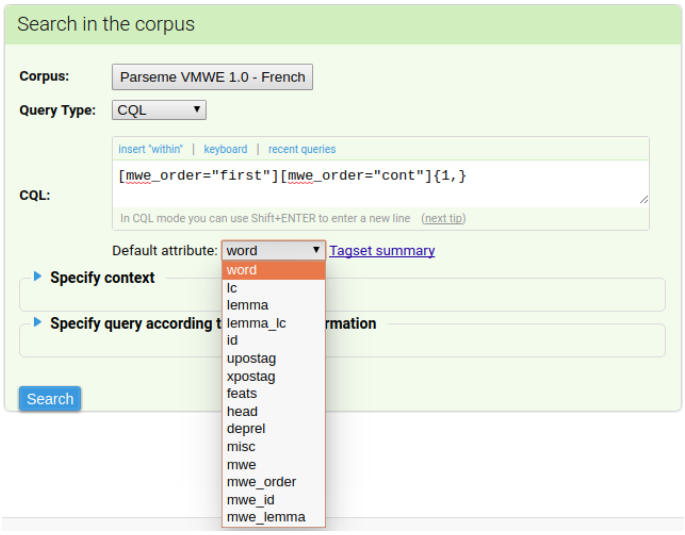

Figure 1: Search interface, attribute menu

The aim is to provide a range of examples to demonstrate both pros and cons of using CQL queries for exploring MWE annotated corpora, given our representation structure using positional attributes.

CQL queries are composed of blocks of the form of `[attribute="value"]`, in which the value expresses a condition over the given attribute. In its simplest form, a CQL query consists of only one pair of attribute-value, and the value is an exact string, e.g., `[word="test"]`, which in turn returns all the occurrences of the word-form *test* in the corpus under investigation. However, these building blocks can be concatenated to form a more complex query involving a sequence of one or more tokens. Additionally, the attribute values may be specified through regular expressions and simple logical operators such as 'and' (&) and or (|) are also available.

All queries in the following examples are linked to the KonText search tool and the result for them can be seen online by clicking on them. In our examples, we use the French, Spanish and German corpora; however, all the queries are valid for other languages in the PARSEME collection. These queries (and more) are also available in the online tutorial at https://ufal.mff.cuni.cz/lindat-kontext/parseme-mwe.

4.1 Continuous MWE Fragments

We start with a simple example:

```
[mwe_order="first"]
```

which results in a KWIC[15] view containing the first word of each MWE in the corpus. Note that if the same word happens to be the first word of several MWEs (such as the word *letting* in *they were letting us in and out*), it will appear in the KWIC output only once.

The query below will display and highlight continuous MWEs:

```
[mwe_order="first"][mwe_order="cont"]{1,}
```

In case of MWEs with more than two tokens, this resulting concordance contains the same location more than once. For example, for a 3 token MWE, the KWIC view contains two lines: one with two tokens highlighted and another – the same sentence – with three tokens, as displayed in Figure 2. One immediate solution to remove unwanted duplicates in the output is to use the so-called *overlaps/sub-hits filter*, which is supported in the NoSke system: only one of the matches is kept, whilst the other lines matching around the same position are omitted from the output.[16]

[15] key word in context

[16] Another solution is to use an additional positional attribute to specifically mark the last token of MWEs. In this case, the corresponding attribute–value pair can be added to the end of the proposed query.

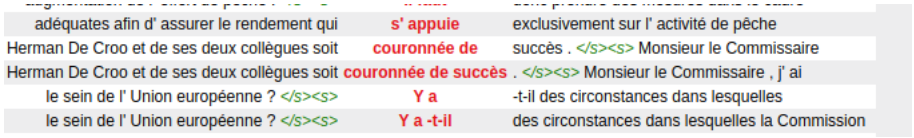

Figure 2: The same occurrence of MWE retrieved twice.

4.2 MWEs with Discontinuity

Intuitively when creating a query we want to see only tokens belonging to MWEs in the concordance. This is not straightforward in a 'linear' corpus query system and we can not reach the state when all MWEs will be highlighted as a whole in discontinuous constructions through only one interaction with the underlying corpus management system. In current implementations of KWIC, the intermediate words (not belonging to the MWE) will be highlighted as well in this case.

In order to show not only the first word of the MWE, but also its continuation, a more complex query that matches also the nodes in between must be executed:

```
1:[mwe_order="first"] []* 2:[mwe_order="cont"] & 1.mwe_id=2.mwe_id within <s/>
```

This query will match the first token in an expression, anything in between and the continuation of the MWE. To avoid greedy matching (`[]*` overshoot and match other MWEs in the sentence) we make the condition that the MWE id tag of the first token and continuation part should be the same (as stipulated by the condition `& 1.mwe_id=2.mwe_id` which uses 1 and 2 as the names previously given to the two nodes in `1:[...]` and `2:[...]`); because the values of `mwe_id` are only unique within a sentence, we also make sure both tokens belong to the same sentence through the `within <s/>` condition.

The previous query will match only two tokens in each MWE. In case more tokens needed to be highlighted, the following query has to be evaluated:

```
1:[mwe_order="first"] []* 2:[mwe_order="cont"] []*  3:[mwe_order="cont"] &
1.mwe_id=2.mwe_id & 1.mwe_id=3.mwe_id within <s/>
```

Another method for highlighting just the two tokens belonging to the same MWE but not the intermediate words is through the use of `meet` operator:

```
(meet 1:[mwe_order="first"] 2:[mwe_order="cont"] 0 5) & 1.mwe_id=2.mwe_id within <s/>
```

The `meet` operator with parameters `0 5` then formulates the condition that node 2 must be at most 0 words to the left and at most 5 words to the right of node 1.

4.3 Overlapping and Embedded MWEs

A single token may be part of multiple MWEs in two cases.

In the first case, one MWE is embedded in another one, as in the case of the Czech LVC *dát se v let* 'begin flying', which contains the inherently reflexive verb *dát se* 'enter into, begin'.

In the second case, two MWEs overlap without being embedded in each other. This happens particularly in cases of coordination mixed with ellipsis, as in this sentence fragment:

```
1 They      _  _
2 were      _  _
3 letting   _  1:VPC;2:VPC
4 us        _  _
5 in        _  1
6 and       _  _
7 out       _  2
```

Here a full linguistic analysis would first expand this fragment to *they were letting us in and they were letting us out*, in which case the two MWEs would not overlap. However, the Parseme annotation style does not attempt to restore ellided tokens and instead annotates the token that is present in the sentence as belonging to both coordinated MWEs.

The following query matches all nodes that belong to multiple MWEs simultaneously:

```
[mwe=".*;.*"]
```

In case of coordinated MWEs, we further expect that both MWEs are of the same type, which translates into

```
[mwe="(.*);\1"]
```

On the other hand, two tokens that share the same pair of MWE ids typically (although not necessarily) belong to a pair of overlapping MWEs:

```
(meet 1:[mwe_id="(.*;.*)"] 2:[] 1 5) & 1.mwe_id=2.mwe_id within <s/>
```

4.4 Queries Involving Morphosyntactic Information

Evidently, CQL queries can be formulated to simultaneously make use of annotations that are specific to MWEs and those that express other linguistic information such as morphosyntactic about their building blocks. For example, the following query:

```
1:[mwe_order="first" & upostag="VERB" & mwe="LVC"] []* 2:[mwe_order="cont" & upostag="NOUN"] &
  1.mwe_id=2.mwe_id within <s/>
```

finds all light verb constructions where the real syntactic head goes first.

Similarly, a simple query such as

```
[mwe="LVC" & upostag="VERB"]
```

followed by a request for a frequency list (through the user interface) returns the frequency list of verbs used in LVCs.

Above we listed basic queries which do not involve constraints on word forms, lemmas, or language-specific morphological tags—examples of this sort can be found at https://ufal.mff.cuni.cz/lindat-kontext/parseme-mwe.

5 Conclusion and future work

Concordance systems (in our paper KonText and NoSke, but also their ancestors such as the Sketch Engine, and the IMS Open Corpus Workbench) for exploring corpora using CQL queries are well known tools among linguists for applications such as lexicography. We believe that these systems are also effective tools for exploring MWE-annotated corpora, particularly at the absence of sufficient resources for developing specialized tools for their manipulation. To this end, we show a method to encode and query an MWE annotated corpus in a concordance system; this can facilitate the search and retrieval of MWEs in corpus based studies.

One possible area of future work is to extend the current interfaces' capability to handle search and retrieval of discontinuous structures, e.g. by extending the concept of the "key word in context" to "key *words* in context" (KWsIC), with "context" denoting not just the left and right context, but also intermediate context between the KWIC words. The `meet` operator goes some way towards this goal, but is not sufficient for more complex cases such as KWsIC consisting of three or more tokens; we propose to add a new operator of the form `(all [attribute="value"] within <structure/>)`.

Acknowledgments

The first author has been supported by the postdoctoral fellowship grant of the Hong Kong Polytechnic University, project code G-YW2P. The second author has been supported by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071); the data is shared through services developed and/or maintained within the same project.

References

Marie Candito, Fabienne Cap, Silvio Cordeiro, Vassiliki Foufi, Polona Gantar, Voula Giouli, Carlos Herrero, Mihaela Ionescu, Verginica Mititelu, Johanna Monti, Joakim Nivre, Mihaela Onofrei, Carla Parra Escartín, Manfred Sailer, Carlos Ramisch, Monica-Mihaela Rizea, Agata Savary, Ivelina Stonayova, Sara Stymne, and Veronika Vincze. 2017. Parseme shared task on automatic identification of verbal MWEs - edition 1.0. Annotation guidelines.

Stefan Evert and Andrew Hardie. 2011. Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 conference*. University of Birmingham, UK.

Natalia Klyueva and Pavel Straňák. 2016. Improving corpus search via parsing. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Paris, France, pages 2862–2866.

Scott Martens. 2013. TüNDRA: A Web Application for Treebank Search and Visualization. In *Proceedings of The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*. pages 133–144.

Behrang QasemiZadeh and Anne-Kathrin Schumann. 2016. The ACL RD-TEC 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*. http://www.lrec-conf.org/proceedings/lrec2016/summaries/681.html.

Victoria Rosén, Gyri Smørdal Losnegaard, Koenraad De Smedt, Eduard Bejček, Agata Savary, Adam Przepiórkowski, Petya Osenova, and Verginica Barbu Mititelu. 2015. A survey of multiword expressions in treebanks. In *Proceedings of the 14th International Workshop on Treebanks & Linguistic Theories conference*. Warsaw, Poland.

Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In Jan Hajič, Koenraad De Smedt, Marko Tadić, and António Branco, editors, *META-RESEARCH Workshop on Advanced Treebanking at LREC2012*. pages 22–29.

Pavel Rychlý. 2007. Manatee/bonito - a modular corpus manager. In *1st Workshop on Recent Advances in Slavonic Natural Language Processing*. Masaryk University, pages 65–70.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The PARSEME Shared Task on Automatic Identification of Verbal Multi-word Expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*. Valencia, Spain.

Agata Savary, Manfred Sailer, Yannick Parmentier, Michael Rosner, Victoria Rosén, Adam Przepiórkowski, Cvetana Krstev, Veronika Vincze, Beata Wójtowicz, Gyri Smørdal Losnegaard, Carla Parra Escartín, Jakub Waszczuk, Matthieu Constant, Petya Osenova, and Federico Sangati. 2015. PARSEME – PARSing and Multiword Expressions within a European multilingual network. In *7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2015)*. Poznań, Poland. https://hal.archives-ouvertes.fr/hal-01223349.

Jan Štěpánek and Petr Pajas. 2010. Querying diverse treebanks in a uniform way. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. http://www.lrec-conf.org/proceedings/lrec2010/summaries/381.html.