Token-based MWE Identification Strategies in the `mwetoolkit`

Silvio Ricardo Cordeiro[1,2] and **Carlos Ramisch**[1] and **Aline Villavicencio**[2]
[1] Aix Marseille Université, CNRS, LIF UMR 7279, 13288, Marseille, France
[2] Federal University of Rio Grande do Sul, Brazil
srcordeiro@inf.ufrgs.br, carlos.ramisch@lif.univ-mrs.fr, avillacicencio@inf.ufrgs.br

1 Token-based MWE identification

The accurate identification of Multiword Expression (MWE) instances in running text is a fundamental task in the pipeline of many NLP applications. For example, MT systems need to know when a group of words must be translated as a unit, and parsers need to recognise the cases where a seemingly unrelated set of words should be grouped as a single lexeme or constituent. In short, many applications would benefit from being able to identify groups of words that behave differently from what would be commonly expected.

MWE identification can be seen as a tagging process that takes as input a corpus and, optionally, an MWE lexicon, outputting an annotated corpus that explicitly indicates where each expression occurs. A toolkit such as `jMWE` (Finlayson and Kulkarni, 2011) can be used to annotate some input based on preexisting lexicons. The output is a corpus where each MWE occurrence found in a lexicon has been matched and tagged.

Tools such as `jMWE` and parsing pipelines performing MWE identification often require predetermined lists of expressions. The construction of such lists can be greatly simplified by using an MWE extractor, such as `mwetoolkit` (Ramisch, 2015). This extractor builds on the notion of regular-expression patterns based on token properties. For example, given a noun compound pattern such as $\texttt{Noun Noun}^{+}$ and a POS-tagged corpus, the extractor lists all occurrences of this expression in the text, which can in turn be passed on for MWE identification.

The pipeline described above will often work, but has some shortcomings. First, if the words in the MWE do not appear contiguously (e.g. split phrasal verbs in English), a contiguous annotator such as `jMWE` will fail to detect them. Secondly, the use of separate tools for extraction and identification will miss the opportunity of sharing information — for example, annotating the source corpus directly during the identification — and this has negative results both in terms of CPU time and in the inability to guarantee that all MWE candidates by one tool have been projected back onto the source corpus by the other tool.

2 Proposed Approach

We propose an extension to the `mwetoolkit` which annotates input corpora based on either a list of MWE candidates or a list of patterns.[1] In order to overcome the limitation of `jMWE`, our annotator has additional features described below.

1. **Different gapping possibilities**
   - Contiguous: Matches contiguous sequences of words from a list of MWEs.
   - Gappy: Matches words with up to a limit number of gaps in between.
2. **Different match distances**
   - Shortest: Matches the shortest possible candidate (e.g. for phrasal verbs, we want to find only the closest particle).
   - Longest: Matches the longest possible candidate (e.g. for noun compounds).
   - All: Matches all possible candidates (useful as a fallback when shortest and longest are too strict).
3. **Different match modes**
   - Non-overlapping: Matches at most one MWE per word in the corpus.
   - Overlapping: Allows words to be part of more than one MWE (e.g. to find MWEs inside the gap of another MWE).
4. **Source-based annotation**: MWEs are extracted with detailed source information, which can later be used for quick annotation of the original corpus.

[1] Software and documentation freely available at `http://mwetoolkit.sf.net/` under open-source licence.

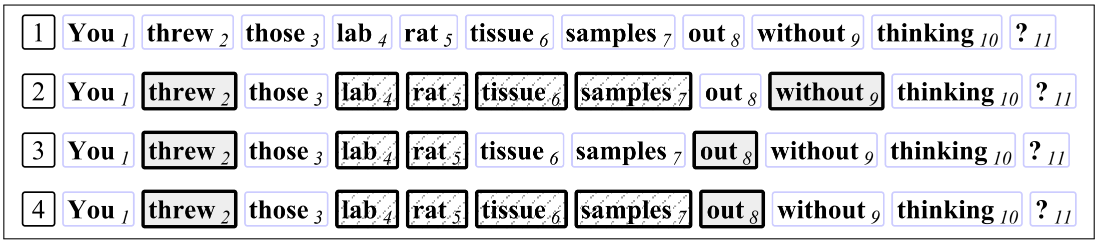
1 You 1 threw 2 those 3 lab 4 rat 5 tissue 6 samples 7 out 8 without 9 thinking 10 ? 11
2 You 1 threw 2 those 3 lab 4 rat 5 tissue 6 samples 7 out 8 without 9 thinking 10 ? 11
3 You 1 threw 2 those 3 lab 4 rat 5 tissue 6 samples 7 out 8 without 9 thinking 10 ? 11
4 You 1 threw 2 those 3 lab 4 rat 5 tissue 6 samples 7 out 8 without 9 thinking 10 ? 11

Figure 1: Gappy MWE-annotated output with different match distances.

3 Example

Consider two different MWE patterns described by the POS regular expressions below:

- `NounCompound` → `Noun Noun`$^+$
- `PhrasalVerb` → `Verb (Word`*`) Particle`

Given an input such as Sentence 1 (Figure 1) the gappy approach with different match distances will detect different types of MWEs. In Sentence 2, we show the result of identification using the *longest* match distance, which although well suited to identify noun compounds, may be too permissive for phrasal verbs combining with the closest particle (*out*). For the latter the *shortest* match distance will yield the correct response, but will be excessively strict when looking for a pattern such as the one for noun compounds, as shown in Sentence 3.

4 Discussion

The proposed modifications to the `mwetoolkit` combine the use of powerful generic patterns with a token-based identification algorithm with different matching possibilities. In the end, a wise choice of the best match distance is always necessary when looking for patterns in corpora, and these new customisation possibilities allow identification under the appropriate conditions, so that one can achieve the result shown in Sentence 4.

With the toolkit's new features, one can either annotate a corpus based on a preexisting lexicon of MWEs or perform MWE type-based extraction, generate a lexicon and subsequently use it to annotate a corpus. When annotating the same corpus from which MWE types were extracted, source-based annotation can be used for best results.

One limitation of this approach concerns the occurrence of ambiguous expressions. Since the toolkit does not perform token disambiguation, an expression such as *piece of cake* would be annotated as a MWE in both of these sentences:

1. I finished the test and it was a *piece of cake*;
2. I went to the bakery and ate a *piece of cake*.

While the second sentence above contains the components of a MWE, its meaning is not idiosyncratic, as the annotation would imply. In this case, the annotated sentences need to be disambiguated by another tool so as to determine what instances do actually correspond to an MWE (Korkontzelos et al., 2013).

5 Future work

As future work, one extension of the annotator that might yield more reliable results would be the internal use of sequence models for the disambiguation of MWE occurrences in the text, so that only the idiosyncratic cases are annotated.

We would also like to test and propose measures to evaluate token-based MWE annotation. Some possible strategies include exact and smoothed precision and recall, and weighted measures that take different MWE types into account.

The work presented here can be used directly as part of a parser, for instance, during preprocessing and tokenisation. It can also be useful as a first step in the annotation of treebanks and linguistic resources for parsing. Therefore, it constitutes a contribution towards better handling of MWEs in parsing and in NLP analysis pipelines in general.

Acknowledgments

This research was partly funded by PARSEME, CAMELEON project (CAPES-COFECUB 707-11) and AIM-WEST project (FAPERGS-INRIA 1706-2551/13-7).

References

Mark Finlayson and Nidhi Kulkarni. 2011. Detecting multiword expressions improves word sense disambiguation. In *Proc. of the ALC Workshop on MWEs (MWE 2011)*, pages 20–24, Portland, OR, USA, Jun. ACL.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 2013. Semeval-2013 task 5: Evaluating phrasal semantics. In *In Proc. of SemEval 2013*, Atlanta, Georgia, USA, Jun.

Carlos Ramisch. 2015. *Multiword Expressions Acquisition: A Generic and Open Framework*, volume XIV of *Theory and Applications of Natural Language Processing*. Springer.