Multiword Expressions in the wild?
The `mwetoolkit` comes in handy

Carlos Ramisch[†*] **Aline Villavicencio**[*] **Christian Boitet**[†]
[†] GETALP – Laboratory of Informatics of Grenoble, University of Grenoble
[*] Institute of Informatics, Federal University of Rio Grande do Sul
{ceramisch,avillavicencio}@inf.ufrgs.br Christian.Boitet@imag.fr

Abstract

The `mwetoolkit` is a tool for automatic extraction of Multiword Expressions (MWEs) from monolingual corpora. It both generates and validates MWE candidates. The generation is based on surface forms, while for the validation, a series of criteria for removing noise are provided, such as some (language independent) association measures.[1] In this paper, we present the use of the `mwetoolkit` in a standard configuration, for extracting MWEs from a corpus of general-purpose English. The functionalities of the toolkit are discussed in terms of a set of selected examples, comparing it with related work on MWE extraction.

1 MWEs in a nutshell

One of the factors that makes Natural Language Processing (NLP) a challenging area is the fact that some linguistic phenomena are not entirely compositional or predictable. For instance, why do we prefer to say *full moon* instead of *total moon* or *entire moon* if all these words can be considered synonyms to transmit the idea of completeness? This is an example of a *collocation*, i.e. a sequence of words that tend to occur together and whose interpretation generally crosses the boundaries between words (Smadja, 1993). More generally, collocations are a frequent type of *multiword expression (MWE)*, a sequence of words that presents some lexical, syntactic, semantic, pragmatic or statistical idiosyncrasies (Sag et al., 2002). The definition of MWE also includes a wide range of constructions like phrasal verbs (*go ahead*, *give up*), noun compounds (*ground speed*), fixed expressions (*a priori*) and multiword terminology (*design pattern*). Due to their heterogeneity, MWEs vary in terms of syntactic flexibility (*let alone* vs *the moon is at the full*) and semantic opaqueness (*wheel chair* vs *pass away*).

While fairly studied and analysed in general Linguistics, MWEs are a weakness in current computational approaches to language. This is understandable, since the manual creation of language resources for NLP applications is expensive and demands a considerable amount of effort. However, next-generation NLP systems need to take MWEs into account, because they correspond to a large fraction of the lexicon of a native speaker (Jackendoff, 1997). Particularly in the context of domain adaptation, where we would like to minimise the effort of porting a given system to a new domain, MWEs are likely to play a capital role. Indeed, theoretical estimations show that specialised lexica may contain between 50% and 70% of multiword entries (Sag et al., 2002). Empirical evidence confirms these estimations: as an example, we found that 56.7% of the terms annotated in the Genia corpus are composed by two or more words, and this is an underestimation since it does not include general-purpose MWEs such as phrasal verbs and fixed expressions.

The goal of `mwetoolkit` is to aid lexicographers and terminographers in the task of creating language resources that include multiword entries. Therefore, we assume that, whenever a textual corpus of the target language/domain is available, it is possible to automatically extract interesting sequences of words that can be regarded as candidate MWEs.

2 Inside the black box

MWE identification is composed of two phases: first, we automatically generate a list of candi-

[1] The first version of the toolkit was presented in (Ramisch et al., 2010b), where we described a language- and type-independent methodology.

$$
\begin{aligned}
\texttt{mle} &= \frac{c(w_1 \dots w_n)}{N} \\
\texttt{dice} &= \frac{n \times c(w_1 \dots w_n)}{\sum_{i=1}^{n} c(w_i)} \\
\texttt{pmi} &= \log_2 \frac{c(w_1 \dots w_n)}{E(w_1 \dots w_n)} \\
\texttt{t-score} &= \frac{c(w_1 \dots w_n) - E(w_1 \dots w_n)}{\sqrt{c(w_1 \dots w_n)}}
\end{aligned}
$$

Figure 1: A candidate is a sequence of words w_1 to w_n, with word counts $c(w_1) \dots c(w_n)$ and n-gram count $c(w_1 \dots w_n)$ in a corpus with N words. The expected count if words co-occurred by chance is $E(w_1 \dots w_n) \approx \frac{c(w_1) \dots c(w_n)}{N^{n-1}}$.

candidate	f_{EP}	f_{google}	**class**
status quo	137	1940K	True
US navy	4	1320K	False
International Cooperation	2	1150K	False
Cooperation Agreement	188	115K	True
Panama Canal	2	753K	True
security institution	5	8190	False
lending institution	4	54800	True
human right	2	251K	True
Human Rights	3067	3400K	False
pro-human right	2	34	False

Table 1: Example of MWE candidates extracted by `mwetoolkit`.

dates from the corpus; then we filter them, so that we can discard as much noise as possible. *Candidate generation* uses flat linguistic information such as surface forms, lemmas and parts of speech (POS).[2] We can then define target sequences of POS, such as `VERB NOUN` sequences, or even more fine-grained constraints which use lemmas, like *take* `NOUN` and *give* `NOUN`, or POS patterns that include wildcards that stand for any word or POS.[3] The optimal POS patterns for a given domain, language and MWE type can be defined based on the analysis of the data.

For the *candidate filtering* a set of association measures (AMs), listed in figure 1, are calculated for each candidate. A simple threshold can subsequently be applied to filter out all the candidates for which the AMs fall below a user-defined value. If a gold standard is available, the toolkit can build a classifier, automatically annotating each candidate to indicate whether it is contained in the gold standard (i.e. it is regarded as a true MWE) or not (i.e. it is regarded as a non-MWE).[4] This annotation is not used to filter the lists, but only by the classifier to learn the relation between the AMs and the MWE class of the candidate. This is particularly useful because, to date, it remains unclear which AM performs better for a particular type or language, and the classifier applies measures according to their efficacy in filtering the candidates.Some examples of output are presented in table 1.

3 Getting started

The toolkit is open source software that can be freely downloaded (`sf.net/projects/mwetoolkit`). As a demonstration, we present the extraction of noun-noun compounds from the general-purpose English Europarl (EP) corpus[5].

To preprocess the corpus, we used the sentence splitter and tokeniser provided with EP, followed by a lowercasing treatment (integrated in the toolkit), and lemmatisation and POS tagging using the TreeTagger[6]. The tagset was simplified since some distinctions among plural/singular and proper nouns were irrelevant.

From the preprocessed corpus, we obtained all sequences of 2 nouns, which resulted in 176,552 unique noun compound candidates. Then, we obtained the corpus counts for the bigrams and their component unigrams in the EP corpus. Adopting the web as a corpus, we also use the number of pages retrieved by Google and by Yahoo! as

[2] If tools like a POS tagger are not available for a language/domain, it is possible to generate simple n-gram lists ($n = 1..10$), but the quality will be inferior. A possible solution is to filter out candidates on a keyword basis, e.g. from a list of stopwords).

[3] Although syntactic information can provide better results for some types of MWEs, like collocations (Seretan, 2008), currently no syntactic information is allowed as a criterion for candidate generation, keeping the toolkit as simple and language independent as possible.

[4] The gold standard can be a dictionary or a manually annotated list of candidates.

[5] `www.statmt.org/europarl`.

[6] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`.

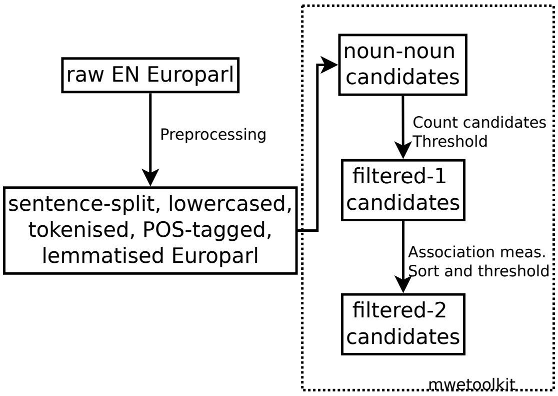
raw EN Europarl
Preprocessing
sentence-split, lowercased, tokenised, POS-tagged, lemmatised Europarl
noun-noun candidates
Count candidates Threshold
filtered-1 candidates
Association meas. Sort and threshold
filtered-2 candidates
mwetoolkit

Figure 2: Step-by-step demonstration on the EP corpus.

counts. The `mwetoolkit` implements a cache mechanism to avoid redundant queries, but to speed up the process[7], we filtered out all candidates occurring less than two times in EP, which reduced the list of candidates to 64,551 entries (*filtered-1 candidates* in figure 2).

For the second filtering step, we calculated four AMs for each of the three frequency sources (EP, Google and Yahoo!). Some results on machine learning applied to the candidate lists of the `mwetoolkit` can be found in Ramisch et al. (2010b). Here, we will limit ourselves to a discussion on some advantages and inconvenients of the chosen approach by analysing a list of selected examples.

4 Pros and cons

One of the biggest advantages of our approach is that, since it is language independent, it is straightforward to apply it on corpora in virtually any language. Moreover, it is not dependent on a specific type of construction or syntactic formalism. Of course, since it only uses limited linguistic information, the accuracy of the resulting lists can always be further improved with language-dependent tools. In sum, the toolkit allows users to perform systematic MWE extraction with consistent intermediary files and well defined scripts and arguments (avoiding the need for a series of ad hoc separate scripts). Even if some basic knowledge about how to run Python scripts and how to pass arguments to the command line is necessary, the user is not required to be a programmer.

Nested MWEs are a problem in the current approach. Table 1 shows two bigrams *International Cooperation* and *Cooperation Agreement*, both evaluated as False candidates. However, they could be considered as parts of a larger MWE *International Cooperation Agreement*, but with the current methodology it is not possible to detect this kind of situation. Another case where the candidate contains a MWE is the example *pro-human right*, and in this case it would be necessary to separate the prefix from the MWE, i.e. to re-tokenise the words around the MWE candidate. Indeed, tools for consistent tokenisation, specially concerning dashes and slashes, could improve the quality of the results, in particular for specialised corpora.

The toolkit provides full integration with web search engine APIs. The latter, however, are of limited utility because search engines are not only slow but also return more or less arbitrary numbers, some times even inconsistent (Ramisch et al., 2010c). When large corpora like EP are available, we suggest that it is better to use its counts rather than web counts. The toolkit provides an efficient indexing mechanism, allowing for arbitrary *n*-grams to be counted in linear time.

The automatic evaluation of the candidates will always be limited by the coverage of the reference list. In the examples, *Panama Canal* is considered as a true MWE whereas *US navy* is not, but both are proper names and the latter should also be included as a true candidate. The same happens for the candidates *Human Rights* and *human right*. The `mwetoolkit` is an early prototype whose simple design allows fine tuning of knowledge-poor methods for MWE extraction. However, we believe that there is room for improvement at several points of the extraction methodology.

5 From now on

One of our goals for future versions is to be able to extract bilingual MWEs from parallel or comparable corpora automatically. This could be done through the inclusion of automatic word alignment information. Some previous experiments show, however, that this may not be enough, as

[7] Yahoo! limits the queries to 5,000/day.

automatic word alignment uses almost no linguistic information and its output is often quite noisy (Ramisch et al., 2010a). Combining alignment and shallow linguistic information seems a promising solution for the automatic extraction of bilingual MWEs. The potential uses of these lexica are multiple, but the most obvious application is machine translation. On the one hand, MWEs could be used to guide the word alignment process. For instance, this could solve the problem of aligning a language where compounds are separate words, like French, with a language that joins compound words together, like German. In statistical machine translation systems, MWEs could help to filter phrase tables or to boost the scores of phrases which words are likely to be multiwords.Some types of MWE (e.g. collocations) could help in the semantic disambiguation of words in the source language. The sense of a word defined by its collocate can allow to chose the correct target word or expression (Seretan, 2008).

We would also like to improve the techniques implemented for candidate filtering. Related work showed that association measures based on contingency tables are more robust to data sparseness (Evert and Krenn, 2005). However, they are pairwise comparisons and their application on arbitrarily long *n*-grams is not straightforward. An heuristics to adapt these measures is to apply them recursively over increasing *n*-gram length. Other features that could provide better classification are context words, linguistic information coming from simple word lexica, syntax, semantic classes and domain-specific keywords. While for poor-resourced languages we can only count on shallow linguistic information, it is unreasonable to ignore available information for other languages. In general, machine learning performs better when more information is available (Pecina, 2008).

We would like to evaluate our toolkit on several data sets, varying the languages, domains and target MWE types. This would allow us to assign its quantitative performance and to compare it to other tools performing similar tasks. Additionally, we could evaluate how well the classifiers perform across languages and domains. In short, we believe that the `mwetoolkit` is an important first step toward robust and reliable MWE treatment. It is a freely available core application providing flexible tools and coherent up-to-date documentation, and these are essential characteristics for the extension and support of any computer system.

Acknowledgements

This research was partly supported by CNPq (Projects 479824/2009-6 and 309569/2009-5), FINEP and SEBRAE (COMUNICA project FINEP/SEBRAE 1194/07).

References

Evert, Stefan and Brigitte Krenn. 2005. Using small random samples for the manual evaluation of statistical association measures. *Comp. Speech & Lang. Special issue on MWEs*, 19(4):450–466.

Jackendoff, Ray. 1997. Twistin' the night away. *Language*, 73:534–559.

Pecina, Pavel. 2008. Reference data for czech collocation extraction. In *Proc. of the LREC Workshop Towards a Shared Task for MWEs (MWE 2008)*, pages 11–14, Marrakech, Morocco, Jun.

Ramisch, Carlos, Helena de Medeiros Caseli, Aline Villavicencio, André Machado, and Maria José Finatto. 2010a. A hybrid approach for multiword expression identification. In *Proc. of the 9th PROPOR (PROPOR 2010)*, volume 6001 of *LNCS (LNAI)*, pages 65–74, Porto Alegre, RS, Brazil. Springer.

Ramisch, Carlos, Aline Villavicencio, and Christian Boitet. 2010b. mwetoolkit: a framework for multiword expression identification. In *Proc. of the Seventh LREC (LREC 2010)*, Malta, May. ELRA.

Ramisch, Carlos, Aline Villavicencio, and Christian Boitet. 2010c. Web-based and combined language models: a case study on noun compound identification. In *Proc. of the 23th COLING (COLING 2010)*, Beijing, China, Aug.

Sag, Ivan, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proc. of the 3rd CICLing (CICLing-2002)*, volume 2276/2010 of *LNCS*, pages 1–15, Mexico City, Mexico, Feb. Springer.

Seretan, Violeta. 2008. *Collocation extraction based on syntactic parsing*. Ph.D. thesis, University of Geneva, Geneva, Switzerland.

Smadja, Frank A. 1993. Retrieving collocations from text: Xtract. *Comp. Ling.*, 19(1):143–177.