mwetoolkit: a Framework for Multiword Expression Identification

Carlos Ramisch[†*], Aline Villavicencio[*], Christian Boitet[†]

[†] GETALP – Laboratory of Informatics of Grenoble, University of Grenoble (France)
[*] Institute of Informatics, Federal University of Rio Grande do Sul (Brazil)
Carlos.Ramisch@imag.fr, avillavicencio@inf.ufrgs.br, Christian.Boitet@imag.fr

Abstract

This paper presents the Multiword Expression Toolkit (mwetoolkit), an environment for type and language-independent MWE identification from corpora. The mwetoolkit provides a targeted list of MWE candidates, extracted and filtered according to a number of user-defined criteria and a set of standard statistical association measures. For generating corpus counts, the toolkit provides both a corpus indexation facility and a tool for integration with web search engines, while for evaluation, it provides validation and annotation facilities. The mwetoolkit also allows easy integration with a machine learning tool for the creation and application of supervised MWE extraction models if annotated data is available. In our experiments, the mwetoolkit was tested and evaluated in the context of MWE extraction in the biomedical domain. Our preliminary results show that the toolkit performs better than other approaches, especially concerning recall. Moreover, this first version can be extended in several ways in order to improve the quality of the results.

1 Introduction

A Multiword Expression (MWE) can be defined as a sequence of words that presents some characteristic behaviour (at lexical, syntactic, semantic, pragmatic or statistical level) and whose interpretation crosses the boundaries between words (Sag et al., 2002). This rough definition includes a very wide range of constructions such as compound nouns (*credit card*, *mountain bike*), phrasal verbs (*carry on*, *go by [a name]*), compound terms (*benign tumor*, *nuclear fusion*), etc. MWEs are an *important* and *frequent* phenomenon in human language, that must be handled adequately in Natural Language Processing (NLP) applications with some degree of semantic interpretation.

MWEs are an *important* aspect of languages, conferring naturalness and fluency to a discourse. While native speakers rarely realize how often they employ MWEs, they are challenging for non-native speakers, since they are not only arbitrary to some extent but also too numerous and flexible to learn/memorise. From a computational perspective, it is crucial to be able to deal with MWEs since they are not only a theoretical limitation of current formalisms but also, pragmatically speaking, common sources of errors like incorrect sentence parses and awkward literal translations.

Their *frequency* can be gauged by Jackendoff's estimate that, in the general lexicon of a native speaker, multiwords correspond to more than half of the entries (Jackendoff, 1997). As pointed out by Sag et al. (2002), this could be an underestimate since the proportion of MWEs in the lexicon of broad-coverage NLP systems tends to increase as they integrate domain-specific Language Resources (LRs). Indeed, the proportion of MWEs in natural language is larger in specialised domains or sub-languages; for Krieger and Finatto (2004), more than 70% of the entries in a terminology are composed of two or more words. In the latter case, we will call *Multiword Term (MWT)* an expression describing a concept of the domain and whose meaning cannot be directly inferred by a non-expert from its parts[1].

However, there is often a disparity between the omnipresence of MWEs in natural language and the proportionally small number of multiword entries in LRs. For instance, while the Cambridge Dictionary of American English contains over 60,000 definitions, the Cambridge Phrasal Verbs Dictionary contains only 10% of that size, with 6,000 entries. Moreover, these phrasal verbs are composed by a very small subset of the verbs found in a dictionary: only 20% of the verbs listed in the Alvey Natural Language Tools lexicon form a verb-particle construction (Villavicencio, 2005). Particularly in the context of domain-specific LRs, this is understandable, since the creation of specialised lexica is an expensive task *per se*, that requires a great amount of manual effort from terminographers, lexicographers and domain experts, whether for simplex or multiword entries. On the other hand, the coverage of specialised LRs influences the performance of NLP systems, and contribute to their portability, helping the adaptation of systems that work well on general-purpose language to domain-specific texts. As a consequence, there is a need for the development of techniques and tools for the (semi) automatic identification of MWTs for inclusion in specialised LRs. More generally, obtaining wide-coverage LRs for general MWEs is a current bottleneck and an important challenge in the development of real-world NLP systems. Particularly for languages that are less resource rich than English, there is a need for viable type, domain and language independent alternatives for MWE identification.

In this context, our paper describes the development of the Multiword Expression Toolkit (mwetoolkit), an environment for MWE identification in corpora (Ramisch, 2009). The mwetoolkit employs a standard methodology, which consists of a phase of candidate *extraction* followed by a phase of candidate *filtering*, where we combine Association Measures (AMs) and descriptive features using a machine learning model to remove noise. The system extracts candidates based either on flat n-grams or specific morphosyntactic patterns (of surface forms, lemmas, POS tags). Once the candidate lists are extracted, it is possible to filter them defining criteria that range from simple count-

[1] Although this definition includes a large number of highly flexible constructions, we focus on MWTs that present limited syntactic flexibility.

based thresholds, to more complex cases such as their AMs. Since AMs are based on corpus word and n-gram counts, the toolkit provides both a corpus indexing facility and a tool for integration with web search engines (for using the web as a corpus). Additionally, for the evaluation phase, the `mwetoolkit` provides validation and annotation facilities. Finally, it also allows easy integration with a machine learning tool for the creation of supervised MWE extraction models if annotated data is available.

Originally conceived to extract multiword terminology from specialised corpora, the `mwetoolkit` can also perform automatic identification of other types of MWEs. It implements a hybrid knowledge-poor technique that only uses shallow linguistic information, thus it can be virtually applied to any corpus independently of the domain and of the language[2]. In short, the main goal of the `mwetoolkit` is to provide lexicographers and terminographers with targeted lists of candidate MWEs and MWTs, thus speeding up the creation of general-purpose and specialised dictionaries.

The remainder of this paper is organised as follows: firstly, we present a brief review of existing techniques for the extraction of multiword and of specialised lexical information from corpora (§ 2). Secondly, we discuss the architecture and implementation of `mwetoolkit` (§ 3), and this is followed by a description of an experiment performed in the biomedical domain (§ 4). This is done in two steps: first, we describe the detailed MWT identification process with an example (§ 5), and then we present the results of a comparative performance evaluation (§ 6). We conclude on a discussion about our perspectives for future extensions of the `mwetoolkit` (§ 7).

2 Related Work

Recently, a large number of techniques and tools has been proposed to that aid in the creation and extension of lexical resources (Carroll and Briscoe, 2002; Preiss et al., 2007; Messiant et al., 2008). Existing lexical acquisition techniques, however, are often developed to deal with simplex words, and may not be easily transferable or even suitable to deal with MWEs due to their characteristics. Calzolari et al. (2002) define an MWE as *a sequence of words that acts as a single unit at some level of linguistic analysis*, with some of the following features:

1. reduced syntactic and/or semantic transparency;
2. reduced compositionality;
3. more or less frozen status;
4. violation of general syntactic rules;
5. high degree of lexicalisation;
6. high degree of conventionality.

Given these characteristics, and the heterogeneous nature of the different types of MWE, the treatment of multiword phenomena concerning not only specialised but also general-purpose lexicon constitute a big challenge and, in most cases, an obvious weakness in current NLP technology (Sag et al., 2002; Copestake et al., 2002; Calzolari et al., 2002).

Among early work on developing methods for MWE identification, there is that of Smadja (1993), who proposed Xtract for general-purpose collocation extraction from text, using a combination of n-grams and a mutual information measure. On general-purpose texts, Xtract has a precision of around 80% for identifying collocational units. Since then, many advances have been made, either looking at MWEs in general (Zhang et al., 2006; Villavicencio et al., 2007), or focusing on specific MWE types, such as collocations (Pearce, 2002), phrasal verbs (Baldwin, 2005; Ramisch et al., 2008), compound nouns (Keller et al., 2002), etc. A popular type-independent alternative to MWE identification is to use statistical AMs (Evert and Krenn, 2005; Zhang et al., 2006; Villavicencio et al., 2007), which have been applied to the task with varying degrees of success. One of the advantages of this approach is that it is also language independent. This is particularly important since although work on MWEs in several languages has been reported, e.g. Dias (2003) for Portuguese and Evert and Krenn (2005) for German, work on English still seems to predominate (Pearce, 2002; Baldwin, 2005; Ramisch et al., 2008).

When it comes to the creation of specialised LRs, there has been some early work to automate terminographic extraction. Justeson and Katz (1995), for instance, used a small set of selected POS tag patterns to extract MWT candidates and then filter them using their raw frequencies in the corpus. Although very simple, this method yields accurate results when high recall is not required. Frantzi et al. (2000) presented a mixed approach in which candidates are extracted using shallow POS patterns and then a statistical test, the *C*-value, is used to guarantee that the extracted pattern is actually a MWT. However, most of the current methods that perform some kind of terminological extraction are based on symbolic and domain-dependent rules, and these often depend on proprietary resources that are rarely made available. Additionally, much of the existing approaches for terminological acquisition are not language independent. Therefore, even though sophisticated systems do exist for particular languages and specific contexts, e.g. Hagège et al. (2002) for English biomedicine, these are difficult to adapt to other languages and domains.

3 System Architecture

The `mwetoolkit` was designed as a set of Python scripts that handle intermediary XML representing the *corpus*, the list of MWE *patterns*, the list of MWE *candidates* and the *reference* dictionary. Each script performs a specific task in the pipeline of MWE extraction, from the raw corpus to the filtered list of MWE candidates including their automatic evaluation if a reference gold standard is given. Figure 1 summarises the architecture of `mwetoolkit`.

Preprocessing using external tools should include (a) consistent tokenisation, (b) lemmatisation and (c) part-of-speech tagging. Steps (b) and (c) are optional, but lemma and POS information can be crucial for determining the

[2]Provided that a good word segmentation tool is available in the case of scripts with no word delimiter.

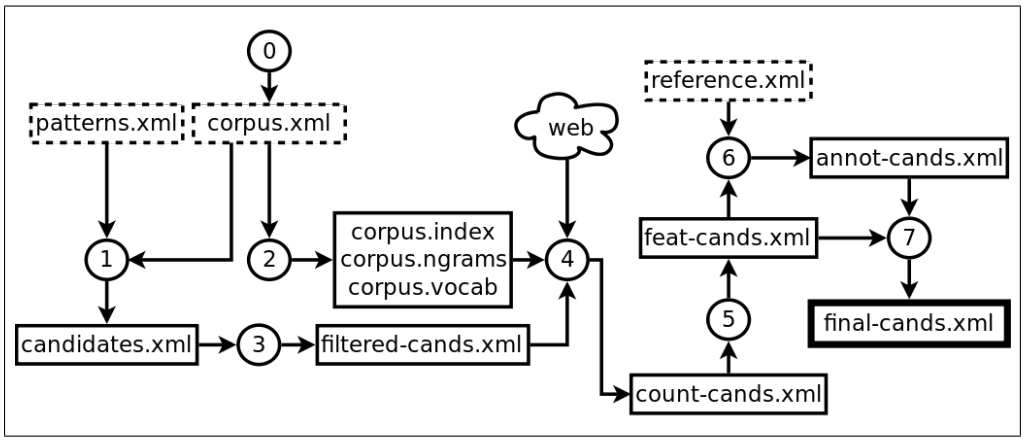

Figure 1: Architectural scheme of `mwetoolkit`: (0) preprocess the corpus, (1) extract the candidates that match the patterns, (2) index the corpora using suffix arrays, (3) filter the candidates list, (4) count n-grams and words in the corpora, (5) calculate AMs and descriptive features, (6) automatically annotate (part of) the candidates and (7) train/apply a machine learning model. Inputs are boxed with dashed lines, output with a thick line.

quality of the extracted MWEs. Case homogenisation can be performed through `mwetoolkit`'s heuristic lowercasing rules that tend to preserve the case of words that occur with different capitalisation throughout the corpus. This might be important because, in some domains, simple lowercasing is not enough, e.g. in the case of chemical components and industry acronyms.

Once the corpus has been preprocessed, `mwetoolkit` generates a first list of candidates based either on raw n-grams or on POS patterns. The former is a straightforward method to extract all possible word combinations ranging from unigrams to 10-grams and could be used as a backoff strategy when no linguistic information is available. The latter allows the definition of fine-grained morphosyntactic constraints on the candidates, e.g. the extraction of *Noun-Noun* and *Adjective-Noun* pairs or of collocations involving the adjective *strong*. Although deeper syntactic constraints (e.g. constituents or dependency relations) are not allowed, it is possible to define patterns containing wildcards, to extract semi-fixed expressions with intervening words. The initial candidate list can be filtered *a posteriori* in order to exclude candidates that contain spurious punctuation, n-grams occurring less frequently than a given threshold or specific words and POS.

For each candidate, a set of features is generated in order to allow the application of machine learning models. Two kinds of features are included in the `mwetoolkit` package: descriptive features and statistical Association Measures (AMs). The latter measure the degree of independence between the number of occurrences of the MWT candidate and the number of occurrences of the individual words that compose it. AMs are calculated as follows:

1. A corpus containing N word tokens is indexed using a suffix array, a memory-efficient data structure that allows for n-grams of arbitrary size to be searched efficiently in very large corpora.

2. For each candidate sequence of n contiguous words w_1 through w_n, `mwetoolkit` gets the individual word counts $c(w_1) \dots c(w_n)$ and the overall n-gram count $c(w_1 \dots w_n)$ from the index.

3. We calculate the expected n-gram count E if words cooccurred by chance, i.e. if we suppose that word occurrences are independent events, an n-gram would occur $E(w_1 \dots w_n) \approx \frac{c(w_1) \dots c(w_n)}{N^{n-1}}$ times[3].

4. That information is used to calculate four statistical AMs for each MWE candidate in each corpus, namely:

   - the maximum likelihood estimator:
   $$\texttt{mle} = \frac{c(w_1 \dots w_n)}{N};$$
   - Dice's coefficient:
   $$\texttt{dice} = \frac{n \times c(w_1 \dots w_n)}{\sum_{i=1}^{n} c(w_i)};$$
   - the pointwise mutual information:
   $$\texttt{pmi} = \log_2 \frac{c(w_1 \dots w_n)}{E(w_1 \dots w_n)};$$
   - and Student's t-score:
   $$\texttt{t-score} = \frac{c(w_1 \dots w_n) - E(w_1 \dots w_n)}{\sqrt{c(w_1 \dots w_n)}}.$$

We are able to calculate these measures for arbitrary-size n-grams because none of them uses contingency tables. Since candidate extraction and counting are two separate steps, an arbitrary number of corpus frequencies can be calculated. One could, for example, complement a small specialised corpus with frequencies coming from a large general-purpose corpus. Additionally, `mwetoolkit` provides full integration with Yahoo!'s API[4] and with Google's

[3] Actually, a corpus with N word tokens contains $N - n + 1$ n-grams, and therefore $E = (N - n + 1) \times \prod_{i=1}^{n} \frac{c(w_i)}{N} = \frac{(N-n+1)}{N^n} \times \prod_{i=1}^{n} c(w_i)$. However, since $n \ll N$, we can ignore the term $n + 1$ and consider that $\frac{(N-n+1)}{N^n} \approx \frac{1}{N^{n-1}}$

[4] `http://developer.yahoo.com/download/download.html`

API[5]. Both search engines provide page hit counts that allow us to see the web as a huge corpus, thus offering an alternative solution to overcome data sparseness. Since web queries can be quite time-consuming, we keep a cache file with recent queries, and this avoids some delay caused by redundant network requests.

Once each candidate has a set of associated features, we can either apply an existing machine learning model to distinguish true and false positives or we can generate a new model by assigning a class to (part of) the new candidate set. Therefore, an evaluation facility is provided so that, if a (potentially limited) reference gold standard is present, the class of the candidate is automatically inferred, i.e. if the candidate is contained in the reference list, it is a true MWT, otherwise nothing can be said about it.

The `mwetoolkit` package provides a conversion facility that allows the importation of a candidates list into the machine learning package WEKA[6]. In future versions, we would like to provide full integration between the candidate extraction step and the filter learning and application step.

4 The Experimental Setup

In the following experiments, we used the `mwetoolkit` to extract MWTs from the Genia corpus. It is composed of a set of 2,000 abstracts of scientific articles from the biomedical domain (Ohta et al., 2002) and contains around 18K sentences and around 490K tokens. The corpus contains information about sentence and word boundaries, POS tags and terminological annotation with respect to the Genia ontology. In order to train machine learning models and test them, the original corpus was divided into a training set and a test set, with the latter containing 895 sentences ($\approx 5\%$ of the corpus), and the former containing all the other sentences.

In order to unify the orthography of the words throughout the corpus, we preprocessed it uniformly according to the following criteria:

- Capitalised words were lowercased using the heuristics described in section 3.
- POS tags were simplified to match a set of patterns.
- Words containing dashes and slashes were retokenised, as these symbols are not used consistently in the Genia corpus (e.g. *T cell* and *T-cell*). Therefore, any word that contained these symbols was split into independent subparts as the symbols were removed (e.g. *T-cell* becomes *T cell*) [7].
- Acronyms were recognised and removed when they occurred between parentheses, e.g. *human immunodeficiency virus (HIV) type 1* was changed to *human immunodeficiency virus type 1*.
- Nouns were lemmatised to their singular form.

These preprocessing steps aim to reduce the problem of data sparseness, which is particularly acute for MWEs and specific domains, and they have a significant impact on the quality of the results. We estimate, for instance, that precision and recall are reduced by more than 50% if the lemmatisation and retokenisation steps are not performed. In order to keep the `mwetoolkit` as language and domain independent as possible, these preprocessing steps are currently not included as part of the `mwetoolkit`, since they may differ according to language and/or domain. Therefore any preprocessing steps have to be performed to the corpus prior to using it as basis for MWE candidate extraction. Depending on the number of steps defined and how complex they are, the preprocessing of the corpus may be the most time-consuming stage in MWE extraction.

As described in section 3, for obtaining the initial list of MWE candidates, two approaches were implemented in the `mwetoolkit`: the first one extracts all the n-grams in the corpus, while the second uses a predefined list of morphosyntactic patterns. In this paper we adopted the second strategy, and defined a list of 57 POS patterns based on those of Justeson and Katz (1995). Their original set of patterns was augmented through the use of a heuristic that enables the extraction of longer sequences of contiguous nouns and adjectives than originally defined. For instance, it is now possible to extract candidates that match POS patterns containing sequences of two to seven adjacent nouns and adjectives (e.g. *T cell*, *thromboxane receptor gene*), foreign words (e.g. *in vitro*) and numbers (e.g. *nucleotide 46*).

5 Toy Experiment

In this section, we present a step-by-step example of MWT extraction using the `mwetoolkit`, focusing on two candidates obtained using one of the POS patterns described in section 4. From the Genia corpus sentence shown in figure 2, we selected two candidates that match the sequence Adjective–Noun–Noun (A–N–N): *human CD4+ T* and *Chinese hamster ovary*. The former, although part of a longer MWT in this sentence (*human CD4+ T cell*), as a trigram it is a false positive.[8]

This initial list of candidates can be further validated using some criteria, in order to, insofar as possible, remove false positives from the list, and only keep genuine MWTs. In this paper this validation is done using a set of AMs implemented in the `mwetoolkit` (described in section 3) as basis for building a classifier. In order to calculate the AMs for each candidate, the `mwetoolkit` determines the corpus counts for the candidate as well as for the individual words that compose it. In figure 2, the n-gram and word counts of only the Genia corpus are represented, but the toolkit also allows the use of several corpora (including the web as a corpus) to calculate the AMs for each candidate in each corpus separately.

[5] `http://code.google.com/apis/ajaxsearch/`

[6] `http://www.cs.waikato.ac.nz/ml/weka/`

[7] In the future, we would like to apply existing techniques to unify the orthography of words around dashes and slashes.

[8] Currently we do not handle nested MWTs, and as a consequence each subpart of an MWT is treated independently from any other subpart. In this case, as the original MWT (*human CD4+ T cell*) matches different POS patterns, it forms 3 different candidates which are treated independently: *human CD4+ T* (A–N–N), *CD4+ T cell* (N–N–N) and *human CD4+ T cell* (A–N–N–N).

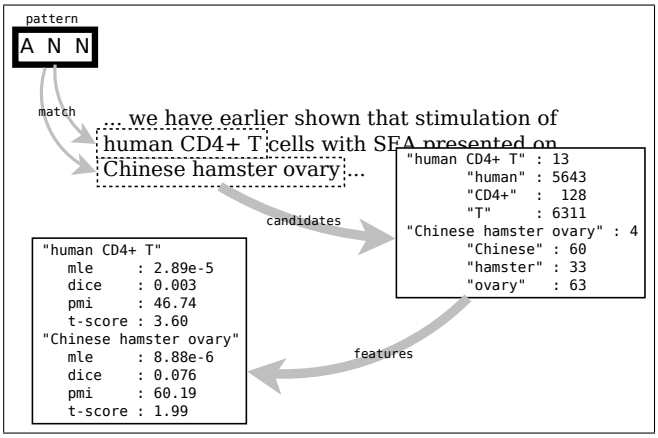

Figure 2: Example of MWT candidates extracted from the Genia corpus.

After obtaining the corpus counts, the toolkit uses this information as input to the formulae that calculate the four association scores for each candidate in each corpus. As the effectiveness of each AM seems to depend on factors like the size of the corpus, the type of texts it consists of, the language in question and others (Evert and Krenn, 2005; Villavicencio et al., 2007) and there has not been an agreement on which one should be used in each case, the toolkit calculates all of them so that the most appropriate AMs to a given case can be selected.

In this experiment, all AMs are used as features for the classifier and it then decides on the best feature combination to decide whether a candidate should be kept in the list or be discarded as noise. Additional features like morphological information (capitalisation, prefixes and suffixes, etc.), dictionary entries (if available), multilingual features, etc. may help the classification process even further and possibly be more discriminative than AMs, but since they would make the classifiers language- and/or domain-dependent, they are not currently part of the `mwetoolkit`.

Figure 3 shows an example of XML representation obtained for one of our example candidates extracted from the Genia corpus: *Chinese hamster ovary*. For each individual word and for the whole candidate, the `freq` elements show their corpus counts in two different corpora: Genia and Yahoo!. The idea is to use two heterogeneous data sources so that we do not loose in accuracy because of the sparseness of the former or because of the rough approximations done by the latter. The first two features are simple properties of the candidate such as the number of words and the POS sequence and the remainder of the features correspond to the AMs in the Genia corpus and in Yahoo!. After the list of features, the special element `tpclass` indicates the class of the candidate with respect to the reference list. This information, when available, can be used to build a new classifier for a given language or domain. In our experiment, its utility is two-fold: on the training corpus, it is used as class information for a supervised learning algorithm that will build our MWT classifier; in the test corpus, it determines whether a candidate is correctly classified as a true positive (or as a true negative), helping us evaluate the performance of the `mwetoolkit`.

```
<cand candid="4582">
    <ngram>
        <w lemma="Chinese" pos="A" >
            <freq name="genia" value="60" />
            <freq name="yahoo" value="1460000000" />
        </w>
        <w lemma="hamster" pos="N" >
            <freq name="genia" value="33" />
            <freq name="yahoo" value="42600000" />
        </w>
        <w lemma="ovary" pos="N" >
            <freq name="genia" value="63" />
            <freq name="yahoo" value="12300000" />
        </w>
        <freq name="genia" value="4" />
        <freq name="yahoo" value="723000" />
    </ngram>
    <occurs>
        <ngram>
            <w surface="Chinese" pos="A" />
            <w surface="hamster" pos="N" />
            <w surface="ovary" pos="N" />
            <freq name="corpus" value="4" />
        </ngram>
    </occurs>
    <features>
        <feat name="pos_pattern" value="A#S#N#S#N" />
        <feat name="n" value="3" />
        <feat name="mle_genia" value="8.8833220071e-06" />
        <feat name="pmi_genia" value="60.193312488" />
        <feat name="t_genia" value="1.99999969239" />
        <feat name="dice_genia" value="0.0769230769231" />
        <feat name="mle_yahoo" value="1.31454545455e-05" />
        <feat name="pmi_yahoo" value="82.8386600941" />
        <feat name="t_yahoo" value="849.996644814" />
        <feat name="dice_yahoo" value="0.00143177767509" />
    </features>
    <tpclass name="genia-reference" value="True" />
</cand>
```

Figure 3: XML fragment describing a MWT candidate extracted from the Genia corpus with `mwetoolkit`.

6 Evaluation

In this experiment we evaluate the performance of the MWT identification in terms of precision, recall and F-measure, using the Genia ontology as MWT gold standard. The Genia ontology is a manually-built resource that contains, among other information, the set of terms found in the Genia corpus (Kim et al., 2006). For a given portion of the Genia corpus, the MWT *reference list* is composed of the multiword entries of the Genia ontology that occur in that portion of the corpus.

A MWT candidate extracted automatically from a portion of the Genia corpus is considered as a *True Positive* (TP) if it is contained in the gold standard for that portion. In this way, for a list of candidate MWTs, *recall* is defined as the proportion of TPs with respect to the number of entries in the reference list and *precision* as the proportion of TPs with respect to the number of extracted candidates in that portion of the corpus. We base our evaluation of the performance of a MWT identification method through F-measure, i.e. the harmonic mean of precision and recall. We assume that the Genia ontology contains most of (or all) the MWTs present in the Genia corpus. This hypothesis allows us to perform fully automatic evaluation and to rapidly assess the effectiveness of improvements implemented in `mwetoolkit`.

The candidates, extracted from the training portion of the Genia corpus through the procedure described in sections 4 and 5, and automatically annotated with MWT information, were fed into a learning algorithm that produced a Support Vector Machine (SVM) classifier. In some experi-

	Xtract	Yahoo! terms	mwetoolkit		
			$t = 0$	$t = 1$	$t = 5$
# cand	1,558	5,404	763	739	174
# ref	27,096	27,096	2,009	2,009	2,009
# TP	1,041	1,616	401	420	129
P	66.81%	29.90%	52.56%	56.83%	74.14%
R	3.84%	5.96%	19.96%	20.91%	6.42%
F	7.26%	9.94%	28.93%	30.57%	11.82%

Table 1: Performance of Xtract and of Yahoo! terms on the whole Genia corpus. Performance of the mwetoolkit considering (a) no filtering threshold, (b) a threshold of $t = 1$ occurrence and (c) a threshold of $t = 5$ occurrences.

ments performed, among all tested machine learning models, SVM with polynomial kernel presented the best balance between precision and recall (Ramisch, 2009). We applied this model to the test corpus (the remaining unannotated 895 sentences of the Genia corpus) and evaluated the output in terms of precision and recall. We compared our system to the output of two other systems: Xtract and Yahoo! terms. Xtract was designed to identify collocations in general-purpose texts (Smadja, 1993) and has a free implementation in the Dragon toolkit[9]. Yahoo! terms is a free web service provided by Yahoo! that performs term identification for English texts. As the application of the extraction algorithms for both of these approaches does not include a training phase and as Xtract identifies collocations based on corpus counts, they were evaluated over the whole Genia corpus, and not on the small test corpus. The mwetoolkit, on the other hand, was evaluated over a small portion of the corpus because the remainder was used as training data for the SVM classifier. However, the results will be reported in terms of the number of MWTs in each (sub)corpus.

The first two columns of table 1 summarise the performance of Xtract and of Yahoo! terms on the whole corpus. Both present a dramatically low recall, but the precision of Xtract is clearly better than that of Yahoo! terms. Despite the fact that Yahoo! terms shows a higher F-measure, the values in these two first columns give us an indication of the difficulty of the task. Indeed, to provide broad coverage and high quality in automatic MWE extraction is a great challenge for NLP systems.

The three last columns correspond to three different filtering configurations applied (both during training and testing) to the candidates extracted by the mwetoolkitfrom the test portion of the Genia corpus. In the first condition, we considered all candidates without any frequency threshold. In the second, we considered all the candidates which occurred more than once in the test corpus, while in the third, we kept all the candidates that occurred at least five times. The results show us that, as expected, statistical AMs calculated including candidates that occur only once are not reliable ($t = 0$), and discarding them helps to improve precision and recall ($t = 1$). A higher threshold like $t = 5$ provides even better precision at the price of drastically reducing recall, but even so recall and F-measure in this configuration are still higher than those of the baseline systems with which we compared the mwetoolkit.

For a given application, the exact value of the threshold can be customised according to whether the preference is for a higher recall or for a higher precision. For instance, if the goal is to create a terminological dictionary, a higher recall may be desirable with manual validation of the results.

The results obtained can give an idea of the usefulness of the mwetoolkit for providing more targeted domain-specific terminological extraction than a general-purpose collocation identification tool like Xtract. Moreover, mwetoolkit allows parametrisation and customisation of its various modules according to a particular application without being language- or domain-dependent. Therefore, its performance could be improved even further with better tuning to the domain or postprocessing of the results.

7 Conclusions and Future Perspectives

In this paper, we presented a new tool for automatic MWE extraction from corpora. The mwetoolkit can be used not only to speed up the work of lexicographers and terminographers in the creation of terminological resources for new domains and languages, but also to contribute to the porting of NLP systems such as Machine Translation and Information Extraction across domains. The methodology employed in the toolkit is not based on symbolic knowledge or dictionaries, and the techniques implemented in it are language independent. Therefore, it can straightforwardly be applied to any language and domain for which a corpus is available, with the execution of simple corpus preprocessing steps and the definition and tuning of POS patterns, for improved performance.

We expect, in the future, to integrate a higher number of features about the MWE candidates into the classifiers, in order to provide more accurate results. Among possible improvements are new descriptive features, contingency-table association measures and information coming from peripheral sources such as parallel corpora (word alignments) and general-purpose or domain-specific dictionaries.

Moreover, we would like to provide better integration between the candidate extraction step and the classifier construction step. Currently, the latter is performed externally using WEKA, but we believe that if this step were integrated into the toolkit's pipeline, we would increase its ease of use. Still under the perspective of usability, we would like to develop or adapt an interface for manual evaluation of the candidates and for testing the results in the context of

[9] http://dragon.ischool.drexel.edu/

LR construction.

As it stands, the performance of the `mwetoolkit` is improved by the preprocessing techniques and algorithms employed such as specialised lowercasing, tokenisation and lemmatisation. While on one hand careful preprocessing of the data is crucial to determine the quality of the output, on the other hand the techniques adopted for a particular corpus/domain may not be straightforwardly applicable to other corpora. Therefore, for future versions, we would like to investigate the use of a plethora of preprocessing alternatives such as language-independent (de)capitalisation and tokenisation tools with customisable parameters and incorporate those to the toolkit, along with the integration with a number of external language-dependent tools like lemmatisers and POS taggers (e.g. for English).

Finally, we would like to perform extensive evaluation in order to build standard machine learning models for MWT extraction in different domains and make them available. This will allow a consistent comparison of similarities and differences between domains based on the models that are created for them. All the data described in this paper, as well as the `mwetoolkit` package are freely available at `http://www.inf.ufrgs.br/~ceramisch/?page=downloads/mwttoolkit`.

8 References

Timothy Baldwin. 2005. Deep lexical acquisition of verb-particle constructions. *Comp. Speech & Lang. Special Issue on Multiword Expressions*, 19(4):398–414.

Nicoleta Calzolari, Charles Fillmore, Ralph Grishman, Nancy Ide, Alessandro Lenci, Catherine Macleod, and Antonio Zampolli. 2002. Towards best practice for multiword expressions in computational lexicons. In *Proc. of the Third LREC (LREC 2002)*, pages 1934–1940, Las Palmas, Canary Islands, Spain, May. ELRA.

John Carroll and Ted Briscoe. 2002. High precision extraction of grammatical relations. In *Proc. of the 19th COLING (COLING 2002)*, Taipei, Taiwan, Aug. ACL.

Ann Copestake, Fabre Lambeau, Aline Villavicencio, Francis Bond, Timothy Baldwin, Ivan Sag, and Dan Flickinger. 2002. Multiword expressions: Linguistic precision and reusability. In *Proc. of the Third LREC (LREC 2002)*, Las Palmas, Canary Islands, Spain, May. ELRA.

Gael Dias. 2003. Multiword unit hybrid extraction. In *Proc. of the ACL Workshop on MWEs: Analysis, Acquisition and Treatment*, pages 41–48, Sapporo, Japan, Jul. ACL.

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

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multiword terms: the C-value/NC-value method. *Int. J. on Digital Libraries*, 3(2):115–130.

Caroline Hagège, Ágnes Sándor, and Anne Schiller. 2002. Linguistic processing of biomedical texts. In Elisabete Ranchod and Nuno J. Mamede, editors, *PorTAL*, volume 2389 of *LNCS*, pages 197–208, Faro, Portugal, Jun. Springer.

Ray Jackendoff. 1997. Twistin' the night away. *Language*, 73:534–59.

John S. Justeson and Slava M. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Nat. Lang. Eng.*, 1(1):9–27.

Frank Keller, Maria Lapata, and Olga Ourioupina. 2002. Using the Web to overcome data sparseness. In Jan Hajič and Yuji Matsumoto, editors, *Proc. of the 2002 EMNLP (EMNLP 2002)*, pages 230–237, Philadelphia, PA, USA, Jul. ACL.

Jin-Dong Kim, Tomoko Ohta, Yuka Teteisi, and Jun'ichi Tsujii. 2006. GENIA ontology. Technical report, Tsujii Laboratory, University of Tokyo.

Maria da Graça Krieger and Maria José Bocorny Finatto. 2004. *Introdução à Terminologia: teoria & prática*. Editora Contexto, São Paulo, SP, Brazil.

Cédric Messiant, Thierry Poibeau, and Anna Korhonen. 2008. LexSchem: a large subcategorization lexicon for French verbs. In *Proc. of the Sixth LREC (LREC 2008)*, Marrakech, Morocco, May. ELRA.

Tomoko Ohta, Yuka Tateishi, and Jin-Dong Kim. 2002. The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proc. of the Second HLT Conf. (HLT 2002)*, pages 82–86, San Diego, CA, USA, Mar. Morgan Kaufmann Publishers.

Darren Pearce. 2002. A comparative evaluation of collocation extraction techniques. In *Proc. of the Third LREC (LREC 2002)*, Las Palmas, Canary Islands, Spain, May. ELRA.

Judita Preiss, Ted Briscoe, and Anna Korhonen. 2007. A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora. In *Proc. of the 45th ACL (ACL 2007)*, pages 912–919, Prague, Czech Republic, Jul. ACL.

Carlos Ramisch, Aline Villavicencio, Leonardo Moura, and Marco Idiart. 2008. Picking them up and figuring them out: Verb-particle constructions, noise and idiomaticity. In Alex Clark and Kristina Toutanova, editors, *Proc. of the Twelfth CoNLL (CoNLL 2008)*, pages 49–56, Manchester, UK, Aug. ACL.

Carlos Ramisch. 2009. Multiword terminology extraction for domain-specific documents. Master's thesis, École Nationale Supérieure d'Informatique et de Mathématiques Appliquées, Grenoble, France, Jun.

Ivan Sag, 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.

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

Aline Villavicencio, Valia Kordoni, Yi Zhang, Marco Idiart, and Carlos Ramisch. 2007. Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In Jason Eisner, editor, *Proc. of the 2007 Joint Conference on EMNLP and Computational*

NLL (EMNLP-CoNLL 2007), pages 1034–1043, Prague, Czech Republic, Jun. ACL.

Aline Villavicencio. 2005. The availability of verb-particle constructions in lexical resources: How much is enough? *Comp. Speech & Lang. Special Issue on Multiword Expressions*, 19(4):415–432.

Yi Zhang, Valia Kordoni, Aline Villavicencio, and Marco Idiart. 2006. Automated multiword expression prediction for grammar engineering. In *Proc. of the COLING/ACL Workshop on MWEs: Identifying and Exploiting Underlying Properties*, pages 36–44, Sidney, Australia, Jul. ACL.