# A MWE Acquisition and Lexicon Builder Web Service

*Valeria Quochi*[1] *Francesca Frontini*[1]
*Francesco Rubino*[2]
(1) ILC CNR, Pisa, Italy
(2) Synthema, Pisa, Italy
valeria.quochi@ilc.cnr.it, francesca.frontini@ilc.cnr.it,
francesco.rubino@synthema.it

ABSTRACT
This paper describes the development of a web-service tool for the automatic extraction of Multi-word expressions lexicons, which has been integrated in a distributed platform for the automatic creation of linguistic resources. The main purpose of the work described is thus to provide a (computationally "light") tool that produces a full lexical resource: multi-word terms/items with relevant and useful attached information that can be used for more complex processing tasks and applications (e.g. parsing, MT, IE, query expansion, etc.). The output of our tool is a MW lexicon formatted and encoded in XML according to the Lexical Mark-up Framework. The tool is already functional and available as a service. Evaluation experiments show that the tool precision is of about 80%.

KEYWORDS: Multiword extraction, lexical resources, LMF, web services.

*Proceedings of COLING 2012: Technical Papers*, pages 2291–2306,
COLING 2012, Mumbai, December 2012.

## 1 Introduction

Multi-word expressions (MWEs) nowadays still pose problems to most language technology and applications, e.g information retrieval, text mining, semantic web. In particular, they impact greatly on the performance of Machine Translation systems and automatic dictionary compilation. In Rule-based MT, many translation mistakes can be attributed to inadequate handling of MWEs. As for Statistical MT, they constitute a problem in the word alignment of parallel corpora. Especially for such applications MWEs can be thought of as a group of words constituting a single meaning unit and often they have an unpredictable, non-literal translation. From a practical point of view, if a group of words cannot be translated using a word-by-word translation, then it is treated as a MWE. If not recognized and handled properly, MWEs will result in mis-translations (or literal translations), and because many of them have an opaque meaning, i.e., the meaning of the unit cannot be derived by the meaning of the individual constituents that make up the unit, the translation will be incorrect and not understandable, thus hampering the overall text readability (see e.g. (Monti et al.), (Bilal, 2005)).

While many experiments on methods for the automatic acquisition of MWEs have been carried out in academic research for some decades now, readily available, and possibly customizable tools for the acquisition of MWEs in languages other than English are not so popular.

In this paper we describe the implementation of a tool for the automatic creation of MWE lexicons, integrated as a web service into a distributed platform, within the context of the PANACEA project[1]. As its goal is to set up a platform for the automatic acquisition of language resources and involves handling large or massive data, the tool described here has been implemented using robust and "computationally light" methods. The purpose is not so much to devise a new or innovative method (as the state-of-the-art seems to perform sufficiently well), but to provide a free to use tool that creates a full lexical resource from web crawled data: multi words with relevant and useful attached information that can be used for further processing (parsing, MT, information extraction, query expansion and the like). Our core extraction algorithm is language independent, making use of positional information only, and does not require a list of words to be used as seeds. The tool however has been tested and evaluated on Italian only[2].

## 2 Related works

Given the importance of MWEs for NLP applications, much research has been conducted for their automatic acquisition, with the aim of building or expanding lexica, i.e. mainly in support of lexicographic activities both for general and specific domains.

Common statistically-based acquisition approaches usually involve the following two steps: (1) the identification of candidates (usually based on *n*-grams or pattern matching)(2) filtering and candidate ranking according to some statistical score and an experimental threshold. Along these lines, different methods have been proposed in the existing literature. The oldest and simplest approaches made use of plain text corpora and identified candidates on the basis of (positional) n-grams (sequences of n adjacent words), optionally using POS filtering to clean the candidate lists and stop word lists to reduce the search space (e.g. (Choueka, 1988), (Smadja, 1993)). Also, the extraction is generally performed for a given set of seed words.

[1] see www.panacea-lr.eu

[2] In fact, recently it has been successfully run on a test Spanish corpus and it correctly produced an output lexicon, which however has not been evaluated.

More recent methods make use of tagged or parsed corpora to first identify relevant patterns in the attempt to improve precision, although for the latter the improvement is not clearly proven due to parsing errors (cfr. (Baldwin, 2005), also see the interesting review in (Seretan and Wehrli, 2009, 73-74)). The filtering and ranking of candidates is then achieved by applying some association measure (hereafter AM) calculated on the basis of co-occurrence frequency of the content words involved in candidates. Some of the most commonly used AMs are: (Pointwise) Mutual Information, Dice, Pearson's chi-squared, Log-Likelihood Ratio, Odds Ratio, Fisher's Exact tests, and various entropy measures. Several works have also carried out detailed comparisons of the methods used in the literature, evaluating the association measures used (among others, (Pearce, 2002),(Evert, 2004), (Evert and Krenn, 2005), (Pecina, 2010)). From these, we understand that the efficacy of a given AM may depend on factors like the language being analysed, the size of the corpus and the type of MWE that has to be identified. Overall, it seems that the simplest measures (frequency, MLE and Log Likelihood) perform best[3].

Although much of the work is on English data, research on MWE extraction has been carried out also for other languages, such as German (Krenn and Evert, 2001), Dutch (Villada Moiron, 2005), Czech (Pecina, 2010), French (Laporte et al., 2008), Portuguese (Villavicencio et al., 2010), among others. For Italian, the first work on collocation extraction used a window method for identifying candidates in a plain text corpus and Mutual Information for ranking (Calzolari and Bindi, 1990). Recent efforts towards the acquisition and/or production of MWE lexica for Italian are (Zaninello and Nissim, 2010), (Bentivogli and Pianta, 2002), and (Bonin et al., 2010). The former is an effort for the creation of a database of Italian MWEs annotated according to their morpho-syntactic pattern, where MWEs were given from pre-existing dictionaries. (Bentivogli and Pianta, 2002) extract from the Collins English-Italian dictionary *hidden* MWEs, i.e. MWEs not explicitly marked as such in the dictionary. (Bonin et al., 2010) extract MW terminology for two domains, adopting a contrastive approach in order to identify domain-specific multi-word terms.

Notwithstanding the vast literature, often evaluation is either not reported or not detailed enough. Also, it is often stressed that a righteous comparison of the performance is impossible due to the differences in: methods applied, targeted MWEs types, corpus used, and evaluation methodology (cfr. (Rayson et al., 2010, 3)). Precision and recall may vary considerably: for example Smadja's reports a precision of 80% for his XTRACT system on English texts (but of 40% before the syntactic-based filtering), with evaluation carried out by manual inspection by a lexicographer. (Seretan and Wehrli, 2009, 80) performed experiments in 4 languages and reported different figures for precision (English=0.42-0.58, Spanish=0.39-0.42, French=0.46-0.35, Italian=0.32-0.37).

## 3 The Panacea MW acquisition tool

Because of the need to operate in a web service distributed environment, where processing time is critical, and because of possible computing and memory limitations, it was decided to avoid computationally intensive methods. Also, as often reported in the literature, simpler methods seem to perform equally well, if not even better, in little constrained set-ups. Our MWE acquisition component is thus inspired by the seminal work by (Smadja, 1993), but integrates also more recently experimented statistical methods and association measures for filtering and ranking the acquired candidates and thus for producing a cleaner output lexicon, promoting

[3]For lack of space, we will not refer here to more sophisticated approaches that aim at measuring the degrees of fixedness and/or opacity, such as e.g. (Fazly et al., 2009).

precision over recall. The input data is part-of-speech tagged corpora in CoNLL format. The tool performs a sequence of steps each implementing different methods, in such a way as to be efficient in terms of processing time and memory usage. These steps are collocation extraction, pre-filtering and ranking by association measures, pattern extraction, pattern selection and lexicon building. In the following sections a description of each step is given.

### 3.1 Step 1: Window based collocation extractor

The search function requires no lemmas but the POS tags of the pair of tokens representing the first and last component of a multi-word, then all instances of the POS pair in the given window are retrieved. Because all words with the given POS tag will be retrieved, differently from other approaches (e.g (Smadja, 1993)), we only consider the right window. Both the POS tag pairs and the window size are passed as user-configurable parameters to the system. The output of this step is a list of candidate collocation pairs, with their related frequencies.

### 3.2 Step 2: Pre-filter and collocation ranking through association measures

A pre-filtering stage is applied based on the raw frequency distribution of the collocations for reducing memory load and discarding pairs that will provide no useful statistical evidence. This filters out pairs below a given proportional threshold. Assuming a zipfian distribution for collocation frequencies, a long tail of low frequency pairs and hapaxes[4] will normally be extracted, that needs to be filtered away. So far our algorithm allows for two kinds of filtering based on two different thresholds.

With the AverageFrequency PreFilter the threshold is set to (1):

$$\theta_{AF} = \overline{X} = \frac{\sum_{i=1}^{n} f_i}{n} \tag{1}$$

where $f_i$ is the frequency of the ith collocation/pair and $n$ is the number of collocations extracted. Thus the AverageFrequency PreFilter filters away all collocations with f below the mean $\overline{X}$. The MaxFrequency PreFilter instead sets the threshold to (2):

$$\theta_{MF} = \frac{f_1}{10} \tag{2}$$

where $f_1$ is the frequency of the most frequent collocation. This latter filter is more selective, in that it discards all pairs that have frequency less than 1/10th of the most frequent pair.

Notice that both thresholds are independent of the size of the corpus. Given the zipfian distribution for each extraction, independently from the corpus, they should identify more or less the same point in the distribution curve. As pre-filter (2) was observed to be too selective for our purposes, it will not be used in the experiments described in the rest of the paper.

After applying the pre-filter, various state-of-the-art Association Measures (AMs) can be calculated, that will be used for alternative rankings of the collocations and of the subsequently acquired full MWEs. Currently Log Likelihood and Pointwise Mutual Information have been implemented.

[4] By hapaxes here we mean word pairs of frequency 1.

### 3.3 Step 3: Pattern extraction

For each collocation in Step 2 the algorithm retrieves the complete patterns of tokens (i.e. word sequences of minimum length 2 and maximum the window size) with their raw and relative frequencies and the attached information about lemma and POS.

The rationale behind this step is to retrieve all possible intervening patterns between word A and word B. Thus, for each word pair AB, the intervening patterns are collected in a data structure that contains lemma, token and POS for each position, including A and B. This is what we shall call the “set of patterns" for the given collocations.

A distinct pattern will thus be a sequence of elements where each element is a combination of Tokens+Lemma+PoS. Frequencies for each pattern are also retrieved.

For instance, for the pair GAS-SERRA (‘gas-greenhouse’, combined frequency: 10353) the algorithm retrieves several patterns such as[5]:

a) *gas serra* (‘greenhouse gas’): frequency 7151
b) *gas ad effetto serra* (‘greenhouse effect gas’): frequency 1547
c) *gas a effetto serra* (‘greenhouse effect gas’): frequency 1365
...

### 3.4 Step 4: Pattern based collocation filtering and MWE selection

Now, the collocations retained after pre-filtering go through an additional filtering step where further collocations are discarded based on the distribution of their “set of patterns". More specifically the frequencies of all patterns in the sets are treated as vectors. The goal of the algorithm is to detect whether the vector has any significantly more frequent items; if not, the collocation itself is discarded, otherwise only the significant patterns (i.e. the outliers) are retained as good MWE candidates.

For example: GAS - SERRA produces a pattern vector *v1*:
*v1* = [7151,1,1,1,1,51,21,2,43,1,1,38,1547,1,1,10,1,2,4,1,5,1,1,1,1,1,1,14,1,1,6,5,71,1365]

where all elements of *v1* are the frequencies of patterns extracted for GAS-SERRA. So, *v1* contains three clearly outstanding patterns for this collocation (with frequencies 7151, 1547, and 1365, corresponding to the above listed patterns A, B, C ).

On the other hand, the collocation MARE-COSTA (‘sea - coast’) produces a long vector as in v2:
*v2* = [1,1,1,1,8,1,1,1, ...]

with less clearly recognizable outstanding patterns. Given our approach, this can be seen as evidence for the lower fixedness of the second collocation with respect to the first, and thus as a criterion for rejecting the collcation altogether (i.e. the collocation and its set of patterns).

In order to quantify this intuition, mean ($\overline{X}$) and standard deviation ($\sigma$) are calculated for each set of patterns.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{k}(f_i - \overline{X})^2}{k}} \tag{3}$$

[5]Here only the sequence of tokens is given

where $f_i$ is the frequency of the ith pattern $k$ is the number of patterns extracted.

We empirically assessed that only collocations whose vectors show $\sigma > 1$ have some chance of producing at least one significant pattern. This excludes those collocations that are normally made up by long series of very low frequency items, differing from each other very little in frequency. The only exception is when $\sigma = 0$ because the collocation extracts just one pattern of identical frequency. In this case the algorithm selects the pattern as a good MWE without further analysis[6].

Now that we have filtered out a lot noise and irrelevant collocations, the algorithm has to select the good patterns to be encoded in the output lexicon. This is done by using the same variance analysis of the distribution in the sets of patterns described above with the aim of selecting more than one relevant pattern. In particular, the presence of outliers is evaluated in terms of standard deviation above the mean. Empirical evidence showed that one $\sigma$ above the mean is a good enough threshold in our case. This threshold normally extracts 1-3 significant MWEs per collocation.

Thus for each collocation A+B the algorithm (which we call henceforth SigmaPatternExtraction) runs as follows (figure 1):

```
f(A+B), the frequency of a collocation (A+B)
v(A+B), the vector of pattern frequencies for each collocation
p_i(A+B), a given pattern i of A+B
f(p_i(A+B)), the frequency of p_i(A+B)
θ = 1, c = 1

if f(A+B)== f(p(A+B))
      then return p_i(A+B)
elseif σ of v(A+B) > θ
      for each p_i(A+B)
        if the f(p_i(A+B)) > c * σ(v(A+B)) + X̄(v(A+B))
          then return p_i(A+B)
```

Figure 1: Outline of the SigmaPatternExtraction Algorithm.

Higher values for both $\theta$ and $c$ result in more filtering and, possibly higher precision / lower recall.

Notice how this approach differs from the one used in (Smadja, 1993)). There the task was searching for collocates on the basis of a given list of words. Thus the vector was built in order to determine the position of any word $W$ with respect to a word from the list. Thus, in Smadja's approach a high standard deviation indicates randomness, and therefore low association strength between the two words.

In our case the pair, not a single word, is considered. Consequently, the only case when a low sigma is indicative of a good candidate pair is when $\sigma$ equals zero because its set of patterns actually contains only one element. Randomness however is null in this case. In all other cases, good pattern vectors will contain a strongly uneven distribution, with ideally one or few very frequent patterns and a long tail of very low frequency elements.

In order to evaluate the SigmaPatternExtraction Algorithm a simpler method was also devised (henceforth FirstPatternExtraction, figure 2), which consists in extracting only the most frequent

[6]This is the case for very fixed MWEs such as *stati membri*, "member states"

pattern for each collocation.

```
f(p1(A+B)), the highest value in v(A+B)

if f(A+B)== f(p(A+B)), then return p(A+B)
else return p1(A+B)
```

Figure 2: Outline of the FirstPatternExtraction Algorithm.

Notice that, so far, the system is language independent, as only distributional information is used[7]. Additional steps may be added as post-filters for further pruning the results. Such post-processors might be built ad-hoc for the language/domain in use: e.g. by using lists of stop words, or special heuristics to deal with language or tagger specific issues.

### 3.5 Step 5: The lexicon builder

The final step of the tool is lexicon building, that compiles the MWEs that were selected according to the steps/filters described above into a full XML-encoded lexicon, that conforms to the LMF standard (Gil Francopoulo, 2006)[8]. Figure 3 below exemplifies the representation of an entry[9].

```
<LexicalEntry id="0">
  <feat att="entryType" val="Multiword"/>
  <feat att="MWEPattern" val="S+E+S"/>
  <feat att="logLikelihood" val="110242.74923578261"/>
  <lemma>
    <feat att="writtenform" val="datore di lavoro"/>
  </lemma>
  <ListOfComponents>
    <Component entry="idEntry_datore">
      <feat att="rank" val="0"/>
      <feat att="pos" val="noun"/>
      <feat att="lemma" val="datore"/>
      <feat att="writtenform" val="datore"/>
    </Component>
    <Component entry="idEntry_di">
      <feat att="rank" val="1"/>
      <feat att="pos" val="prep"/>
      <feat att="lemma" val="di"/>
      <feat att="writtenform" val="di"/>
    </Component>
    <Component entry="idEntry_lavoro">
      <feat att="rank" val="2"/>
      <feat att="pos" val="noun"/>
      <feat att="lemma" val="lavoro"/>
      <feat att="writtenform" val="lavoro"/>
    </Component>
  </ListOfComponents>
</LexicalEntry>
```

Figure 3: Example of the encoding of an extracted MWE entry following LMF-XML.

At the entry level we record information on its type (MWE or NE[10]), on the POS pattern it instantiates, on the frequency and the Log Likelihood measure calculated during the acquisition process. Each entry is then characterized by a specification of the components forming the

[7] The tool is still inevitably format and tag set dependent. Still, we believe it is fairly general: the input data format is in fact CoNLL, a widely used de-facto standard, and the initial search input is a pair of POS tags which is passed as an external parameter set by the user.

[8] In fact, it is valid according to the LMF DTD-rev16.

[9] Notice that, for the sake of readability of the example, the single word LexicalEntries corresponding to each component have been omitted.

[10] NEs can be identified on the basis of simple heuristics exploiting the tagging of proper names, if available in the POS tagged corpus.

MWEs with related information: information about the rank or position of the components, their POS, orthographic form and lemma.

### 3.6 Requirements and deployment as a Web service

The tool was developed as a java application and it runs in less than 30 minutes any recent server, but it requires a maximum heap memory of 4GB for a corpus of 37 million words (250.3 MB) with window = 5. This is due to the remarkable size of the data structure containing the pairs and their intervening patterns before pre-filtering. Pre-filtering is therefore highly necessary in order to reduce the data-structure to a tractable size for further processing. The tool is then deployed as a web service using Soaplab on Apache Tomcat[11]. Services can be chained together and run as work-flows using a work-flow manager such as Taverna. A typical work-flow for our extractor will include a POS-tagger and a converter to CoNLL[12], as well as the extractor [13]. Post-filters might be developed as stand-alone services and chained to the MW_Extractor to obtain higher quality results.

## 4 Evaluation

In order to tune the extraction steps and to evaluate the tool, experiments have been performed on a domain specific corpus of Italian: the domain is ENVIRONMENT, one of the domains targeted in the PANACEA project. The tool is evaluated according the standard intrinsic evaluation procedure, against a reference resource that we will refer to as the gold standard.

Evaluation is split into three phases: 1) evaluation of the pre-filtering phase, where the removal of the long tail of low frequency co-occurrences is justified, followed by 2) the evaluation of the pattern extraction algorithm: automatic evaluation of the extracted patterns against the gold standard with the standard precision, recall and F-measures; 3) manual inspection of false positives.

This last step is necessary since, unsurprisingly, the gold standard is incomplete in terms of coverage at different levels: not only are good (domain) multi-word expressions missing, but in some cases, as an analysis of the false negatives has shown, the multi-word acquired from the corpus occurs in a different form than in the gold standard (usually the citation/lemmatised form). A manual exploration of the false positive results is thus necessary. As we are acquiring by POS patterns and not by a list of words, we are likely, and hopefully, extracting multi-word terms that are not present in existing resources, or were not considered interesting domain terms for the given glossary, but which may well be interesting and useful for NLP applications, and especially MT.

In the following sections we describe the data, the experiments and the results obtained.

### 4.1 The corpus and targeted MWEs

The (Environment) corpus used in the experiments was automatically produced with focussed monolingual crawling and cleaning services within the project[14]; its size is of about 37 million word tokens.

[11] `http://langtech3.ilc.cnr.it:8080/soaplab2-axis/services/panacea.estrattore_mw`

[12] See also (Rubino et al., 2012)

[13] The work-flows can be found on the PANACEA work-flow registry `http://myexperiment.elda.org/workflows/`

[14] http://registry.elda.org:3001/services/160, http://registry.elda.org/services/158

The evaluated extraction, henceforth called SIGMA extraction, was carried out by using the following parameters:
**target** = extraction of nominal Multiwords, i.e. multiwords whose first and last word is a noun (N-N henceforth)
**window** = 5 tokens including the first and last element (i.e. the extracted MWEs have a maximum length of 5 words)
**prefilter** = AverageFrequency PreFilter as in equation (1)
**pattern extraction** = using the SigmaPatternExtraction as in figure (1)

This extraction will be evaluated against a simpler one, henceforth FIRST extraction, where:
**pattern extraction** = uses the FirstPatternExtraction as in figure (2)

## 4.2 The gold standard

A gold standard, or reference resource, for the Environment domain has been created by semi-manually by collecting from several authoritative web glossaries and thesauri relevant nominal Italian MWEs (i.e. N-N MWEs). For each MWEs collected, its frequency in the corpus was computed using simple regular expressions to search for potential morphological variants, and never occurring MWEs are "discarded".

In the gold standard the citation forms were kept as they were found in the given resources. If the same multi-word was present in two sources with two different citation forms - e.g. singular and plural - they were not merged into one single entry in the gold standard nor was their relatedness marked.

## 4.3 Evaluation of the pre-filtering phase

Before we evaluate the core of the pattern extraction algorithm, it is important to analyze how much is lost in the pre-filtering phase. A MWE can never be extracted with our method if the corresponding collocation (that is the pair of lemmas corresponding to the first and last words of the MWE) is thrown away by the pre-filter. The extraction of the relevant collocation is thus a necessary pre-condition, although not a sufficient one, in that the pattern selection algorithm may then fail to extract the correct pattern, that is the one corresponding to the MWE.

For our evaluation we chose the Average Frequency pre-filter (1). Without this pre-filter our corpus of 37 million words produces 2,046,532 collocations, containing a long tail of hapaxes. With the Average Frequency pre-filter the collocations reduce to 259,848[15].

When evaluated against the gold standard, the non pre-filtered extraction contains 2710 eligible pairs, that is collocations whose first and last word are the same as a gold standard entry[16]. With pre-filtering eligible pairs are reduced to 1746. This may seem a heavy loss, but the 964 lost eligible pairs are to be found among 1,786,684 others. This means that the portion of the extraction that is filtered away (1,786,684 pairs) has a precision in terms of collocations of less than 0.0005 with respect to our gold standard, whereas the density in terms of eligible pairs of the remaining portion (259,848) is, 0.007 that is ten times higher[17]. Since our goal is to

[15]The lowest collocation has frequency 5, the highest has frequency 10,353

[16]The recall in terms of eligible pairs is maximum, which is not surprising, considering that the gold standard contains only MWEs that are present in the corpus.

[17]Although the gold standard is far from complete, these figures can help us getting a general picture of the distribution of our data.

achieve high precision in an acceptable processing time, this loss in recall can be considered acceptable. Notice that an eligible collocation may still not produce a genuine MWE, and that very low frequency collocations defy analysis with any association measure and are discarded in several approaches (Evert, 2004).

Evaluation of ranking is also important, since users may want to take into account only the top portion of the returned MWEs. We shall take ranking into account here, considering that association measures are a property of the collocation rather the MWE, since they are a measure of the association strength between the first and the last element of the MWE.

Two association measures - Pointwise Mutual Information and Log Likelihood - are calculated for each collocation. Collocations can be thus re-ranked by AM as well as by frequency. It is interesting to check which one is the best ranking by calculating the interpolated precision over a complete extraction. No matter what pre-filter is used to extract MWEs from collocations, raw frequency and Log Likelihood produce similar interpolated precision curves (see figure 4); both manage to rank eligible collocations at the top (frequency working slightly better), and both perform significantly better than Pointwise Mutual Information. In the figure the PMI line is almost invisible, being constantly below the baseline and close to zero in the portion taken into account. It is known in the literature (Pecina, 2010) that different kinds of AMs give different results depending on the kind of MWE extraction task. Frequency is also known to perform well (Justeson and Katz, 1995) when filtering on PoS is used.

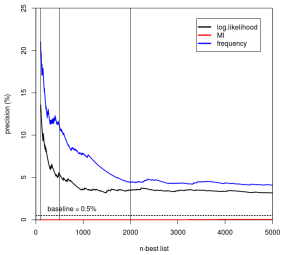


Figure 4: Interpolated precision graph for the different rankings (AverageFrequency pre-filter and SigmaPatternExtraction method). Baseline represents the average precision. The graph shows how precision progresses in the different rankings (only the first 5000 positions are shown).

## 4.4 Evaluation of MWE extraction

In this section the SigmaPatternExtraction Algorithm is evaluated. Thus two extractions are compared, the SIGMA extraction and the FIRST extraction as described above in 3.4, with FIRST acting as our baseline. Two kinds of evaluations are carried out:

SIMPLE: Simply check the extracted patterns against the gold standard. This evaluation is interesting for the recall; given that the gold standard is incomplete precision is not really significant

REDUCED: Only collocations that could produce patterns in the gold standard are selected

from the full extractions, and all MWEs related to those collocations used for evaluation. This is useful in order to evaluate precision more realistically and to allow an approximate comparison with other approaches described in the literature (see section 2 above on comparability). The "reduction" algorithm is a fairly simple one and implies some stemming, due to the fact that the gold standard is not lemmatised (and that, although UTF-8 is used, some issues related to accented chars still remain) and runs as follows 5:

```
for each mwe of length l in gold-standard
      select w_0 and the w_{l-1}
      remove the last two characters from w_0 and w_{l-1}
      add w_0 + w_{l-1} to eligible-pairs
for each mwe of length l in extraction
      select w_0 and the w_{l-1}
      remove the last two characters from w_0 and w_{l-1}
      if w_0 + w_{l-1} is in eligible-pairs
        add mwe to the reduced-set
evaluate reduced-set against the gold-standard
```

Figure 5: Outline of the reduction algorithm.

For example: if *fondo del mare* is in the gold standard, we search for pairs of the form *fon** - ma***, and return all MWEs that the algorithm has extracted for such pairs in order to evaluate them against the gold standard[18].

The results for both kinds of evaluation and for both methods are given below. Table 1 shows the results for the SIMPLE evaluation.

| *SIMPLE* | **FIRST** | **SIGMA** |
|---|---|---|
| **test** | 259848 | 209471 |
| **gold** | 2191 | 2191 |
| **precision** | 0.0038 | 0.0050 |
| **recall** | 0.4505 | 0.4770 |
| **F1** | 0.0075 | 0.01 |

Table 1: Precision and recall for the FIRST and the SIGMA extraction, with a SIMPLE evaluation method. 'Test' and 'gold' show the number of MWEs in, respectively, the evaluated extraction and the gold. While the number of MWEs in the gold remains the same, the number of MWEs in the test changes depending on the kind of algorithm used.

Notice how the test set (the number of MWEs in the extraction that are being evaluated) is smaller with SIGMA, since a number of "low sigma" pairs have been filtered out by the SigmaPatternExtraction Algorithm. Still both precision and recall are increased, that is more good patterns are extracted by not just stopping at the most frequent ones (see Figure 4 for the precision graph). Table 2 shows the results for the REDUCED evaluation.

While in the non-reduced evaluation the SIGMA method was increasing the precision due to the reduction of the test set, in the reduced evaluation FIRST seems to perform better in terms

[18] Notice how this stemming method is quite unsophisticated, and produces more pairs than it should. In this case it would also extract expressions such as *fonti nel mare*, if extracted. This means that the precision figures given for the REDUCED evaluation are probably slightly underestimated.

| *REDUCED* | **FIRST** | **SIGMA** | **SIGMA** +edit | **SIGMA** manual |
|---|---|---|---|---|
| **test** | 1746 | 2105 | - | - |
| **gold** | 2191 | 2191 | - | - |
| **precision** | 0.56 | 0.50 | 0.60 | 0.81 |
| **recall** | 0.45 | 0.48 | 0.58 | 0.60 |
| **F1** | 0.50 | 0.49 | 0.59 | 0.67 |

Table 2: Precision and recall for the FIRST and the SIGMA extractions, with a REDUCED evaluation method

of precision although SIGMA retrieves a higher absolute number of true positives (1043 vs 984) and thus improves in terms of recall. The reason for this is to be found in a higher test set for SIGMA. This is not surprising if we consider how the REDUCED evaluation is carried out. In this case many of the low sigma pairs that the SIGMA algorithm removes are filtered away by the reduction algorithm also for FIRST; at the same time SIGMA extracts more than one pattern per pair, thus ending up with a higher (reduced) test set. Given that it is infrequent for MWEs from the same collocation (same first and last word) to be present in the gold standard, extracting more than one pattern per collocation, as SIGMA does, is penalizing for the precision with respect to our gold standard. Still, a quick manual check of the false positives reveals that most of the extracted patterns are actually correct, in that they are variants of the first pattern, such as:

```
fonte di inquinamento > fonti di inquinamento
```

Thus, if a more flexible comparison is applied, such as allowing for edit distance (Damerau, 1964) up to 3 between the strings, these variants are recognized as true positives and for SIGMA improves by 10 points, as is shown in the fourth column of table 2.

As shown by the interpolated precision graph for this last evaluation (figure 6), when reducing the set of collocations, association measures actually perform better than simple frequency.

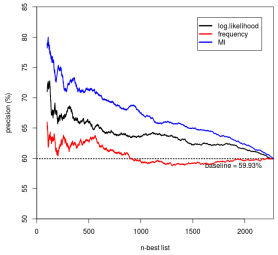


Figure 6: Interpolated precision graph for Pointwise Mutual Information, Log Likelihood and frequency for an extraction with AverageFrequency pre-filter and SigmaPatternExtraction. REDUCED evaluation method; match with edit distance = 3

### 4.4.1 Manual evaluation of MWE extraction

Further manual inspection of the false positives shows that precision is much higher in fact.

For instance the gold standard contains *zona di pressione* ('pressure zone'), which selects ZONA-PRESSIONE as an eligible pair. This collocation is thus retained in the REDUCED evaluation set, extracting from the corpus *zona di pressione* and *zona di bassa pressione* ('low pressure zone'). The latter is not contained in the gold standard, but is in fact a genuine MWE. By analysing false positives for the REDUCED evaluation and adding the good MWEs found to our gold standard, we obtain a precision of 81% (see table 2).

As the REDUCED evaluation method roughly simulates an extraction from a fixed, predefined set of targeted lemmas, as is usually the case with experiments reported in the papers, it allows for some comparison with other approaches.

Our result is thus in line with the precision performance of Smajda's XTRACT (Smadja, 1993)[19].

However, we observed that with the REDUCED evaluation, much of the noise present in the data (mainly due to the automatically crawled nature of the corpus) was filtered out through the reduction of eligible pairs (and therefore did not impact on the evaluation scores). To arrive at a more realistic assessment of the tool then a manual evaluation of the complete extraction (that is without evaluation filter) was also attempted. Considering the large amount of results obtained by our SIMPLE method, our priority was to verify the precision of the top portion of our extraction. This is meant to ensure that a user who blindly extracts MWEs from a (domain) corpus will get a significant amount of genuine results in the higher ranks. As we have seen when evaluating the pre-filter, our best ranking is raw frequency, followed by LogLikelihood. We thus evaluated the first 1000 highest frequency results against the gold standard, then we checked the false negatives and added such as turned out to be genuine to the gold standard for a second run.

$$\text{first run precision} = 0.40, \text{second run precision} = 0.78$$

The results of the first run tell us that 40% of the MWEs from the original gold are to be found in the top 1000 results. The second run, most significantly, tells us that 78% of the first 1000 returned results are correct, and that it is thus possible to extract MWEs without seed words and still get a precision that approaches the state of the art. Notice also how the "real" precision of the top 1000 portion of the extraction is just 2% lower than the one obtained with the REDUCED evaluation method, the latter thus providing a fairly good approximation for the first.

## Conclusion and Future Work

We have presented a tool for the acquisition of multi-word expressions of various lengths and types that generates an LMF MWEs lexicon as output. The tools is already functional and has been deployed as a web service within a distributed platform that deals with large/massive data. First evaluation results (with a reduced extraction that is rendered comparable to the gold standard) are encouraging. If possible, we plan to extend the manual evaluation in order to have a more accurate estimate of the real precision.

Regarding the service, future improvements are possible especially based on users' feedback in particular regarding the properties to be left as configurable parameters and output formats. Regarding the tool, we intend to continue improving the method by:

[19] In fact, our system is likely even to outperform it given that the evaluation carried out is slightly penalizing for the system both because of the naive stemming and because the gold-standard also contains MWEs with frequency 1 and 2, which are most likely not retrieved by the system.

1. Adding further filtering/cleaning of the extracted MWEs. In particular we might experiment with merging patterns that are sub strings of others when they have the same frequency, as they create noise and lower precision. For instance: *sicurezza sul posto* (lit. 'safety on the place') and *sicurezza sul posto di lavoro* (lit. 'safety on the place of work') all have the same frequency (630); clearly, the genuine MWE is *sicurezza sul posto di lavoro* and the others are substrings thereof.

2. Extracting MWEs with PoS patterns by progressive test and reduction of patterns for learning the "free slots" in the multiwords. For instance we may want to derive a pattern of the form *articolo NUM della legge* ("article NUM of the law") from a series of patterns of the form: *articolo 6 della legge, articolo 12 della legge, articolo 23 della legge, ....*

3. Language specific fine tuning will also be implemented, in the form of post-filtering (e.g. via optional stop words list and legitimate patterns check), as well as of post-editing (e.g. with head detection heuristics).

4. Automatic conversion to, or direct output in-, an RDF format (e.g. according to the Lemon model), as to make the resource publishable as L(O)D potentially exploited for reasoning or by other web services[20].

Regarding the evaluation, the next step will be a task based evaluation. Interesting tasks could be: rule-based machine translation, syntactic parsing, or subcategorisation frame acquisition (which would be interesting to assess the impact of automatically acquired multi-word prepositions).

## Acknowledgments

This work has been realized at CNR-ILC within the EU FP7 funded project PANACEA (Platform for Automatic. Normalized Annotation and Cost-Effective Acquisition of Language Resources for Human Language Technologies) under grant agreement n. 248064. We thank PANACEA reviewers and the COLING anonymous reviewers for their comments and incentives to improve. In particular, we want to thank Fabio Affé who helped us with the final modifications to the java code and with the deployment in Soaplab.

[20]We thank an anonymous reviewer for his/her suggestions concerning LOD and NIF. Although the NLP Interchange Format (`http://nlp2rdf.org/nif-1-0`) seems to be a format for corpus data, while our tool outputs a lexicon, in our future work within the LOD framework we will attempt to ensure compatibility with NIF in terms of basic vocabulary and data categories used.

## References


Baldwin, T. (2005). Deep lexical acquisition of verb-particle constructions. *Comput. Speech Lang.*, 19(4):398–414.

Bentivogli, L. and Pianta, E. (2002). Detecting hidden multiwords in bilingual dictionaries. In EURALEX., editor, *Proceedings of the Tenth EURALEX International Congress*. Center for Sprogteknologi.

Bilal, K. (2005). Extracting Multiword Expressions in Machine Translation from English to Urdu using Relational Data Approach. *ENFORMATIKA International Transactions on Engineering, Computing and Technology*, 6:312–314.

Bonin, F., Dell'Orletta, F., Montemagni, S., and Venturi, G. (2010). A contrastive approach to multi-word extraction from domain-specific corpora. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Calzolari, N. and Bindi, R. (1990). Acquisition of lexical information: from a large textual italian corpus. In *Proceedings of the 13th conference on Computational linguistics - Volume 3*, COLING '90, pages 54–59, Stroudsburg, PA, USA. Association for Computational Linguistics.

Choueka, Y. (1988). Looking for needles in a haystack or locating interesting collocation expressions in large textual databases. In *Proceedings of the RIAO*, pages 38–43.

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176.

Evert, S. (2004). The statistics of word cooccurrences: word pairs and collocations. *Unpublished doctoral dissertation, Institut fuer maschinelle Sprachverarbeitung, Universitaet Stuttgart*.

Evert, S. and Krenn, B. (2005). Using small random samples for the manual evaluation of statistical association measures. *Computer Speech and Language*, 19(4):450 – 466. Special issue on Multiword Expression.

Fazly, A., Cook, P., and Stevenson, S. (2009). Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, 35(1):61–103.

Gil Francopoulo, Laurent Romary, M. M. N. C. (2006). Lexical Markup Framework (LMF). In *LREC 2006*.

Justeson, J. S. and Katz, S. M. (1995). Principled disambiguation: Discriminating adjective senses with modified nouns. *Computational Linguistics*, 21(1):1–27.

Krenn, B. and Evert, S. (2001). Can we do better than frequency? A case study on extracting PP-verb collocations. In *Proceedings of the ACL Workshop on Collocations*, Toulouse, France.

Laporte, E., Nakamura, T., and Voyatzi, S. (2008). A French Corpus Annotated for Multiword Nouns. In *Proceedings of the Language Resources and Evaluation Conference. Workshop Towards a Shared Task on Multiword Expressions*, pages 27–30, Marrakech, Maroc.

Monti, J., Barreiro, A., Elia, A., Marano, F., and Napoli, A. Taking on new challenges in multi-word unit processing for machine translation. In *Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation, net/10609/5646*, year=2011.

Pearce, D. (2002). A comparative evaluation of collocation extraction techniques. In *In Third International Conference of on Language Resources and Evaluation*, Las Palmas, Spain.

Pecina, P. (2010). Lexical association measures and collocation extraction. *Language Resources and Evaluation*, 44:137–158.

Rayson, P., Piao, S., Sharoff, S., Evert, S., and Moirón, B. n. V. (2010). Multiword expressions: hard going or plain sailing? *Language Resources and Evaluation*, 44:1–5.

Rubino, F., Frontini, F., and Quochi, V. (2012). Integrating nlp tools in a distributed environment: A case study chaining a tagger with a dependency parser. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Seretan, V. and Wehrli, E. (2009). Multilingual collocation extraction with a syntactic parser. *Language Resources and Evaluation*, 43(1):71–85.

Smadja, F. (1993). Retrieving collocations from text: Xtract. *Comput. Linguist.*, 19(1):143–177.

Villada Moiron, B. (2005). *Data-driven identification of fixed expressions and their modifiability*. PhD thesis, University of Groningen.

Villavicencio, A., Ramisch, C., Machado, A., de Medeiros Caseli, H., and José Finatto, M. (2010). Identificação de Expressões Multipalavra em Domínios Específicos. *Linguamàtica*, 2(1):15–33.

Zaninello, A. and Nissim, M. (2010). Creation of Lexical Resources for a Characterisation of Multiword Expressions in Italian. In *LREC 2010*.