Automatic extraction and evaluation of MWE

Leonardo Zilio[1], Luiz Svoboda[2], Luiz Henrique Longhi Rossi[2], Rafael Martins Feitosa[2]

[1]Programa de Pós-Graduação em Letras da Universidade Federal do Rio grande do Sul

[2]Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande do Sul

leonardozilio@yahoo.de, luizek@gmail.com, lh.rossi@gmail.com, rafael.feitosa@ufrgs.br

Abstract. *This short paper aims at presenting a method for automatically extracting and evaluating MWE in the Europarl corpus. For this purpose we make use of mwetoolkit and utilize its output to find rules for the automatic evaluation of MWE. We then developed an XML parser to evaluate MWE candidates against those rules and also against online dictionaries. A sample of the results was manually evaluated by linguists and we had 87% of precision.*

1. Introduction

The automatic extraction of multiword expressions (MWE) is an important task not only for lexicographical purposes, but also for Natural Language Processing, because the recognition of these compound expressions helps in the process of automatically understanding written or spoken texts.

Thinking of the various possibilities of use that MWE represents, this study aims at presenting some difficulties for the extraction of MWE and shows the method applied to automatically extract 1,885 MWE using the mwetoolkit (Ramisch et al. 2010a; Ramisch et al. 2010b) associated with other resources.

In the Section 2 of this study we present the corpus and all the steps we followed for the automatic extraction and validation. In Section 3 we describe the results of our study and the results of the manual validation. Finally, in Section 4 we discuss our results and the possibilities for future work.

2. Method

2.1 Corpus

For the purposes of this study, we selected the full Europarl corpus (Koehn 2005) as source for MWE. It has a relatively large size – if we take into consideration that our results were not only automatic evaluated, but also manually validated – consisting of 43,919,903 running words as of October 2010 (version 4). The size of the Europarl varies from time to time, since it incorporates the sections of the European Parliament and is updated on a regular basis. The selected language was English.

Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology, pages 214–218, Cuiabá, MT, Brazil, October 24–26, 2011.

2.2 Steps of automatic extraction and evaluation

This study was developed through a series of steps that we describe below. They range from the preprocessing for and with the mwetoolkit to the automatic evaluation using our developed XML parser.

2.2.1 First step – mwetoolkit (pre)processing

The first thing to do is preprocessing the corpus with the Tree Tagger (Schmid 1994; Schmid 1995). Those tags will then be simplified by the mwetoolkit for its own purposes.

We then ran the mwetoolkit looking for only five bigram patterns: N + N; N + Num; A + N; V + N; V + P. The extraction was made excluding candidates that occurred less than 10 times. The extraction output is then presented in XML and ARFF files. This first extraction of only bigrams was done because we needed an automatic preclassification for the extraction of rules (as explained next), and mwetoolkit's gold standard only accounts for bigrams.

2.2.2 Second Step – Extraction of rules

After extracting the MWE candidates, the mwetoolkit computes various association measures (Maximum Likelihood Estimator = MLE; Pointwise Mutual Information = PMI; T-score = T; Dice's coefficient = DICE; and Loglikelihood = LL) for them and also compares them with an internal gold standard for preclassification purposes. At the end of the process, mwetoolkit generates an ARFF file, which contains information on association measures and gold standard preclassification (it marks the MWE candidates as "true" or "false") and can be used in Weka (Hall et al. 2009) for machine learning.

In Weka, we used some implemented algorithms with the ARFF file to obtain the threshold values and decide which of MLE, PMI, T, DICE and/or LL would be useful. And this was the most difficult part. Since the candidates in the ARFF files classified as True by the gold standard were very sparse, we couldn't extract good results with its raw form. This happened because the results seem much alike the ones of a random validation. So we processed the results a bit further.

At first, we excluded the MLE value, because it was much too sparse, and many of the MWE candidates didn't have this value. The second step was taking away the MWE candidates which didn't have any of the values. The last filter was the SMOTE (Synthetic Minority Over-sampling TEchnique) (Chawla et al. 2002), which is a method of over-sampling the minority class and under-sampling the majority class to achieve a better classification quality with the nearest neighbor value 5.

Using T, DICE and LL values we obtained the best results using the JRIP (Cohen 1995), which is an optimized rule-based implementation of the IREP (Fürnfranz and Widmer 1994) algorithm. With JRIP we could extract the following rule, with 67,6% precision:

- T values over 5.57;
- DICE values over 0.02;
- LL values between 51000 and 22712.

Although we had three values, the LL formula used by the mwetoolkit is only

applicable to bigrams, and our study, as will be shown in the next section, comprised MWE that ranged from bigrams to hexagrams, so the LL value was disregarded. PMI was disregarded in the rule generated by JRIP.

2.2.3 Third step – Full extraction of patterns

With this rule for automatic validation, we reprocessed the corpus with the mwetoolkit, but this time we used 26 patterns suggested by a linguist. Although not complete, it represents a good set.

N+N / N+N+N / N+N+N+N / A+N / A+N+N / A+A+N / A+A+N+N / A+N+N+N
A+N+N+N+N / N+Num / A+A+A+N / A+A+N+N+N / V+N / V+N+N / V+DT+N
V+DT+N+N / V+DT+A+N / V+DT+A+N+N / V+P / V+P+N / V+P+DT+N / V+P+DT+N+N
V+P+DT+A+N / V+P+DT+A+N+N / V+P+A+N / V+P+A+N+N

The result of this extraction was also confronted against the mwetoolkit's gold standard. Since it only comports bigrams, we needed something more complex to validate the other n-grams. For this purpose, we developed a XML parser, which is explained in the next section.

2.2.4 Fourth step – XML Parser

As part of this study, we developed a tool to automate the evaluation process of the XML file generated by the mwetoolkit. This tool aims at classifying each MWE candidate as a true or false MWE. For its development, we used Java.

This tool analyses the XML using the Document Object Model (DOM)[1] through the Java API for XML Processing (JAXP)[2]. By using DOM, we had an easy way to manipulate the XML file and execute arbitrary modifications. The candidates are then retrieved and checked against a stoplist of treatment pronouns, so as to remove MWE candidates with those kind of words. After this, the association measures are verified against the thresholds (from Section 2.2.2) and classified as True or False. All candidates marked as False in the previous step are then checked against the Free Dictionary[3], if the candidate is present, then it is reclassified as True.

3. Results and validation

The final extraction, using 26 patterns of MWE candidates, returned more than 82 thousand MWE candidates, as we can see in Table 1. Since the automatic evaluation with the Free Dictionary is rather time consuming, and our final objective was to retrieve only the necessary amount for manual validation, we divided those candidates and automatically evaluated only the first 17,528 MWE candidates. From these, 1,885 were automatically marked as True (10.75%).

After using the XML parser for the automatic evaluation, we started the manual

[1] http://www.w3.org/DOM/
[2] http://jaxp.java.net/
[3] http://www.thefreedictionary.com/

validation purposes, which was made by 3 linguists[4] using a random sample that contained the first 100 MWE candidates marked as True, and the first 100 MWE candidates marked as False.

Table 1. Results of the extraction and automatic evaluation

Method	# of Patterns	# of Automatically evaluated MWE candidates	# of True
mwetoolkit, Threshold and Free Dictionary	26	17,528 (from more than 82 thousand)	1,885

The results can be seen in Table 2. From the 100 MWE candidates automatically evaluated as True, 87% were validated as true positives. Among their False counterpart, 19% were validated as false negatives.

Table 2. Confusion matrix of the 200 MWE candidates sample

	Validated as True	Validated as False	Total
Marked as True	87	13	100
Marked as False	19	81	100
Total	106	94	

With this results, we computed Precision, Recall and F-measure, which can be seen in the Table 3 below.

Table 3. Results: Precision, Recall and F-measure based on the validated sample

Precision	Recall	F-measure
0.87	0.82	0.84

4. Discussion

The results we found had a good percentage of correctness in the automatic evaluation of the extracted MWE candidates, with .87 of precision, and a good result for coverage, with .82 of recall. The use of The Free Dictionary seems to have been a right step towards the improvement of precision in the automatic evaluation, as were the thresholds evaluated for the association measures.

Although the results were encouraging, this study has its limitations. It was done using a relatively large corpus of language for general purposes, but we can't assure the same results will be found for language for special purposes. We believe, though, that this limitation may be overcome by the use of other online, specialized dictionaries, which is a goal for future works.

We also need to check the performance of the individual MWE patterns that were used in this study, so that we can see which ones are better suited for automatic extraction.

[4] We used only three linguists because we couldn't count on more of them. Also, the number allows for no draw.

References

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. (2002). Synthetic Minority Over-sampling Technique. In: *Journal of Artificial Intelligence Research*, 16, p. 321-357. New Brunswick, NJ: Morgan Kaufmann.

Cohen, W. W. (1995) Fast Effective Rule Induction. In: *Proceedings of the Twelfth International Conference on Machine Learning*, p. 115-123. New Brunswick, NJ: Morgan Kaufmann.

Fürnkranz, J.; Widmer, G. (1994) Incremental reduced error pruning. In: *Proceedings of the 11th International Conference on Machine Learning* (ML-94), p. 70-77. New Brunswick, NJ: Morgan Kaufmann.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P; Witten, I. (2009) The WEKA Data Mining Software: An Update. In: *ACM SIGKDD Explorations Newsletter*, Volume 11, Issue 1, p. 10-18. New York, NY: ACM.

Koehn, P. (2005) Europarl: A Parallel Corpus for Statistical Machine Translation. In: *Machine Translation Summit*, p. 79-86.

Ramisch, C.; Villavicencio, A.; Boitet, C. (2010a) Web-based and combined language models: a case study on noun compound identification. In: 23rd International Conference on Computational Linguistics (Coling), 2010, Pequim. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Morristown, NJ: Association for Computational Linguistics, p. 1041-1049.

Ramisch, C.; Villavicencio, A.; Boitet, C. (2010b) mwetoolkit: a Framework for Multiword Expression Identification. In: *Proceedings of the Seventh conference on International Language Resources and Evaluation* (LREC'10), p. 662-669.

Schmid, H. (1994) Probabilistic Part-of-Speech Tagging Using Decision Trees. In: *Proceedings of the International Conference on New Methods in Language Processing* (NeMLaP-1), p. 44–49.

Schmid, H. (1995) Improvements in part-of-speech tagging with an application to German. In: Helmut Feldweg and Erhard Hinrichts (Eds.) *Lexikon und Text*. Tübingen: Niemeyer, p. 47-50.