A Comparative Study of Different Classification Methods for the Identification of Brazilian Portuguese Multiword Expressions

Alexsandro Fonseca **Fatiha Sadat**

Université du Québec à Montréal, 201 av. President Kennedy,
Montreal, QC, H2X 3Y7, Canada
affonseca@gmail.com sadat.fatiha@uqam.ca

Abstract

This paper presents a comparative study of different methods for the identification of multiword expressions, applied to a Brazilian Portuguese corpus. First, we selected the candidates based on the frequency of bigrams. Second, we used the linguistic information based on the grammatical classes of the words forming the bigrams, together with the frequency information in order to compare the performance of different classification algorithms. The focus of this study is related to different classification techniques such as support-vector machines (SVM), multi-layer perceptron, naïve Bayesian nets, decision trees and random forest. Third, we evaluated three different multi-layer perceptron training functions in the task of classifying different patterns of multiword expressions. Finally, our study compared two different tools, MWEtoolkit and Text-NSP, for the extraction of multiword expression candidates using different association measures.

1 Introduction

The identification of multiword expressions (MWEs) and their appropriate handling is necessary in constructing professional tools for language manipulation (Hurskainen, 2008). MWEs are considered as a very challenging problem for various natural language processing (NLP) applications, such as machine translation.

There are several definitions of MWE in the scientific literature. Smadja (1993) defines MWE as an arbitrary and recurrent word combination; while Choueka (1988) defines them as a syntactic and semantic unit whose exact meaning or connotation cannot be derived directly and unambiguously from the meaning or connotation of its components. Moreover, Sag et al. (2002) defines MWE as an idiosyncratic interpretation that exceeds the limit of the word (or spaces).

We adopt in this paper a definition similar to the one given by Sag et al. (2002): a MWE is an expression formed by two or more words, whose meaning can vary from totally dependent to completely independent of the meaning of its constituent words. Examples of MWEs: “take care”, “Bill Gates”, “coffee break” and “by the way”.

This study treats only two-word MWEs. We are not considering some common Portuguese MWEs, such as “tempo de espera” (waiting time, lit.: time of waiting), “dar um tempo” (to have a break, lit.: to give a time) or “começar tudo de novo” (restart, lit. start everything of new). However, our experience and some related work show that we are already covering the majority of MWEs. For their data, for example, Piao et al. (2003, Section 5) found that 81.88% of the recognized MWEs were bigrams. Moreover, our focus is in MWE formed by nouns, adjectives, verbs and adverbs. As a consequence, two-word MWEs formed by prepositions were not considered, such as “de novo” (again, lit. of new), “à toa” (for nothing), “apesar de” (despite of) or “desde ontem” (since yesterday). In resume, we evaluated the performance of different classification algorithms and tools for the recognition of two-word MWEs formed by nouns, adjectives, verbs and adverbs. We intend, in the future, to extend this study to MWEs formed by words belonging to any grammatical class and having any number of words.

Proceedings of the First Workshop on Computational Approaches to Compound Analysis, pages 53–62,
Dublin, Ireland, August 24 2014.

The correct identification of MWEs is important for different NLP applications, such as machine translation, information retrieval and the semantic web, to which the principle of syntactic or semantic unit is important (Watrin and François, 2011).

Methods for identifying MWEs rely on statistical measures, especially association measures, such as mutual information (Church and Hanks, 1990), log-likelihood or Dice's coefficient (Smadja, 1996).

The basic idea behind such measures can be summarized as follows: the higher the association among the words that appear together in a text, the higher the probability that they constitute a single semantic unit.

There are other methods, which use linguistic information or hybrid approaches that combine statistical measures with the linguistic information, such as the grammatical class of each word, the sense of the composite expression or the syntactic regularities.

2 Related Work

Dias and Lopes (2005) present a method for the extraction of MWEs based only on statistics with an application on the Portuguese language. This method consists of a new association measure called "mutual expectation". Their method can be applied to extract MWEs formed by two or more words, contiguous or not. The mutual expectation method is based on the LocalMaxs (Silva and Lopes, 1999) algorithm. This algorithm deduces that a n-gram is a MWE if the degree of attraction between its words is greater or equal to the degree of attraction of all its subsets of n-1 words (i.e. all groups of n-1 words contained by the n-gram) and if it is strictly greater than the degree of attraction of all of its super groups of n+1 words (i.e. all groups of n+1 words containing the n-gram). When the n-gram is a bigram (n = 2), only the degree of attraction of its super groups of n+1 words is calculated.

Ramisch et al. (2008) analyze the extraction of MWEs based only on statistical information, comparing three association measures: the mutual information, chi-squared and permutation entropy. Then they introduce a method called entropy of permutation and insertion (a hybrid approach), that takes into account linguistic information of the MWE type. Following some patterns, they modify each original MWE candidate by inserting some types of words in some positions and they test if the new MWE are still MWE and they try to identify which kind of modification an MWE type accepts or refuses in a particular language. The new measure is calculated using a formula that combines the probability of occurrence of the original and of the generated MWE.

Agarwal et al. (2004) present an approach for extracting MWEs in languages with few resources based on a morphological analyser and a moderate size untagged text corpus. First, they divide the MWEs in categories. For example, Category-2 is formed by noun-noun, adjective-noun and verb-verb bigrams. Then they apply a set of rules to identify or eliminate candidates as MWE. Those rules take into consideration the precedent and/or the next word in the pair and the possible inflections of the words. After this step, association measures are computed.

Piao et al. (2003) use, what they call, a semantic field annotator. They use a semantic tagger for the English language called USAS, developed in Lancaster University. This tagger labels words and expressions in a text using 21 categories. For example, Category-A is used for "general and abstract terms", Category-B is used for "the body and the individual", Category-E is used for "emotion", etc. A text labeled with those categories is used to extract the MWE candidates. The differential of this approach is that the candidates are selected not based only on statistical measures. The problem with this is that most of the MWEs, about 68% in the work of Piao et al., appear in the text with a low frequency. As a consequence, most of the methods for extracting MWEs give good precision, but low recall.

3 The Data

The current study used the corpus CETENFolha (Corpus de Extractos de Textos Eletrónicos NILC/Folha de São Paulo), available on the website Linguateca Portuguesa (CETENFolha, 2008). This corpus is composed by excerpts from Brazilian newspaper "Folha de São Paulo", and contains over 24 million words. It is part of a project on the automatic processing of the Portuguese (Kinoshita et al., 2006). As the current stage, we used a small fraction of the corpus, composed by 3,409 excerpts of text (about 250,000 words). Each excerpt corresponds to individual news, which covers different areas.

4 Comparison of different classification algorithms

4.1 Pre-processing the data

Before the indexation, some pre-processing methods on the corpus were completed, such as lemmatization and elimination of stop words (articles, prepositions, conjunctions). In this study, we are mostly interested in analyzing MWEs formed by nouns, adjectives, adverbs and verbs. And since those stop words are very common in Portuguese, their elimination reduces considerably the number of MWE candidates that would not be relevant to this study.

We created two indexes: one formed only of bigrams and the other only by unigrams. Our results show 49,589 bigrams, with 1,170 having a frequency higher than 3. We selected those 1,170 bigrams as our MWE candidates. By hand, from the 1,170 candidates, we recognized 447 as being Portuguese MWEs.

The main criterion used to consider a bigram as a MWE was that the bigram had a sense on its own. For example: proper names, like "Adelson Barbosa", "George Bush" and "Belo Horizonte"; support verb constructions: "tomar cuidado" (to take care), "fazer sentido" (to make sense); expressions having some idiomatic sense: "abrir mão" (to give up, lit. to open hand), "fazer questão" (to insist, to require [that something be done in a specific way], lit. to make question); fixed expressions: "bens duráveis" (durable goods), "senso comum" (common sense), "curto prazo" (short term). Example of bigrams not considered as MWE: "Brasil foi" (Brazil was), "apenas dois" (only two), "bomba matou" (bomb killed), etc.

For each bigram, we found the frequency of its constituent words in the unigram index. Then, we classified by hand each of the words by their grammatical class: 1 for nouns, 2 for adjectives, 3 for verbs, 4 for other classes (mostly adverbs and pronouns) and 5 for proper names. This gave us 25 patterns of bigrams: N-N, N-ADJ, N-V, V-N, PN-PN, etc. We decided not to use a POS-tagger, to ensure that each word would have its grammatical class assigned correctly, creating the most correct possible training and testing data sets for the classification algorithms.

We then created a matrix of 1,170 lines and five columns. For each line, the first column represents the frequency of a bigram in the excerpt of text, the second column represents the frequency of the first bigram's word, the third column represents the frequency of the second bigram's word, the fourth column represents the grammatical class of the first bigram's word and the fifth column represents the grammatical class of the second bigram's word. This matrix was used to evaluate the precision and recall of different classification algorithms.

4.2 Evaluation

We applied nine different classification algorithms to our data set. The parameters used with each algorithm are listed below.

Decision tree: C4.5 algorithm (Quinlan, 1993) with confidence factor = 0.25.

Random Forest (Breiman, 2001): number of trees = 10; max depth = 0; seed = 1.

Ada Boost (Freund and Schapire, 1996): classifier = decision stamp; weight threshold = 100; iterations = 10; seed = 1.

Bagging (Breiman, 1996): classifier = fast decision tree learner (min. number = 2; min. variance = 0.001; number of folds = 3; seed = 1; max. depth = -1); bag size percent = 100; seed = 1; number of execution slots = 1; iterations = 10.

KNN (Aha and Kibler, 1991): K = 3; window size = 0; search algorithm = linear NN search (distance function = Euclidian distance).

SVM (Chang and Lin, 2001): cache size = 40; cost = 1; degree = 3; eps = 0.001; loss = 0.1; kernel type = radial basis function; nu = 0.5; seed = 1.

Multilayer perceptron: learning rate = 0.3; momentum = 0.2; training time = 500; validation threshold = 500; seed = 0;

Bayesian net: search algorithm = k2 (Cooper and Herskovits, 1992); estimator = simple estimator (alpha = 0.5).

As we can see in Table 1, the values of precision are very similar for all the algorithms, varying between 0.830 (random forest) and 0.857 (bagging), with the exception of SVM, which gave a precision of 0.738. The recall values were between 0.831 and 0.857 (0.655 for SVM).

We obtained good precision and recall. However, we must consider that the values of recall are based only on the MWEs present in our reference list, and not in the entire corpus, since we could not count all the MWEs present in the corpus.

Algorithm	TP Rate	FP Rate	Precision	Recall
Decision tree	0.853	0.158	0.854	0.853
Random forest	0.831	0.194	0.830	0.831
Ada boost	0.837	0.196	0.836	0.837
Bagging	0.857	0.163	0.857	0.857
KNN – k = 3	0.846	0.171	0.846	0.846
SVM	0.655	0.553	0.738	0.655
M. perceptron	0.852	0.174	0.851	0.852
Naïve B. net	0.836	0.170	0.839	0.836
Bayesian net	0.842	0.170	0.843	0.842

Table 1: True-positive rate, false-positive rate, precision and recall for nine classification algorithms.

5 Bigrams patterns classification

We chose one of the algorithms with the best performance (multi-layer perceptron) and we evaluated it using three different training functions, bayesian regulation back propagation (br), Levenberg-Marquardt (lm) and scaled conjugate gradient (scg), and compared their performance in the classification of different patterns of bigrams as MWE. For this comparison we used the patterns that gave 10 or more samples of MWEs. We had eight patterns that together represent 59% of the candidate bigrams (689/1,170) and 94% of the MWEs that appear three or more times in the corpus (420/447). The tables 2.a, 2.b and 2.c show the results. "N" stands for "Noun", "A" for adjective, "O" for other classes (mostly adverbs and pronouns) and "PN" for "proper names".

Analyzing the three tables, we see that we had best results with the patterns N-A (e.g. "agencias internacionais", "ajuste fiscal", "America Latina") and PN-PN ("Adelson Barbosa", "Ayrton Senna", "Bill Clinton"). The function lm gave the best value for the F1 measure (0.912) for the pattern N-A, and the function scg gave the best value for the pattern PN-PN (0.931).

In general, we obtained the weakest results with the patterns O-N (e.g. ex-presidente, primeiro mundo) and A-PN (São Paulo, Nova York). Using the training functions "lm" and "scg", none of the 10 MWEs belonging to the pattern O-O (apesar disso, além disso) was recognized, and none of the 46 MWEs belonging to the pattern O-N was recognized, when using the training function "scg".

The last line of each table show the total values for the eight patterns, for the three learning functions. We had the best precision and recall using the "lm" function.

Pattern	Bigrams	MWE	TP	FP	TN	FN	Prec.	Recall	F1
N-A	229	193	176	27	9	17	0.867	0.912	0.889
O-N	164	46	14	23	95	32	0.378	0.304	0.337
PN-PN	117	101	94	15	1	7	0.862	0.931	0.895
A-N	53	21	13	3	29	8	0.813	0.619	0.703
O-O	46	10	5	9	27	5	0.357	0.500	0.417
N-PN	34	16	7	9	9	9	0.438	0.438	0.438
N-N	31	20	11	6	5	9	0.647	0.550	0.595
A-PN	15	13	3	1	1	10	0.750	0.231	0.353
All Pat.	689	420	323	93	176	97	0.776	0.769	0.773

Table 2a: Multi-layer perceptron using Bayesian regulation back-propagation as training function: precision, recall and *F*-measure in the classification of the most common bigram's patterns.

Pattern	Bigrams	MWE	TP	FP	TN	FN	Prec.	Recall	F1
N-A	229	193	191	35	1	2	0.845	0.990	0.912
O-N	164	46	11	6	112	35	0.647	0.239	0.349
PN-PN	117	101	101	16	0	0	0.863	1.000	0.927
A-N	53	21	17	4	28	4	0.810	0.810	0.810
O-O	46	10	0	2	34	10	0.000	0.000	0.000
N-PN	34	16	11	5	13	5	0.688	0.688	0.688
N-N	31	20	16	7	4	4	0.696	0.800	0.744
A-PN	15	13	2	2	0	11	0.500	0.154	0.235
All Pat.	689	420	349	77	192	71	0.819	0.831	0.825

Table 2b: Multi-layer perceptron using Levenberg-Marquardt as training function: precision, recall and *F*-measure in the classification of the most common bigram's patterns.

Pattern	Bigrams	MWE	TP	FP	TN	FN	Prec.	Recall	F1
N-A	229	193	187	33	3	6	0.850	0.969	0.906
O-N	164	46	0	0	118	46	0.720	0.000	0.000
PN-PN	117	101	101	15	1	0	0.871	1.000	0.931
A-N	53	21	17	10	22	4	0.630	0.810	0.708
O-O	46	10	0	0	36	10	0.783	0.000	0.000
N-PN	34	16	2	7	11	14	0.222	0.125	0.160
N-N	31	20	18	8	3	2	0.692	0.900	0.783
A-PN	15	13	2	1	1	11	0.667	0.154	0.250
All Pat.	689	420	327	74	195	93	0.815	0.779	0.797

Table 2c: Multi-layer perceptron using scaled conjugate gradient as training function: precision, recall and *F*-measure in the classification of the most common bigram's patterns.

6 Evaluation of two different tools

Using the same excerpts of our corpus, we proceeded to the evaluation of two different tools for extracting MWEs from text: MWEtoolkit[1] (Ramisch, 2012) and Text-NSP[2] (Banerjee and Pedersen, 2003).

6.1 MWEtoolkit

Before using this tool, we POS-tagged the corpus using TreeTagger[3] (Schmid, 1994), with a Portuguese parameter file. Then we transformed the tagged corpus to the xml format used by MWEtoolkit using MWEtoolkit script treetagger2xml.

After generating the index, we defined the patterns file using the following bigrams patterns: N-N, N-ADJ, N-V, ADJ-ADJ, ADJ-N, ADJ-V, V-V, V-N and V-ADJ. There is not a PN tag for proper name in TreeTagger, so the proper names were treated as nouns (N). And we decided not to use the other grammatical classes (adverbs, pronouns, etc.), labeled as "O" in the previous section, because the only patterns that gave more than 10 MWEs with frequency higher than 3 using those classes were O-N and O-O, and we did not obtain good values for their classification using the multi-layer perceptron classification algorithms.

We used those patterns to generate all the bigrams and we obtained 28,738 candidates, with their frequencies and the frequencies of each word composing the bigram. Then we calculated five different association measures for each candidate: maximum likelihood estimator (mle), pointwise mutual information (pmi), Student's t-test (t), Dice's coefficient (dice), and log-likelihood (ll).

[1] http://mwetoolkit.sourceforge.net/PHITE.php?sitesig=MWE

[2] http://search.cpan.org/~tpederse/Text-NSP/

[3] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Then we created candidates files ordered by each of these five association measures and we ranked the *n* best candidates, *n* = 50, 100, 500, 1000 and 5000. Finally, we used MWEtoolkit's automatic evaluation script to evaluate each of these ranked candidates against our reference file.

The reference file was created with the 447 MWEs selected according to the method described in Section 4, i. e., all the bigrams that appear three or more times in our corpus and that we manually considered as a MWE. It is important to note that our reference file does not contain all the MWEs with two words in the corpus, since we generated more than 49,000 bigrams and we could not evaluate all of them by hand. Furthermore, the corpus is formed by newspaper texts, treating different subjects, thus it is more difficult to create a closed set of all possible two-word MWEs. Therefore, our evaluation is a comparison of how many of the most frequent two-word MWEs in our corpus are ranked as the *n* best candidates by each of the association measures.

Table 3 and Figure 1 show the result of our evaluation using the MWEtoolkit. Each number in the table represents how many of the MWEs in our reference list were found among the *n* best ranked candidates. For example, for the "ll" measure, among the 50 best ranked candidates 27 are MWEs that appear in our reference list.

	dice	**ll**	**mle**	**pmi**	**t**
50	0	27	5	0	11
100	12	55	10	0	31
500	34	152	49	1	105
1000	59	170	87	9	169
5000	161	187	179	94	186

Table 3: MWEtoolkit: number of MWEs among the first *n*-best candidates, ranked by five association measures.

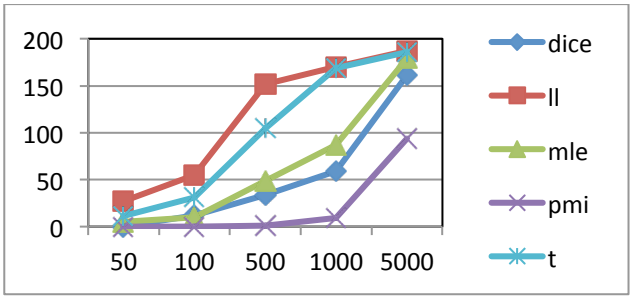

Figure 1: MWEtoolkit: five association measures performance comparison.

ll	**TP**	**Prec.**	**Recall**	**F1**
50	27	0.54	0.06	0.11
100	55	0.55	0.12	0.20
500	152	0.30	0.34	0.32
1000	170	0.17	0.38	0.23
5000	187	0.04	0.41	0.07

Table 4: MWEtoolkit: precision, recall and *F*-measure for the log-likelihood measure.

Analyzing the results, we notice that with log-likelihood measure we could find the highest number of MWEs present in our reference list, for all values of *n*. Since our reference list is formed by the most frequent MWEs in the corpus (frequency higher than three), this is an evidence of how suitable this measure is when the task is to find the most frequent two-word MWEs.

Table 4 presents the results of precision, recall and *F*-measure for the ll-measure for different values of *n* candidates. However, it should be kept in mind that precision and recall here are based on our reference list, which does not contain all the two-word MWEs in the corpus.

6.2 Text-NSP

Before applying this tool, the only pre-processing performed was to remove the XML tags. The next step was to define a stop words list file, since we are interested in finding MWEs following the patterns N-N, N-ADJ, N-V, like in Sections 4 and 5.

We ran the program using the script "count.pl", giving as parameter the stop words file and the corpus file, and 2 as n-gram value, meaning that we wanted to generate only bigrams.

The exit file is a list of all bigrams in the corpus, and each line contains a bigram, the frequency of the bigram, and the frequency of each of the two words forming the bigram.

Using the exit file and the script "statistics.pl" we generated the candidates' files ranked by four different association measures: Dice's coefficient (dice), log-likelihood (ll), pointwise mutual information (pmi) and Student's t-test (t). Maximum likelihood estimator is not implemented by Text-NSP. Then we transformed each of the candidates files to the XML format used by the MWEtoolkit and we used the MWEtoolkit scripts to create files with the *n* best candidates (*n* = 50, 100, 500, 1000 and 5000) and to evaluate each of the files against our reference file. Table 5 and Figure 2 show the results of those evaluations.

	dice	**ll**	**pmi**	**t**
50	7	31	0	23
100	7	64	0	39
500	8	241	1	180
1000	11	314	4	331
5000	73	382	15	406

Table 5: Text-NSP: number of MWEs among the first *n*-best candidates, ranked by four association measures.

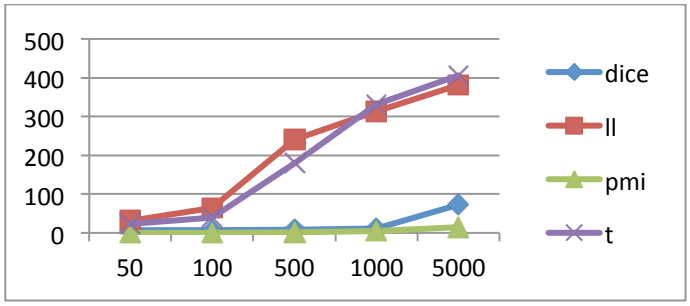

Figure 2: Text-NSP: four association measures performance comparison.

ll	**TP**	**Prec.**	**Recall**	**F1**
50	31	0.62	0.07	0.12
100	64	0.64	0.14	0.23
500	241	0.48	0.54	0.51
1000	314	0.31	0.70	0.43
5000	382	0.08	0.85	0.14

Table 6: Text-NSP: precision, recall and *F*-measure for the log-likelihood measure.

The results show that for values of *n* = 50,100, and 500 we obtained the best results using the log-likelihood measure and for *n* = 1000 and 5000, Student's t-test gave the best results.

Comparing with MWEtoolkit, we had better results with Text-NSP for the log-likelihood and the Student's t-test measures, and weaker results for the dice and pmi measures.

Table 6 shows the precision, recall and *F*-measure that we obtained for the log-likelihood measure. We had very good precision values using the Text-NSP with the log-likelihood measure. For example, from the 50 best ranked candidates by this measure, 31 were MWEs present in our reference list.

6.3 Comparison between MWEtoolkit and Text-NSP

Using the 500 best candidates generated by MWEtoolkit and Text-NSP, ranked by Student's t-test, we analyzed by hand those 500 candidates to decide which ones are Brazilian Portuguese MWEs. Table 7 shows the precision given by each of the tools for the first *n* candidates, *n* = 50, 100, 150…500.

Text-NSP showed higher precision than MWEtoolkit for all values of *n* candidates, especially for the smaller values of *n*. With MWEtoolkit, the precision was around 40%, while with Text-NSP it starts with 62% for the first best 50 candidates and decreases to 48% for the first best 500 candidates.

We can suppose that for an application interested in a small number of Brazilian Portuguese MWE candidates, Text-NSP would be a better choice, and as the number of candidates increases, the programs tend to have similar performance.

Checking the best ranked candidates generated by MWEtoolkit, we noted that it ranked well some bigrams formed by a noun + the preposition "a" (the/fem.), a pattern that is common in a Brazilian Portuguese corpus, but that usually does not form MWEs. This happened, despite not having any pattern that includes preposition in our patterns' list, because the POS-tagger used (TreeTagger) wrongly labelled those "a" prepositions as nouns. The same is true for the pronoun "seu/sua" (his/her), which was labelled as adjective. This can explain the difference in performance between the tools, when comparing the implementation of the same association measures.

As in the tests performed in Section 5, the most common patterns of MWE found by both programs were noun-adjective (e.g. Casa Branca, plano real, Estados Unidos) and proper name-proper name (e.g. Fernando Henrique, Ayrton Senna, Paulo Maluf).

***n* first cand.**	**MWEtoolkit**	**Text-NSP**
50	0.34	0.62
100	0.47	0.57
150	0.43	0.55
200	0.41	0.53
250	0.40	0.54
300	0.41	0.53
350	0.37	0.53
400	0.41	0.50
450	0.42	0.52
500	0.40	0.48

Table 7: MWEtoolkit and Text-NSP precision for the first *n* best candidates, using Student's t-test association measure.

7 Conclusions and future work

We obtained very similar results using different algorithms for the classification of MWEs, with bagging, decision trees and multi-layer perceptron having a slightly better performance.

Using multi-layer perceptron with three different training functions, we identified the bigram's patterns that are better classified as MWE. With the function Levenberg-Marquardt we had better results in classifying the pattern noun-adjective (the most common in our corpus) and the function Scaled Conjugate Gradient was the most successful in classifying MWEs following the pattern proper name-proper name.

The comparison between two programs for automatic extraction of MWEs showed that Text-NSP had a better precision than MWEtoolkit, especially for smaller number of candidates. As the number of candidates increases, the difference in performance between the two programs decreases.

It is important to note that MWEtoolkit is more complete, in the sense it implements more statistical measures, makes the comparison between the output candidates file and a reference list file and generates a list of candidates having more complete information, including all the statistical measures of each candidate in the same file, and in a XML format more easily consumable by other programs.

As a future work, we intend to perform a similar comparison of tools and classification algorithms for the extraction of Brazilian Portuguese MWEs, not limiting our candidates to bigrams, but studying n-grams in general, also allowing noncontiguous n-grams.

References

Agarwal, A., Ray, B., Choudhury, M., Sarkar, S., Basu, A.: Automatic Extraction of Multiword Expressions in Bengali: An Approach for Miserly Resource Scenarios. In: *Proceedings of ICON 2004*, pp. 165-174. Macmillan, Basingstoke (2004).

Aha, D. and Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning*. 6:37-66.

Antunes, S. and Mendes, A. MWE in Portuguese - Proposal for a Typology for Annotation in Running Text. *Proceedings of the 9th Workshop on Multiword Expressions (MWE 2013)*, pp. 87–92, Atlanta, Georgia, 13-14 June 2013.

Banerjee, S and Pedersen, T. (2003). The Design, Implementation, and Use of the Ngram Statistic Package. *In Proceedings of th Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 370-381, Mexico City.

Breiman, L. (2001). Random Forests. *Machine Learning*. 45(1):5-32.

Breiman , L. (1996). Bagging predictors. *Machine Learning*. 24(2):123-140.

CETENFolha (Corpus de Extractos de Textos Electrónicos NILC/Folha de S. Paulo) (2008). Linguateca – Portugal – www.linguateca.pt/ACDC/

Chang, Chih-Chung and Lin, Chih-Jen (2001). LIBSVM - A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Choueka, Y. (1988). Looking for needles in a haystack or locating interesting collocational expressions in large textual databases. *In RIAO'88*, pp. 609–624.

Church, K. W. and Hanks, P (1990). Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, 16(1):22–29.

Cooper, G. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*. 9(4):309-347.

Dias, G. H. and Lopes, J.G.P. (2005). Extracção automática de unidades polilexicais para o português. In: *A Língua Portuguesa no Computador*. São Paulo: Ed. Mercado de Letras.

Freund, Y. and Schapire, R. E (1996). Experiments with a new boosting algorithm. *In: Thirteenth International Conference on Machine Learning, San Francisco*, pp. 148-156.

Hendrickx, I., Mendes, A. and Antunes, S. (2010). Proposal for Multi-Word Expression Annotation in Running Text Portuguese.

Hurskainen, A. (2008). Multiword Expressions and Machine Translation. *Technical Reports in Language Technology Report No 1, 2008 (*http://www.njas.helsinki.fi/salama).

Kinoshita, J., Nascimento Salvador, L.D. and Dantas de Menezes, C., E. (2006). CoGrOO: a Brazilian-Portuguese Grammar Checker based on the CETENFOLHA Corpus. In proceedings of LREC 2006. http://www.pcs.usp.br/~cogroo/papers/Artigo_LREC_2006.pdf

Piao, S., Rayson, P., Archer, D., Wilson, A., and McEnery, T. (2003). Extracting Multiword Expressions with a Semantic Tagger. *In: Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, at *ACL 2003, 41st Annual Meeting of the Association for Computational Linguistics*, 2003-07-12, Sapporo, Japan.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.

Ramisch, C. (2012). A generic and open framework for MWE treatment – from acquisition to applications - Ph.D. Thesis, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil.

Ramisch, C., Schreiner, P., Idiart, M. and Villavicencio, A. (2008). An Evaluation of Methods for the Extraction of Multiword Expressions, *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, Marrakech, Morocco, June, 2008.

Sag, I., Baldwin, T., Bond, F., Copestake, A. and Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. *In Proc. of the 3rd CICLing (CICLing-2002)*, volume 2276/2010 de *LNCS*, pp. 1–15, Mexico City, Mexico.

Schmid, Helmut (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Silva, J., and Lopes, G. (1999). A local Maxima Method and a Fair Dispersion Normalization for Extracting Multiword Units. In *6th Meeting on the Mathematics of Language*, pp. 369-381.

Smadja, F. A. (1996). Translating Collocations for Bilingual Lexicons: A Statistical Approach. *Association for Computational Linguistics*, 22 (1):1-38.

Smadja, F. A. (1993). Retrieving collocations from text: Xtract. *Computational Linguistics.*, 19(1):143–177.

Watrin, Patrick and François, Tomas (2011). An N-gram frequency database reference to handle MWE extraction in NLP applications. *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011)*, pp. 83–91.