# Extraction of Definitions in Portuguese: An Imbalanced Data Set Problem

Rosa Del Gaudio and António Branco

University of Lisbon
Faculdade de Ciências, Departamento de Informática
NLX - Natural Language and Speech Group
Campo Grande, 1749-016 Lisbon, Portugal
rosa@di.fc.ul.pt antonio.branco@di.fc.ul.pt

**Abstract.** Definition extraction is an important task in NLP and IR fields in the context of e.g. question answering, ontology learning, dictionary and glossary construction. When addressed with learning algorithms, it turns out to be a challenging task due to the structure of the data set, the reason being that the definition-bearing sentences are much fewer than the sentences that are non definitions. In this paper, we present results from experiments that seek to obtain optimal solutions for this problem by using a corpus written in the Portuguese language. Our results show an improvement of 29 points regarding AUC metric and more than 60 points when considering the F-measure.

**Key words:** automatic definition extraction, machine learning, imbalanced data set.

## 1 Introduction

Definition Extraction is an important task in Natural Language Processing (NLP) and Information Retrieval (IR), in the context of e.g. Question Answering (QA), ontology learning, dictionary and glossary construction, etc.

The interest on definitions dates back to Antiquity. According to Aristotle, the formal structure of a definition should resemble an equation with the *definiendum* (what is to be defined) on the left hand side and the *definiens* (the part which is doing the defining) on the right hand side. The *definiens* should consist of two parts: the *genus* (the nearest superior concept) and the *differentiae specificae* (the distinguishing characteristics). In this way, definitions would adequately capture the concept to be defined.

A thorough study[1] on dictionary entries and definitions automatically extracted presents a description of the linguistic structure of definition sentences, identifying 16 different types of definitions. Nevertheless, in fields such as QA most of the research is focused on the extraction of a definition in a sentence composed by a subject, a copular verb and a predicative phrase.

In this field, the particular type of question, termed as definition question or "what" question, presents characteristics that differentiate it from other questions. All other types of questions, introduced by "who", "when", etc., give some clues on the type of answer which is supposed to be obtained. For example, which semantic type of named entity would provide a better answer to the question. In the case of definition questions, the space of answers is open and this implies that this class of questions needs specific techniques to be dealt with. In particular, when learning algorithms are used, this broadness gives rise to an imbalanced data set, which, depending on the corpus and the techniques used, may present different degrees of imbalance. For example, using a corpus consisting of encyclopedic text and web documents, [2] report that only 18% of the sentences were definitions. On the other hand, using only encyclopedic documents, [3] had a balanced corpus where the definition-bearing sentences represent 59% of the whole corpus.

Similarly to other works in this field, in this paper a definition is considered to be a sentence containing an expression (the *definiendum*) and its definition (the *definiens*), connected by the verb "to be", as in the example "FTP is a protocol that allows the transfer of archives from a place to another through the Internet." The corpus used in the work reported in this paper is composed mostly of tutorials and scientific papers in the Information Technology field, where the definition-bearing sentences were manually annotated and represent the 9% of all sentences.

The definition extraction problem can thus be represented as a binary classification task, where for each sentence in the corpus it is possible to assign the correct class: "definition" or "no definition".

In this paper, we present results obtained with the application of several sampling approaches to our imbalanced data set in order to build more effective classifiers for the definition extraction problem. We try to keep our model as general as possible in order for it to be applicable in different domains. For this reason, we only use information on part of speech (POS) as a feature. This makes the present approach viable for all those languages that are not equipped with rich lexical resources as learning data or in a situation where the domain is too specific to benefit from such resources.

Our task handles several aspects that are common to different machine learning tasks in NLP applications: small amounts of data, inherent ambiguity, noisy data (human annotators make mistakes), imbalanced class distribution, this last aspect being the main issue addressed in this paper.

The rest of the paper is organized in the following way: Section 2 reports on work in the area of definition extraction. Section 3 gives a brief description of the corpus used. Sections 4 and 5 report respectively on the algorithms and on the sampling techniques used. Section 5 is devoted to discussing possible evaluation metrics to be used in the case of imbalanced data sets. Finally, Section 6 presents the results and in Section 7 conclusions are drawn and future directions of our investigation are put forward.

## 2 Related Work

Previous research on automatic extraction of definitions explored the lexico-syntactic patterns in texts taking into consideration mainly POS or lemmas as main linguistic features. Since the 90's, several authors have proposed methods to identify lexico-syntactic patterns [4, 5].

DEFINDER [6] is an automatic definition extraction system that combines simple cue-phrases and structural indicators introducing the definitions and the defined term. It was developed with a well-structured medical corpus, where 60% of the definitions are introduced by a set of limited text markers. The nature of the corpus used can explain the high performance obtained by this system (87% precision and 75% recall).

Malaise and colleagues [7] developed a system for the extraction of definitory expressions containing hyperonym and synonym relations from French corpora, using corpora from different domains for training and testing. These authors used lexical-syntactic markers and patterns to detect at the same time definitions and relations. For the two different relations (hyponym and synonym), they obtained, respectively, 4% and 36% of recall, and 61% and 66% of precision.

More recently, machine learning techniques were combined with pattern recognition in order to improve the general results. In particular, [3] used a maximum entropy classifier to extract definitions in order to distinguish actual definitions from other sentences. They propose several attributes to classify definition sentences, namely text properties (such as n-gram and bag-of-words), sentence position, syntactic properties and named entity classes. The corpus used was derived from medical pages of the Dutch Wikipedia, from which they extracted sentences based on syntactic features, ending up with 2,299 senteces of which 1,366 are actual definitions. This gives an initial accuracy of 59%, that was improved with machine learning algorithms up to 92.21%

In [8], a system to extract definitions from off-line documents is presented. They experimented with three different algorithms, namely Naïve Bayes, Decision Tree and Support Vector Machine (SVM), obtaining the best score with SVM with a F-measure of 0.83 with a balanced data set.

Westerhout and Monachesi [9] combine syntactic patterns with a Naïve Bayes classification algorithm with the aim of extracting glossaries from tutorial documents in Dutch. They used several properties and several combinations of them, obtaining an improvement in precision of 51.9%, but a decline in the recall of 19.1% in comparison with the syntactic pattern system developed previously by these authors, using the same corpus.

In spite of the increasing attention the imbalanced data set issue has attracted in the machine learning community, shown by two different workshops held in 2000 [1] and 2003 [2], little attention has beeen payed to the data set structure of the definition detection task when addressed with machine learning techniques.

[1] Japkowicz, editor, Proceedings of the AAAI2000 Workshop on Learning from Imbalanced Data Sets, AAAI Tech Report WS-00-05.

[2] N. V. Chawla, N. Japkowicz, and A. Kolcz, editors, Proceedings of the ICML2003 Workshop on Learning from Imbalanced Data Sets. 2003.

Recently, some authors have started to look at this problem of imbalanced data sets in the context of definition extraction. In particular, [10] down-sampled their corpus using different ratios (1:1, 1:5, 1:10) in order to seek for best results. The corpus they used presented an original ratio of non-definitions to definitions of about 19. Although they obtained some improvement in terms of the F-measure, in particular with the ratio 1 to 5, they couldn't improve results obtained with a rule based grammar previously developed using the same corpus. These authors also investigated the use of Balanced Random Forest algorithm in order to deal with this imbalance, succeeding in outperforming the rule based grammar previously developed by 5 percentage points [11].

## 3 Data Set Description

The corpus used for the experiments was collected in the context of the LT4eL project [12]. It was used to develop different tools, such a keyword extractor, a glossary candidate detector and an ontology, in order to support e-learning activities[13, 14]. The corpus is composed of several tutorials and scientific papers in the field of Information Technology and has a size of 274,000 tokens. It was automatically annotated with morpho-syntactic information using the LX-Suite [15, 16].

Definition-bearing sentences were manually annotated. In each sentence, the term defined, the definition and the connection verb were annotated using a different XML tag. As the focus of this work are definitions conveyed by the verb "to be", a simple grammar was developed in order to extract all the sentences with this verb as main verb. A sub-corpus was obtained, composed by 1,360 sentences, 121 of which are definitions, with a ratio of about 10:1.

As for feature selection, works in similar areas tend to use different types of properties (text, document, syntactic, etc.). Examples of text properties are bag-of-words and n-grams [17] either of part-of-speech or of base forms. Regarding document properties, the position of the definition inside the document is often used as a property [18], as well as the presence of determiners in the *definiens* and in the *definiendum* [3]. Other relevant properties can be the presence of named entities [3] or data from an external source such as encyclopedic data, wordnets, etc. [19].

Most of these features are strictly related to the corpus used, rendering generalizations for other corpus very difficult. For example, in [3] the use of the position of a definition-bearing sentence as a feature is based on the observation that definitions tend to occur at the beginning of a document, but the corpus used in their work was based on wikipedia articles, and this is just a characteristic of this public encyclopedia. Similar problems arise when information other than part of speech is used as a feature. In this case, the results obtained are typically hard to generalize to other text domains.

For all these reasons, in the present work, instances were represented as n-grams of POS. Different configurations were tested with $n$ ranging from 1 to 4. From all POS n-grams extracted from the set of 1,360 definitions, the 100

most frequent were used as features. Each sentence was represented as an array where cells record the number of occurrences of these n-grams. In this paper, due to the limited number of pages available, only results obtained with the best representation are shown, that is with bi-grams.

## 4 Machine Learning Algorithms

When selecting learning algorithms, two different considerations were taken into account. First, we want to use those algorithms that in literature represent the state of the art for definition extraction and also for imbalanced data sets problems. Second, we want to cover different classes of algorithms, having at least a representative algorithm for different classes. In this way, results obtained with different sampling techniques may be generalized to a larger range of algorithms. Five different algorithms were selected: Naïve Bayes, C4.5, Random Forest, k-NN, SVM.

Naïve Bayes is a simple probabilistic classifier that is very popular in Natural Language applications. In spite of its simplicity, it permits to obtain results quite similar to those obtained with more complex algorithms.

C4.5 and Random Forest are two decision tree algorithms. The first is a relatively simple algorithm that splits the data into smaller subsets using the information gain in order to chose the attribute for splitting the data. The second is a classifier consisting of a collection of decision trees. For each tree, a random sample of the data set is selected (the remaining is used for error estimation), and for each node of the tree, the decision at that node is based on a restricted number of variables.

The k-NN algorithm is a type of instance-based learning, also called lazy learning because, unlike the algorithms above, the training phase of the algorithm consists only in storing the feature vectors and class labels of the training samples and all computation is deferred for the classification phase. In this phase, the algorithm computes the distance between the target sample and $n$ samples in the data set, assining the most frequent class. Two different K nearest neighbors classifiers were constructed, with k equal to 1 and to 3.

SVM is a classifier that tries to find an optimal hyperplane that correctly classifies data points as much as possible and separates the point of two classes as far as possible.

All the classifiers were implemented using the Weka workbench [20].

## 5 Sampling Techniques

In many real-world classification applications, most of the examples are from one of the classes, while the minority class is the interesting one. As most of the learning algorithms are designed to maximize accuracy, the imbalance in the class distribution leads to a poor performance of these algorithms. The issue is therefore how to improve the classification of the minority class examples.

A common solution is to sample the data, either randomly or intelligently, to obtain an altered class distribution.

Random over-sampling consists of random replication of minority class examples, while in random down-sampling majority class examples are randomly discarded until the desired amount is reached. These two very simple methods are often criticized due to their drawbacks. Several authors pointed out that the problem with under-sampling is that this method can discard potentially useful data that could be important for the induction process. On the other hand, random over-sampling can increase the likelihood of overfitting, since it makes exact copies of the minority class examples.

When speaking about negative and positive examples in a data set, it is important to have in mind that not all the examples have the same value. There are examples that are more prototypical than others and represent better the class to which they belong, while others are too similar to be useful, and others are just noise.

Building on these considerations, several methods were proposed in order to retain safe examples in the re-balanced data set. We consider here two of such methods, namely the Condensed Nearest Neighbour Rule and Tomek Links algorithm.

Condensed Nearest Neighbor Rule (CNN) [21] finds a consistent subset of examples in order to eliminate the examples from the majority class that are distant from the decision border, since these examples might be considered less relevant for learning. The CNN is sensitive to noise and noisy examples are likely to be misclassified as many of them will be added to the training set.

Tomek Links [22] removes both noise and borderline examples. Tomek Links are pairs of instances of different classes that have each other as their nearest neighbors. As an under-sampling method, only examples belonging to the majority class are eliminated. The major drawback of Tomek Links under-sampling is that this method can discard potentially useful data that could be important for the induction process. This method has an higher order computational complexity and will run slower than other algorithms.

While the previous methods are intelligent down sampling techniques, SMOTE is an over-sampling method that produces new synthetic minority class examples. SMOTE [23] forms new minority class examples by interpolating between several minority class examples that lie together in "feature space" rather than "data space". For each minority class example, this algorithm introduces synthetic examples along the line segments joining any/all of the k minority class nearest neighbors (in this work k is equal to 3).

## 6 Evaluation Issues

Using the confusion matrix in Table 1 as a starting point, we discuss the possible metrics for the evaluation of the classifiers investigated in this work. One of the most used metrics is the Error Rate, defined as 1.0-(TP+TN)/(TP-FP+FN+TN). However, using this metric implies that the class distribution is

known and fixed, an assumption that does not hold in real world applications as the one proposed here. Moreover, Error Rate is biased to favor the majority class, making it a bad choice when evaluating the effects of class distribution. Another aspect against the use of Error Rate is that it considers different classification errors as equally important, and in domains such medical diagnosis, the error of diagnosing a sick patience as healthy is a fatal error while the contrary is considered a much less serious error. In general, any performance metric, such as accuracy and Error Rate, that uses values from both columns will be sensitive to class imbalance.

| | Positive Prediction | Negative Prediction |
|---|---|---|
| Positive Class | True Positive (TP) | False Negative (FN) |
| Negative Class | False Positive (FP) | True Negative (TN) |

**Table 1.** Confusion Matrix for a binary classification problem

Starting from the confusion matrix it is possible to derive metrics that are not sensitive to the skew of the data. In particular, four metrics are proposed in [24]:

- False Negative Rate: FN/(TP+FN) - the percentage of positive examples misclassified as belonging to the negative class
- False Positive Rate: FP/(FP+TN) - the percentage of negative examples misclassified as belonging to the positive class
- True Negative Rate: TN/(FP+TN) - the percentage of negative examples correctly classified as belonging to the negative class
- False Positive Rate: TP/(TP+FN) - the percentage of positive examples correctly classified as belonging to the positive class

A good classifier should try to minimize FN and FP rates, and maximize TN and TP rates.

Unfortunately, there is a tradeoff between these two metrics, and in order to analyze this relationship ROC graphs are used. ROC graphs are two-dimensional graphs where the TP rate is plotted on the Y axis and the FP rate is plotted on the X axis. ROC graphs are consistent for a given problem even if the distribution of positive and negative instances is highly skewed.

In order to compare classifiers, it is possible to reduce a ROC curve to a scalar value representing the performance of the classifier. The area Under the ROC (AUC) is a portion of the area of the unit square. Its value will always be between 0 and 1. However, because random guessing produces the diagonal ine between (0,0) and (1,1), which has an area of 0.5, no realistic classier should have an AUC less than 0.5.

In this work, we will use the AUC measure in order to assess the performance of classifiers. Furthermore, for each classifier, we present also the F-measure[3] in order to compare our results with the results of previous works in this area.

## 7 Results and Discussion

In this section, we present the results for the different learning algorithms used, namely Naïve Bayes, C4.5, Random Forest, k-NN and SVM. For each classifier, results regarding the different sampling techniques discussed above in Section 5, that is, random over and down sampling, SMOTE, CNN and Tomek Links are shown. We also present results obtained using the original data set, which is the data set with the original imbalance. This result represents our base line against which results obtained with sampled data sets are to be compared with. Values in bold represent the best score for each classifier.

Since the data set size does not allow us to split the corpus into two samples, a training set and a test set, 10-fold cross validation was used.

Tables 2 and 3 display the performance of the two classifiers using k-NN algorithm. In particular, Table 2 reports on the results of the most basic implementation of k-NN, that is with k equal to 1 (1-NN). In this case, a test example is simply assigned to the class of its nearest neighbor. Table 3 displays results obtained by a classifier using k-NN algorithm with k equal to 3 (3-NN).

Regarding the results in Table 2, it is possible to notice that, for the AUC metric, only the SMOTE sampling technique is able to significantly improve the base line, obtaining a score of 0.66 with an improvement of 10 points. If we focus on the F-measure, there is a substantial improvement with the different techniques, namely SMOTE and random down-sampling. As for AUC metric, also when considering the F-measure, the SMOTE presents the best score, namely 0.63 with an improvement on base line of 42 points.

Regarding results in Table 3, there are 4 sampling techniques that outperform the base line for F-measure: SMOTE (with the best score), followed by CNN, Tomek Links and random down-sampling. As to the AUC metric, the best performance is achieved by SMOTE and Tomek Links, with an improvement of 13 and 9 points respectively in comparison with the base line. Although the base lines for the classifiers above are very similar, they differ in the way they respond to the sampling techniques. In particular, the 3-NN algorithm seems to take more advantage from the use of sampling, since it obtains better results in all the experiments.

Tables 4 and 5 show the performance of the two classifiers based on decision tree algorithms, namely the C4.5 and Random Forest. The results displayed in Table 4 refer to the best setting for the C4.5 classifier, where the tree was pruned using the C4.5 standard pruning procedure with no Laplace correction. Regarding Table 5, the classifier was built using 10 different trees.

Similarly to previous classifiers, SMOTE sampling method presents the best results in terms of AUC for both classifiers, with a rise of 23 and 29 points for

[3] $F - measure = \frac{2 * Precision * Recall}{(Precision + Recall)}$

| 1-NN | | |
|---|---|---|
| Sampling | F-m | AUC |
| Original | 0.19 | 0.56 |
| Downsampling | 0.62 | 0.57 |
| Oversampling | 0.36 | 0.55 |
| SMOTE | **0.63** | **0.66** |
| CNN | 0.23 | 0.52 |
| Tomek | 0.57 | 0.59 |

**Table 2.** Results obtained for the classifier using **k-NN** algorithm with **k=1**

| 3-NN | | |
|---|---|---|
| Sampling | F-m | AUC |
| Original | 0.17 | 0.57 |
| Downsampling | 0.62 | 0.59 |
| Oversampling | 0.51 | 0.58 |
| SMOTE | **0.66** | **0.70** |
| CNN | 0.65 | 0.61 |
| Tomek | 0.64 | 0.66 |

**Table 3.** Results obtained for the classifier using **k-NN** algorithm with **k=3**

C4.5 and Random Forest respectively. The same observation holds for the F-measure, with an improvement of 60 and 63 points respectively. For this metric, good results are also achieved by Tomek Links and CNN.

| C4.5 | | |
|---|---|---|
| Sampling | F-m | AUC |
| Original | 0.17 | 0.65 |
| Downsampling | 0.58 | 0.59 |
| Oversampling | 0.37 | 0.67 |
| SMOTE | **0.77** | **0.87** |
| CNN | 0.62 | 0.61 |
| Tomek | 0.63 | 0.60 |

**Table 4.** Results obtained for the classifier using **C4.5** algorithm

| Random Forest | | |
|---|---|---|
| Sampling | F-m | AUC |
| Original | 0.13 | 0.65 |
| Downsampling | 0.57 | 0.65 |
| Oversampling | 0.21 | 0.64 |
| SMOTE | **0.75** | **0.94** |
| CNN | 0.59 | 0.66 |
| Tomek | 0.65 | 0.59 |

**Table 5.** Results obtained for the classifier using **Random Forest** algorithm

Table 6 displays results obtained with a SVM classifier using a sigmoid kernel. The AUC base line for this classifier is very low, with a value below 0.5. Using sampling techniques the performance of this classifier is comparable to the 1-NN, reaching an AUC of 0.68 with random down-sampling. It is interesting to observe that although SVM is a complex algorithm, it achieves a performance similar to the simplest algorithm used in this work, namely 1-NN. Furthermore it is the only classifier where the SMOTE does not show the best result, considering either AUC or F-measure. Only the classifier based on SVM presents the best result when coped with the random down-sample method.

The results in Table 7 refer to a Naïve Bayes classifier using normal distribution. The base line for this classifier is higher than for the other classifiers in terms of both metrics taken in consideration. Nevertheless, the improvements achieved with the use of sampling do reach the performance of other classifiers, namely C4.5 and Random Forest.

In general, the SMOTE sampling technique shows the best results in terms of AUC, followed by Tomek Links and random over-sampling. The best score for SMOTE is achieved by Random Forest with 0.94 followed by C4.5 with 0.87.

| | SVM | |
|---|---|---|
| Sampling | F-m | AUC |
| Original | 0.12 | 0.48 |
| Downsampling | **0.67** | **0.68** |
| Oversampling | 0.61 | 0.59 |
| SMOTE | 0.60 | 0.60 |
| CNN | 0.59 | 0.57 |
| Tomek | 0.64 | 0.49 |

**Table 6.** Results obtained for the classifier using **SVM** algorithm

| | Naïve Bayes | |
|---|---|---|
| Sampling | F-m | AUC |
| Original | 0.24 | 0.66 |
| Downsampling | 0.62 | 0.62 |
| Oversampling | 0.67 | 0.68 |
| SMOTE | **0.72** | **0.76** |
| CNN | 0.64 | 0.63 |
| Tomek | 0.69 | 0.72 |

**Table 7.** Results obtained for the classifier using **Naïve Bayes**

These results are comparable with those reported in the literature on imbalanced data sets in general. In a comprehensive study on the behavior of several methods for balancing training data, using the 11 UCI data set [4], Batista and colleagues [24] show that in most of the cases and with several data sets in different domains SMOTE and random over-sampling are the most effective methods. In general, they obtain a rise in the AUC metric of few percentage points (1 to 4), when the base line was already high (more than 0.65); when the base line was under this value the improvement was comparable to the one obtained in our work. In particular for the flag data set, they obtained an improvement of 34 percentage points.

Focusing on Natural Language applications, [25] apply these methods to sentence boundary detection in speech, showing that SMOTE and random down-sampling get the best results with an AUC of 0.89 (the base line being 0.80). However, they did not experiment intelligent down-sampling methods such as CNN or Tomek Links. Batista [26], in a case study on automated annotation of keywords , gets the best results in terms of AUC with an improvement of 4 percentage points on the original data set using a combination of SMOTE with Tomek Links, followed by simple SMOTE.

In our case the improvement regarding the original data set is between 10 and 29 points, demonstrating how these methods can be effective in this application.

Regarding the comparison with other work in definition extraction, the improvement obtained on the F-measure, with the best result of 0.77 with C4.5 classifier, outperforms most of the systems presented in Section 2, confirming the importance of sampling techniques in supporting definition extraction tasks. For instance, [9] reports on a F-measure of 0.73, obtained with a combination of syntactic rules and a Naïve Bayes classifiers for the Dutch language, in turn, [10], with a similar approach, but for the Polish, obtain a F-measure of 0.35.

Furthermore, in all these works a combination of features is used in order to reach better results, while in this paper we only use bi-grams of POS as feature.

To conclude, our results outperform those systems that represent the state of the art in the area, such as DEFINDER, which shows a F-mesure of 0.80.

[4] http://archive.ics.uci.edu/ml/

## 8 Conclusions and Future Work

In this work, we presented a study on the better way to deal with imbalanced data sets in the context of definition extraction. We reported results for five classifiers and five different sampling techniques. Our results are comparable to the results obtained in previous work in the area, confirming the SMOTE sampling method as one of the most effective in dealing with imbalanced data sets.

Furthermore, this work empirically demonstrates the effectiveness of sampling methods in the definition extraction field. This finding is supported by the magnitude of the improvement obtained in comparison with the original data sets for both the metrics used. In particular, our results show an improvement of 29 points regarding the AUC metric and more than 60 points when considering the F-measure.

In future work we are planning to experiment with more sampling techniques as well combining them in different ways. Additionally, we want to use different data sets in different languages in order to validate our findings.

## References

1. Barnbrook, G.: Defining Language: a local grammar of definition sentences. John Benjamins Publishing Company (2002)
2. Tjong, E., Sang, K., Bouma, G., de Rijke, M.: Developing offline strategies for answering medical questions. In: Proceedings of the AAAI-05 workshop on Question Answering in restricted domains. (2005) 41–45
3. Fahmi, I., Bouma, G.: Learning to identify definitions using syntactic feature. In Basili, R., Moschitti, A., eds.: Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications, Trento, Italy (2006)
4. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1992) 539–545
5. Person, J.: The expression of definitions in specialised text: a corpus-based analysis. In Gellerstam, M., Jaborg, J., Malgren, S.G., Noren, K., Rogstrom, L., Papmehl, C., eds.: 7th Internation Congress on Lexicography (EURALEX 96), Goteborg, Sweden (1996) 817–824
6. Klavans, J., Muresan, S.: Evaluation of the DEFINDER system for fully automatic glossary construction. In: Proceedings of the American Medical Informatics Association Symposium (AMIA 2001). (2001)
7. Malais, V., Zweigenbaum, P., Bachimont, B.: Detecting semantic relations between terms in definitions. In: the 3rd edition of CompuTerm Workshop (CompuTerm 2004) at Coling 2004. (2004) 55–62
8. Chang, X., Zheng, Q.: Offline definition extraction using machine learning for knowledge-oriented question answering. In Huang, D.S., Heutte, L., Loog, M., eds.: ICIC (3). Volume 2 of Communications in Computer and Information Science., Springer (2007) 1286–1294
9. Westerhout, E., Monachesi, P.: Extraction of Dutch definitory contexts for elearning purposes. In: CLIN proceedings 2007. (2007)

10. Przepiorkowski, A., Marcinczuk, M., Degorski, L.: Noisy and imbalanced data: Machine learning or manual grammars? In: Text, Speech and Dialogue: 9th International Conference, TSD 2008, Brno, Czech Republic, Lecture Notes in Artificial Intelligence, Berlin, Springer-Verlag (2008)
11. Kobylinski, L., Przepiorkowski, A.: Definition extraction with balanced random forests. In Ranta, A., ed.: GoTAL 2008, Gothenburg, Springer-Verlag Berlin Heidelberg (2008) 237–247
12. Monachesi, P., Lemnitzer, L., Simov, K.: Language technology for elearning. In: Proceedings of EC-TEL, Springer LNCS (2006)
13. Avelãs, M., Branco, A., Gaudio, R.D., Martins, P.: Supporting e-learning with language technology for portuguese. In: Proceedings of the International Conference on the Computational Processing of Portuguese (PROPOR2008), Springer (2008)
14. Gaudio, R.D., Branco, A.: Learning to identify definitions using syntactic feature. In: Progress in Artificial Intelligence, 13th Portuguese Conference on Aritficial Intelligence, EPIA 2007, Workshops: GAIW, AIASTS, ALEA, AMITA, BAOSW, BI, CMBSB, IROBOT, MASTA, STCS, and TEMA, Guimarẽs, Portugal, Springer Berlin (2007) 659–670
15. Silva, J.R.: Shallow processing of Portuguese: From sentence chunking to nominal lemmatization. Master's thesis, Universidade de Lisboa, Faculdade de Ciências (2007)
16. Branco, A., Silva, J.R.: Lx-suite: Shallow processing tools for portuguese. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06). (2006) 179–183
17. Miliaraki, S., Androutsopoulos, I.: Learning to identify single-snippet answer to definition questions. In: Proceeding of the 20th International Conference on Computational Linguistic (COLING 2004), Geneva, Switzerland (2004) 1360–1366
18. Joho, H., Sanderson, M.: Retrieving descriptive phrases from large amounts of free text. In: Proceeding of hte 9th international conference on Information and knowledge management. (2000) 180–186
19. Saggion, H.: Identifying definitions in text collections for question answering. In: LREC 2004. (2004)
20. Witten, I.H., Frank, E.: Data Mining: Pratical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann (2005)
21. Hart, P.E.: The condensed nearest neighbor rule (corresp.). Information Theory, IEEE Transactions on **14**(3) (1968) 515–516
22. Tomek, I.: Two modifications of cnn. Systems, Man and Cybernetics, IEEE Transactions on **6**(11) (1976) 769–772
23. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16** (2002) 321–357
24. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. **6**(1) (2004) 20–29
25. Liu, Y., Chawla, N.V., Harper, M.P., Shriberg, E., Stolcke, A.: A study in machine learning from imbalanced data for sentence boundary detection in speech. Computer Speech & Language **20**(4) (2006) 468–494
26. Batista, G.E.A.P.A., Bazzan, A.L.C., Monard, M.C.: Balancing training data for automated annotation of keywords: a case study (2003)