Language Independent System for Definition Extraction: First Results Using Learning Algorithms

Rosa Del Gaudio 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

In this paper we report on the performance of different learning algorithms and different sampling technique applied to a definition extraction task, using data sets in different language. We compare our results with those obtained by handcrafted rules to extract definitions. When Definition Extraction is handled with machine learning algorithms, two different issues arise. On the one hand, in most cases the data set used to extract definitions is unbalanced, and this means that it is necessary to deal with this characteristic with specific techniques. On the other hand it is possible to use the same methods to extract definitions from documents in different corpus, making the classifier language independent.

Keywords

machine learning, imbalanced data set, language independent, definition extraction

1 Introduction

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.

In Hebenstreit [9], two more types of definition are pointed out. Firstly, the definition by enumeration of the concept species on the same level of abstraction (extensional definition), e.g. a chess piece is a king, a queen, a bishop, a knight, a rook or a pawn. Secondly, the definition by enumeration of the parts of the concept (partitive definition), e.g. the solar system is made of the planets Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune and Pluto. Barnbrook [2] identifies 16 different types of definitions analysing dictionary entries. In spite of the richness of this classification, in automatic definition extraction application only the simplest type is taken in consideration, that is a sentence composed by a subject, a copular verb and a predicative phrase. In this paper a definition is a sentence containing an expression (the *definiendum*) and its definition (the *definiens*) connected by the verb "to be".

Two different approaches are possible when dealing with automatic definition extraction. The first one consists in building a system of rules, based on lexical and syntactic clues. The second one is to consider the task as a classification problem, where for each sentence in the corpus it is possible to assign the correct class. The problem of the first approach is that it is language dependent, and in case of a large use of lexical clues, the performance on different corpus get worst. In the case of classification approach one of the main issue to be dealt with is the sparseness of definitions in a corpus. It is a matter of fact that the number of definition bearing sentences is much lesser than the number of sentences that are not definitions. This configuration gives rise to an imbalanced data set, which may present different degrees of imbalance, depending on the corpus used. For corpus composed mostly by encyclopedic documents it is likely to get a balanced data set. For example [8] used a balanced corpus where the definition-bearing sentences represent 59% of the whole corpus, while [24] using a corpus consisting of encyclopedic text and web documents reports that only 18% of the sentences were definitions.

In this work we deal with the problem of imbalanced data sets in definition extraction tasks in a language independent way. We show not only that sampling techniques can improve the performance of classifiers but also that this improvement is language independent. Other researches using learning algorithms relay strongly on lexical and syntactic components as features to describe the data set. These kinds of features are not only language dependent but also domain dependent, and as we want our classifier to be as general as possible we select the most basic features, that is n-grams of part of speech (POS). 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, and moves away from previous works that use features such as words, word lemmas, position of the sentence in the document he document, etc. In this paper we apply the same techniques we applied to a Portuguese Corpus in a previous experiment to a corpus in Dutch and compare results. Our task handles several aspects that are common to

different machine learning tasks in NLP application: small amounts of data, inherent ambiguity (definition detection is sometimes a matter of judgment), noisy data (human annotators make mistakes), imbalanced class distribution, this last aspect being the main issue addressed in this paper.

2 Related Work

As we said in the previous section there are two main approach to deal with automatic definition extraction, the rule based and the classification one. Regarding the first approach Hearst [11] proposed a method to identify a set of lexico-syntactic patterns to extract hyponym relations from large corpora and extend WordNet with them. This method was adopted by [19] to cover other types of relations.

DEFINDER [13] is considered a state of the art system. It combines simple cue-phrases and structural indicators introducing the definitions and the defined term. The corpus used to develop the rules consists of well-structured medical documents, 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 [16] focused their works on the extraction of definitory expressions containing hyperonym and synonym relations from French corpora. 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. Turning more specifically to the Portuguese language. Pinto and Oliveira [20] present a study on the extraction of definitions with a corpus from a medical domain. They first extract the relevant terms and then extract definition for each term. An evaluation is carried out for each term; for each term recall and precision are very variable ranging between 0% and 100%.

In the last years machine learning techniques were combined with pattern recognition in order to improve the general results. In particular, [8] used a maximum entropy classifier to extract definition in order to distinguish actual definitions from other sentences. As attributes to classify definition sentences they used-such as n-gram and bag-of-words, sentence position, syntactic properties and named entity classes. The corpus used was composed by medical pages of Dutch Wikipedia, where they extracted sentences based on syntactic features. The data set were composed by 2,299 senteces of which 1,366 actual definitions. This gives an initial accuracy of 59%, that was improved with machine learning algorithms until 92.21%

In [6], it is presented a system to extract definition from off-line documents. They experimented with three different algorithms, namely NaïveBayse, Decision Tree and Support Vector Machine (SVM), obtaining the best score with SVM with a a F-measure of 0.83 with a balanced data set.

In [26] they combine syntactic patterns with a Naïve Bayes classification algorithm with the aim of extracting glossaries from tutorial documents in Dutch. They use several properties and several combination of them, obtaining an improvement of precision of 51.9% but a decline in the recall of 19.1% in comparison with a the syntactic pattern system developed previously by the authors using the same corpus.

Recently, some authors have started to look at this problem of imbalanced data set in the context of definition extraction. In particular, [21] 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 F-measure, in particular with the ratio 1 to 5, they cannot 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 outperform the rule based grammar previously developed of 5 percentage points [14].

3 Corpora

All the two corpora used for experiments were collected in the context of the LT4eL project [1]. They were used to develop different tools, such a key-word extractor, a glossary candidate detector and an ontology, in order to support e-learning activities[1] in a multi-language context. The corpora are encoded with a common XML format. The DTD of this format is conforming to a DTD derived from the XCESAna DTD, a standard for linguistically annotated corpora [18]. 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.

The Dutch Corpus is composed by 26 tutorials with a size of about 350,000 tokens. The corpus was annotated part-of-speech information and morphosyntactic features with the Wotan tagger and with lemmatization information with the CGN lemmatizer (for more information about this corpus see [26].

The Portuguese Corpus is composed by 23 tutorials and scientific papers in the field of Information Technology and has a size of 274,000 tokens. It was then automatically annotated with morpho-syntactic information using the LX-Suite [23] a set of tools for the shallow processing of Portuguese with state of the art performance.

In order to prepare the data set for to be used in our experiments a simple grammar for each language was create that extracts all the sentences where the verb "to be" appears as the main verb. For Dutch we obtained a sub-corpus composed by 4,829, 120 of which are definitions, with a ratio of 39:1. For Portuguese we obtained a sub-corpus composed by 1,360 sentences, 121 of which are definitions, with a ratio of about 10:1.

Commonly used features are: bag-of-word, n-grams [17] (either of part-of-speech or of base forms), the position of the definition inside the document [12], the presence of determiners in the *definiens* and in the *definiendum* [8]. Other relevant properties can be the

[1] www.lt4el.eu

presence of named entities [8] or data from en external source such as encyclopedic data, wordnet, etc. [22].

Some features work well with a corpus but not so well in a different corpus, resulting in the impossibility to use the learner with different corpora. The use of the position of a definition-bearing sentence in [8] is an example of a feature that is corpus dependent. The same issue arise when lexical information is used as feature. In order to avoid such limitation we represented instances as n-grams of POS. From both the corpora the 100 most frequent n-grams were extracted and 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, for question of space, only results obtained with the best representation are showed, that is with bi-grams.

4 Machine Learning Algorithms

Five different algorithms were used: C4.5, Random Forest, Naïve Bayes, k-NN, SVM. The reason that motivated this choice is twofold: we want to cover different class of algorithms and we want to use algorithms representing the state of the art for definition extraction.

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, it is selected a random sample of the data set (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. Regarding C4.5, different configuration were tested: reduced-error pruning instead of C.4.5 pruning, pruned and unpruned option, and with or without Laplace smoothing. Regarding Random Forest, we experimented with different numbers of randomly chosen attributes.

Naïve Bayes is a simple probabilistic classifier that is very popular in natural language application. In spite of its simplicity, it permit to obtain results similar to the results obtained with more complex algorithms. Two different implementation were tested: one in which the numeric estimator precision values are chosen using a kernel estimator for numeric attributes and another using a normal distribution.

The k-NN algorithm is a type of instance-based learning, also called lazy learning because, differently from 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 until the classification phase. In this phase, it 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 separate the point of two classes as far as possible. In this experiment four different classifiers were implemented, using four different kernels, linear, polynominal, radial and sigmoid.

Weka workbench [27] was used to build all the learners.

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 example 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 example in a dataset, 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, others are too similar to be useful, and others are just noise.

It is possible to divide examples in four different classes:

- Noise examples - examples that are incorrectly classified
- Borderline examples - dangerous since a small amount of noise can make them fall on the wrong side of the decision border.
- Redundant examples - too similar to other examples to be useful.
- Safe examples - examples that fit perfectly the class to which they belong.

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

Condensed Nearest Neighbor Rule [10] 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. A subset $E' \subset E$ is consistent with E if using a 1-nearest neighbor,E' correctly classies the examples in E. First, it randomly draw one majority class example and all examples from the minority class and put these examples in E'. Next, it uses a 1-NN over the examples in E' to classify the

examples in E. Every misclassified example from E is moved to E'. It is important to note that this procedure does not find the smallest consistent subset from E. 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 [25] removes both noise and borderline examples. Tomek links are pairs of instances of di?erent classes that have each other as their nearest neighbors. Given two examples x and y belonging to different classes, and $d(x, y)$ the distance between x and y, a (x, y) pair is called a Tomek link if there is not an example z such that $d(x, z) < d(x, y)$ or $d(y, z) < d(x, y)$. If two examples form a Tomek link, then either one of these examples is noise or both examples are border-line. As an under-sampling method, only examples belonging to the majority class are eliminated. The major drawback of Tomek Link undersampling 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 [7] 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). Synthetic samples are produced taking the difference between the feature vector (sample) under consideration and its nearest neighbors. The difference is multiplied by a random number between 0 and 1 and added to the feature vector under consideration.

6 Evaluation Issues

One of the most used metric is the Error Rate, defined as 1.0- (True Positive+True Negative)/(True Positive-False Positive+False Negative+True Negative). 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. Other 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. This means that a metric such as Error Rate is sensitive to class imbalance.

It is possible to derive metrics that are not sensitive to the skew of the data. In particular, four metrics are proposed in [4]:

- False Negative rate: F N /(T P +F N) - the percentage of positive examples misclassified as belonging to the negative class
- False Positive rate: F P /(F P +T N) - the percentage of negative examples misclassified as belonging to the positive class
- True Negative rate: T N /(F P +T N) - the percentage of negative examples correctly classified as belonging to the negative class
- False Positive rate: T P /(T P +F N) - 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 TP rate is plotted on the Y axis and 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.

It is important to notice that the lower left point (0, 0) represents the strategy of never issuing a positive classification: such a classifier produces no false positive errors but also gains no true positives. The opposite strategy, of unconditionally issuing positive classifications, is represented by the upper right point (1, 1).

In order to compare classifiers, it is possible to reduce a ROC curve to a scalar value representing the performance of the classifier. 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 line between(0,0) and(1,1), which has an area of 0.5, no realistic classier should have an AUC less than 0.5. The AUC is equivalent to the Wilcoxon test of ranks and it is also related to Gini coefficient (for an exhaustive description of ROC and AUC in assessing machine learning algorithms see [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 in order to compare our results to previous works in this area. F-measure is a combination of Recall and Precision metrics:

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

7 Results and Discussion

In this section, we show the results obtained with the different learning algorithms and with the different sampling techniques used for both corpora. 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.

Tables 1 and 2 display the performance of the two classifiers using k-NN algorithm. In particular Table1 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 2 displays results obtained by

1-NN				
	P	T	D	U
Sampling	F-m	AUC	F-m	AUC
Original	0.19	0.56	0.06	0.55
Dowsampling	0.62	0.57	**0.57**	0.55
Oversampling	0.36	0.55	0.18	0.52
SMOTE	**0.63**	**0.66**	0.40	**0.70**
CNN	0.23	0.52	0.56	0.54
Tomek	0.57	0.59	0.35	0.56

Table 1: *Results using **k-NN** algorithm with **k=1***

3-NN				
	P	T	D	U
Sampling	F-m	AUC	F-m	AUC
Original	0.17	0.57	0.20	0.51
Dowsampling	0.62	0.59	**0.59**	0.61
Oversampling	0.51	0.58	0.33	0.56
SMOTE	**0.66**	**0.70**	0.42	**0.74**
CNN	0.65	0.61	0.57	0.55
Tomek	0.64	0.66	0.28	0.63

Table 2: *Results using **k-NN** algorithm with **k=3***

C4.5				
	P	T	D	U
Sampling	F-m	AUC	F-m	AUC
Original	0.17	0.65	0.09	0.49
Dowsampling	0.58	0.59	0.66	0.67
Oversampling	0.37	0.67	0.25	0.65
SMOTE	**0.77**	**0.87**	**0.81**	**0.91**
CNN	0.62	0.61	0.55	0.56
Tomek	0.63	0.60	0.58	0.63

Table 3: *Results using **C4.5** algorithm*

Random Forest				
	P	T	D	U
Sampling	F-m	AUC		
Original	0.13	0.65	0.02	0.56
Dowsampling	0.57	0.65	0.61	0.69
Oversampling	0.21	0.64	0.02	0.64
SMOTE	**0.75**	**0.94**	**0.77**	**0.96**
CNN	0.59	0.66	0.58	0.58
Tomek	0.65	0.59	0.61	0.73

Table 4: *Results using **Random Forest** algorithm*

a classifier using a k-NN algorithm with k equal to 3 (3-NN).

Regarding the results obtained with the algorithm 1-NN in Table 1, it is interesting to notice that, for the AUC metric, only the SMOTE sampling technique is able to significantly improve the base line for both corpora. For the Portuguese corpus there is an improvement of 10 points while for the Dutch corpus the improvement is even greater, reaching 15 points. The situation is slightly different for the F-measure. In this case, the best result is obtained by SMOTE for the Portuguese and by down sampling for Dutch. Results obtained with the 3-NN algorithm are very similar to those obtained with the 1-NN in terms of which sampling technique shows the greater improvements. It is worthwhile to notice that although the base lines for the above classifiers 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 and for both languages.

The results displayed in Table 3 refer to the best setting for the C4.5 classifier, where the tree was pruned using the C4.5s standard pruning procedure and no Laplace correction. Regarding Table 4, the classifier was built using 10 different trees. For both corpora SMOTE sampling method presents the best results in terms of AUC and F-measure, but in the case of Dutch the improvement regarding the base line was much greater in comparison with the improvement for Portuguese. Even if the base Iine for Dutch was worst at the end the it outperformed results obtained with the Portuguese corpus. The same observation holds for results present in Table 4.

Table 5 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 or equal to 0.5. With the use of sampling techniques the performance of this classifier is comparable to the 1-NN. 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.

The results in Table 6 refer to a Naïve Bayes classifier using normal distribution. As for the previous algorithm (except for SVM), the best results is obtained with the SMOTE technique, but there is a difference between the two corpora. For the Portuguese data set the base line is higher than for the other classifiers in terms of both metrics taken in consideration, but the improvements achieved with the use of sampling do not outperform the performance of other classifiers, namely C4.5 and Random Forest. On the other hand, for the Dutch data set the best results are obtained with Naïve Bayes even if the initial base line is similar to that obtained with 3-NNm atleast regarding F-measure.

In general for both the languages, the SMOTE sampling technique shows the best results in terms of AUC, followed by Tomek Link and Random oversampling. 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 11

SVM				
	P	T	D	U
Sampling	F-m	AUC	F-m	AUC
Original	0.12	0.48	0.02	0.50
Dowsampling	**0.67**	**0.68**	**0.65**	**0.65**
Oversampling	0.61	0.59	0.60	0.64
SMOTE	0.60	0.60	0.32	0.59
CNN	0.59	0.57	0.61	0.59
Tomek	0.64	0.49	0.63	0.66

Table 5: *Results using **SVM** algorithm*

	Naïve	Bayes		
	P	T	D	U
Sampling	F-m	AUC		
Original	0.24	0.66	0.12	0.75
Dowsampling	0.62	0.62	0.70	0.72
Oversampling	0.67	0.68	0.68	0.75
SMOTE	**0.72**	**0.76**	**0.95**	**0.97**
CNN	0.64	0.63	0.66	0.69
Tomek	0.69	0.72	0.67	0.77

Table 6: *Results using **Naïve Bayes***

UCI data sets [2], Batista and colleagues [4] show that in most cases and with several data sets in different domains SMOTE and Random over-sampling are the most effective methods. In general, they lead to a rise in the AUC metric of few percentage points (1 to 4), when the base line was already high (more than 0.65), while where 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 [15] apply these methods to sentence boundary detection in speech, showing that SMOTE and 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 Link. Batista in [3] 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 link, followed by simple SMOTE, in a case study on automated annotation of keywords.

In our case the improvement regarding the original data set is between 10 and 29 percentage 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 using learning algorithms, confirming the importance of sampling techniques in supporting definition extraction tasks. [26], using the same corpus we used, reports on a F-measure of 0.73, obtained with a combination of syntactic rules and a Naïve Bayes classifiers for Dutch while [21], with a similar approach, but for the Polish language, obtain a F-measure of 0.35. Furthermore in all these works a combination of features are used in order to reach best results, while in this paper we only use bi-grams of POS as features. To conclude, our results are comparable with systems that represent the state of the art in the area, such as DEFINDER, which shows a F-mesure of 0.80.

8 Conclusions and Future Work

In this paper we have compared the performance of different learning algorithms and different sampling technique on a definition extraction task, using data sets in different language. Results presented show that this approach can be very effective in comparison to handcrafted rule to extract definitions, in terms of amount of time and performance. Furthermore techniques here presented are language and domain independent, making them a interesting resource in the field of Question Answering. Next steps in our researches will be integrate our classifier in a QA system in order to test this results in a much real world context.

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

References

[1] M. Avelãs, A. Branco, R. D. Gaudio, and P. Martins. Supporting e-learning with language technology for portuguese. In *Proceedings of the International Conference on the Computational Processing of Portuguese (PROPOR2008)*. Springer, 2008.

[2] G. Barnbrook. *Defining Language: a local grammar of definition sentences*. John Benjamins Publishing Company, 2002.

[3] G. E. A. P. A. Batista, A. L. C. Bazzan, and M. C. Monard. Balancing training data for automated annotation of keywords: a case study, 2003.

[4] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, 2004.

[5] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.

[6] X. Chang and Q. Zheng. Offline definition extraction using machine learning for knowledge-oriented question answering. In D.-S. Huang, L. Heutte, and M. Loog, editors, *ICIC (3)*, volume 2 of *Communications in Computer and Information Science*, pages 1286–1294. Springer, 2007.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[8] I. Fahmi and G. Bouma. Learning to identify definitions using syntactic feature. In R. Basili and A. Moschitti, editors, *Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications*, Trento, Italy, 2006.

[9] H. Gernot. Defining patterns in translation studies: Revisiting two classics of german translation. *Translationswissenschaft in Target*, 19(2):197–215, 2007.

[10] P. E. Hart. The condensed nearest neighbor rule (corresp.). *Information Theory, IEEE Transactions on*, 14(3):515–516, May 1968.

[11] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[12] H. Joho and M. Sanderson. Retrieving descriptive phrases from large amounts of free text. In *Proceeding of hte 9th international conference on Information and knowledge management*, pages 180–186, 2000.

[13] J. Klavans and S. Muresan. Evaluation of the DEFINDER system for fully automatic glossary construction. In *Proceedings of the American Medical Informatics Association Symposium (AMIA 2001)*, 2001.

[14] L. Kobylinski and A. Przepiorkowski. Definition extraction with balanced random forests. In A. Ranta, editor, *GoTAL 2008*, pages 237–247, Gothenburg, 2008. Springer-Verlag Berlin Heidelberg.

[15] Y. Liu, N. V. Chawla, M. P. Harper, E. Shriberg, and A. Stolcke. A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech & Language*, 20(4):468–494, 2006.

[16] V. Malais, P. Zweigenbaum, and B. Bachimont. Detecting semantic relations between terms in definitions. In *the 3rd edition of CompuTerm Workshop (CompuTerm 2004) at Coling 2004*, pages 55–62, 2004.

[17] S. Miliaraki and I. Androutsopoulos. Learning to identify single-snippet answer to definition questions. In *Proceeding of the 20th International Conference on Computational Linguistic (COLING 2004)*, pages 1360–1366, Geneva, Switzerland, 2004.

[18] I. N. and S. K. Xml, corpus encoding standard, document xces 0.2. Technical report, Department of Computer Science, Vassar College and Equipe Langue ed Dialogue, New York, USA and LORIA/CNRS, Vandouvre-les-Nancy, France, 2002.

[19] J. Person. The expression of definitions in specialised text: a corpus-based analysis. In M. Gellerstam, J. Jaborg, S. G. Malgren, K. Noren, L. Rogstrom, and C. Papmehl, editors, *7th Internation Congress on Lexicography (EURALEX 96)*, pages 817–824, Goteborg, Sweden, 1996.

[20] A. S. Pinto and D. Oliveira. Extraçcão de definições no Corpógrafo. Technical report, Faculdade de Letras da Universidade do Porto, 2004.

[21] A. Przepiorkowski, M. Marcinczuk, and L. Degorski. Noisy and imbalanced data: Machine learning or manual grammars? In *Text, Speech and Dialogue: 9th International Conference, TSD 2008*, Brno, Czech Republic, September 2008. Lecture Notes in Artificial Intelligence, Berlin, Springer-Verlag.

[22] H. Saggion. Identifying definitions in text collections for question answering. In *LREC 2004*, 2004.

[23] J. R. Silva. Shallow processing of Portuguese: From sentence chunking to nominal lemmatization. Master's thesis, Universidade de Lisboa, Faculdade de Ciências, 2007.

[24] E. Tjong, K. Sang, G. Bouma, and M. de Rijke. Developing offline strategies for answering medical questions. In *Proceedings of the AAAI-05 workshop on Question Answering in restricted domains*, pages 41–45, 2005.

[25] I. Tomek. Two modifications of cnn. *Systems, Man and Cybernetics, IEEE Transactions on*, 6(11):769–772, November 1976.

[26] E. Westerhout and P. Monachesi. Extraction of Dutch definitory contexts for elearning purposes. In *CLIN proceedings 2007*, 2007.

[27] I. H. Witten and E. Frank. *Data Mining: Pratical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, 2005.