Applying Dependency Relations to Definition Extraction

Luis Espinosa-Anke and Horacio Saggion

Tractament Automàtic del Llenguatge Natural
Department of Information and Communication Technologies
Universitat Pompeu Fabra
luis.espinosa@upf.edu
horacio.saggion@upf.edu**

Abstract. Definition Extraction (DE) is the task to automatically identify definitional knowledge in naturally-occurring text. This task has applications in ontology generation, glossary creation or question answering. Although the traditional approach to DE has been based on hand-crafted pattern-matching rules, recent methods incorporate learning algorithms in order to classify sentences as definitional or non-definitional. This paper presents a supervised approach to Definition Extraction in which only syntactic features derived from dependency relations are used. We model the problem as a classification task where each sentence has to be classified as being or not definitional. We compare our results with two well-known approaches: First, a supervised method based on Word-Class Lattices and second, an unsupervised approach based on mining recurrent patterns. Our competitive results suggest that syntactic information alone can contribute substantially to the development and improvement of DE systems.

1 Introduction

Encyclopedias and terminological databases are knowledge bases of great importance for human and automatic processing, but their manual development is costly and slow [3]. Definition Extraction (DE), which is the task to automatically identify definitions in naturally-occurring text, can play an important role in this context. DE has received an increasing interest in Natural Language Processing, Computational Linguistics and Computational Lexicography. It has been proven effective for developing glossaries [16] [21], lexical databases [17] [34], developing question answering tools [25] [5], supporting terminology applications [13] [29] [28], automatically acquiring ontologies [39] or for developing e-learning applications [35] [8].

DE has been widely studied as a binary classification problem, where each sentence should be assigned the correct class, i.e. whether it is a definition. The majority of systems developed in the last few years addressing DE are based

** <http://taln.upf.edu>

on hand-crafted rules aiming at identifying definitions in texts through pattern-matching, although a renewed interest in incorporating learning algorithms has emerged [7]. Following this line of work, we introduce a method for DE that exploits linguistically motivated features extracted from dependency relations. Benefiting from syntactic information extracted from dependency structures has proven efficient in other fields, like Information Extraction [32] [30] [1] or paraphrase identification [33]. With regard to DE, we are unaware of any supervised system which relies only on features derived from dependency relations, although these have been previously used, either for the extraction of definition candidates [9] [36] or as part of more comprehensive feature sets [6].

We evaluate the performance of our system and compare it with a supervised method based on word-class lattices [18] and with an unsupervised method that exploits recurrent patterns at various linguistic levels [38]. Moreover, our approach relies only on Machine Learning and skips the pattern-crafting stage, which seems to address issues like language-dependence and domain-specificity [7]. The main contributions of this work can be summarized as:

- A set of features based exclusively on dependency relations for identifying definitional statements in text.
- A set of experiments that demonstrate competitive performance when compared with state-of-the-art systems.

The remainder of this paper is divided as follows. Section 2 provides the reader with a general review of the most prominent work carried out in the field. Section 3 describes in detail the datasets used, the method followed for designing the features, and provides an illustrative example. Section 4 describes the experimental setup. Next, Section 5 covers the evaluation metrics and provides results with respect to comparable work in the field. Finally, Section 6 presents the main conclusions that can be drawn from the system described, as well as suggesting potential lines for future work.

2 Related Work

Traditionally, DE has been approached as a pattern-matching problem, where a set of hand-crafted rules are designed and applied to a dataset. One of the cue phrases that has been looked at has been the presence of specific trigger verbs [23] [25] [27] [31]. Moreover, other features like punctuation or layout features have also been considered in the pattern-matching process [16] [12] [26] [22] [15].

However, the increasing number of systems which are based totally or partially on Machine Learning reflect the current trend in DE. For instance, [9] exploit features like text, document and syntactic properties as well as named entity tags for training a classifier. [5] contributed with probabilistic lexico-syntactic patterns to modelling definitions. A soft matching model is described based on an n-gram language model. [4] use genetic programming to learn rules for typical linguistic forms in definitional sentences, and then genetic algorithms to weight these rules. Another generalizing approach relies on Word Class Lattices, which

are generated from sentence clusters and applied to identify different types of definitions [18]. On the other hand, unsupervised pattern-mining approaches take into consideration sets of definitional sentences, which are extracted at three different levels (Ngram patterns, subsequence patterns, and dependency subtree patterns) [38]. Moreover, [24] compare two methods: First, a bootstrapping approach where lexico-syntactic patterns and candidates for definition terms are iteratively constructed, and second, a deep semantic analysis which exploits the semantic predicate-argument structure.

Finally, let us refer to recent work that addresses the DE problem not as a binary classification task, but as a sequential labeling task where each token is classified as being part of a definition or not [11] [8]. In both cases, the classification is performed using a probabilistic model based on Conditional Random Fields. However, these works differ in that the former aims at classifying different parts of a definition whereas the latter, although makes use of these features, does not explicitly tag these components.

3 Methodology

This section discusses the methodology followed for the implementation of our experiments. Firstly, the dataset used is described and, secondly, the linguistic motivation behind our data modelling is provided, followed by a description of the feature set.

3.1 Dataset

Our dataset consists of a collection of 1,908 definitional sentences and 2,711 non-definitional sentences extracted from Wikipedia [19]. Following the terminology generalization approaches from [18] and [38], all the definiendum¹ terms are reduced to the token *TARGET* in order to include this wildcard as a feature, thus assuming a prior step in terminology identification. In this context, a term is considered to be a word or phrase that has an entry in the English Wikipedia.

3.2 Modelling the Data

In the linguistic theory of Dependency Grammar, the notion of dependency is fundamental [20]. A syntactic structure is described by the distribution of lexical elements linked by asymmetrical relations called dependencies. One of the main characteristics is that, unlike constituent structures, a dependency tree has no phrasal nodes. Moreover, the dependency representations provide a direct encoding of predicate-argument structures (see the dotted dependencies in Figure 1). Finally, the relations between units in a dependency tree are bilexical, which makes them beneficial for disambiguation [20].

¹ The *definiendum* refers to the term to be defined, while the *definiens* is a cluster of words that defines the term.

Within the dependency structure of a sentence, certain relations seem to be more informative in the context of DE. For example, sentences S1 and S2 below share the same surface structure (*TARGET is * which was **). However, only S1 is a definition [38].

S1: *TARGET is the independent school which was opened.*

S2: *TARGET is secure against CCA2 which was proved.*

The main difference pointed out is that the dependency relation between the verb and the phrase “the independent school” is of **objective noun phrase**, while in S2, the relation with the adjective “secure” is of **adjectival phrase**. This is the kind of information that (in S1) an *Ngram* approach would be unable to tackle due to the non-adjacent distance between the components.

In the specific case of this paper, we consider two kinds of subtrees, which are: Parent-Child-Grandchild (PCG) (left) and Parent-Child-Child (PCC).

The PCG subtree allows extracting sequences like $Verb \rightarrow NN \rightarrow IN$, which in definitional sentences is an indicator of a hypernym being introduced and described (e.g. *The TARGET is a **family** of personal computers originally developed by Amiga Corporation*). This pattern accounts for 7% of all the 3-node subtrees in the definitional sentences present in the Wikipedia corpus, which can be considered to be high given the great sparseness of this set.

Fig. 1. Example sentence with dependency parsing

Syntactic dependency relations can reveal the domain or discipline governing a definition, and can be expressed by a locative at the beginning of a sentence. Consider the following sample sentence: ***In law**, an **TARGET is** a brief statement that contains the most important points of a long legal document or of several related legal papers.* The highlighted words $is(Verb, Root) \rightarrow In(Prep, Loc) \rightarrow law(NN, PMOD)$ form a *Parent $\rightarrow$ Child $\rightarrow$ Grandchild* subtree, where the locative preposition *in* connects the topic with the verb. In the definitional section of the Wikipedia corpus, almost 20% of all the definitions have this structure at the beginning of the sentence.

Moreover, TARGET-is-a-hypernym patterns are a potential candidate for our subtree-based feature extractor. By attempting an ngram-based approach, we would be approaching the task as a surface pattern matching, perhaps including the Part-of-Speech of the candidate hypernym (to disregard, for example, noun phrases whose first word is an adjective). However, by looking at the syntactic function of the noun phrases involved (SBJ for TARGET, and Predicate for the hypernym), we can filter out some of the noisy subtrees retrieved.

With regard to the PCC subtree type, this can be useful for identifying SVO relations [30], as well as extracting multiword terminology. This can be further illustrated with the following sentence: *An **TARGET is** a **segmental writing system** which is based on consonants but in which vowel notation is obligatory.* The highlighted pattern follows a $(segmental, NN, NMOD) \leftarrow (system, NN, PRED) \rightarrow (writing, NN, NMOD)$ structure, and since we know that the word *system* is the syntactic predicate, we are highly confident that it can constitute the hypernym of the term to be defined.

Finally, let us highlight some informative instances of the PCG and PCC hybrid subtrees that are considered in our approach: (1) non-adjacent subject-predicate patterns: $(NN, SBJ) \leftarrow (VBZ) \rightarrow (NN, PRD)$; (2) description of the definiens' head: $(NN, PRD) \rightarrow (IN) \rightarrow (NN, PMOD)$; and (3) synonymy relation between the genus and the hypernym in a definitional sentence: $(NN, PRD) \rightarrow (CC, COORD) \rightarrow (NN)$.

4 Experimental Setup

After reviewing and justifying the potential of dependency-based syntax analysis for DE, we discuss the features implemented for modelling our system. These are developed after parsing the initial set of documents with a graph-based parser [2].

1. **Subtrees:** From all the available three-node subtree combinations, we extract the following information for each node: Surface form (SF), Part-of-Speech (POS), and Dependency Relation (DR). Each sentence is transformed to a feature vector of the 15 most frequent subtrees of each type. The six types used in this feature as well as examples are shown in Table 1. Additionally, let us highlight that the number of subtrees used as features is not arbitrary. It comes from a manual analysis of the frequency distribution of

each type across the dataset. The definitional sentences in the Wikipedia corpus tend to have recurrent syntactic patterns, which provokes a long-tailed frequency distribution and thus a remarkable gap between systematic and idiosyncratic features. By keeping only the 15 most frequent subtrees we design a consistent feature set across the types while at the same time disregarding the long tail in each type (see Figure 2 for an illustrative example of three types of subtrees).

Fig. 2. Distribution of three selected types of hybrid subtrees illustrating the asymmetry of their frequency distribution.

2. **Degree of X :** The *degree* of a node X in a graph is the number of edges adjacent to it, i.e. the sum of of its children + 1 (its head). We reduce the search space of X to

$$X \in \{PRD, SUBJ, APPO\}$$

because in this way, subject nodes with many modifiers are given more importance. For example, in the sample sentence in Figure 1, $\text{degree}(PRD) = 4$.

Type of Subtree	Example
(SF, SF, SF)	(<i>TARGET</i> , refers, to) (is, used, in)
(POS, POS, POS)	(DT, JJ, NN) (IN, NN, VBD)
(DR, DR, DR)	(SBJ, ROOT, PRD) (PMOD, COORD, CONJ)
(SF + POS, SF + POS, SF + POS)	((is, VBZ), (a, DT), (unit, NN)), ((a, DT), (form, NN), (of, IN))
(SF + DR, SF + DR, SF + DR)	((In, LOC), (<i>TARGET</i> , SBJ), (was, ROOT)) ((is, ROOT), (any, PRD), (of, NMOD))
(POS + DR, POS + DR, POS + DR)	((NN, PMOD), (CC, COORD), (NN, CONJ)) ((DT, NMOD), (NNP, NAME), (NNP, PMOD))

Table 1. Summary of the types of subtrees used and examples for each type. SF = Surface Form, POS = Part of Speech, DR = Dependency Relation.

3. **Morphosyntactic chains starting in node X :** X can have the same node value as in the previous feature. If it exists, we extract all the children from that node in a recursive fashion. We then extract the POS and dependency relation chains and order it according to their index in the sentence. This approach has been used successfully in other NLP tasks, such as Semantic Role Labeling [10]. For example, in our sample sentence we would extract the following chain from the *PRD* node: *NMOD* – *PRD* – *COORD* – *CONJ* – *NMOD* – *PMOD*..., and so on until the last child is reached in recursive manner. Note that the head node (*PRD*) does not necessarily constitute the first element in the chain, as it has a child appearing before in the sentence (*(the, DT, NMOD)*).
4. **Ordered cartesian product of two subtrees:** The ordered cartesian product of two graphs G_1 and G_2 produces a new graph H with the vertex set $V(G_1) \times V(G_2)$, with the tuples $\{(i_1, i_2), (j_1, j_2)\}$ forming an edge if $\{i_1, j_1\}$ forms an edge in G_1 and $i_2 = j_2$, or $\{i_2, j_2\}$ forms an edge in G_2 and $i_1 = j_1$. Our intuition is that by extending the relationships between pairs of specific head nodes and their children, relations between modifiers of *SBJ* and *PRD* nodes, for example, would be captured and reinforced. We perform this operation only if the head of G_1 has the syntactic function *SBJ* and the head of G_2 has the head *PRD* or *APPO*. The result is a string that contains the surface, Part-of-Speech or dependency relation chain of the resulting graph H .
5. **Semantic similarity:** We hypothesize that we high semantic similarity between definiens and predicate, for example, might point towards a definitional sentence. In our sample sentence, this would be the case between *abortion* and *removal*. We extend this feature to other nodes like appositives or their postmodifiers, and apply it to the following pairs: (*SBJ, PRD*), (*SBJ, APPO*), (*PRD, JJ_PMOD*) and (*APPO, JJ_PMOD*). The Leacock Chodorow Similarity between each pair is computed, following the path between them in the WordNet hierarchy [14].

5 Evaluation

The above features are incorporated in a sentence’ feature vector, and this information is used for training different classification algorithms present in the Weka workbench [37]. The evaluation results we report are based on 10-fold cross-validation. It is important to highlight that the Bag-of-Subtrees features are extracted only from the training set in each fold, which ensures there is no bias during the feature design stage, and the generalization potential is appropriately measured. Table 2 shows the scores for the different setups on which the experiments were carried out. For each algorithm, the different setups designed are: S_1 includes the full feature set, S_2 disregards chain and cartesian product features, and S_3 disregards chain, cartesian product, degree and similarity features. Likewise, Table 3 shows a comparative table among the different systems studied, which are the following:

- **Bigrams:** Baseline 1 based on the bigram classifier for soft pattern matching proposed by [5].
- **Star Patterns:** Baseline 2 based on a pattern-matching approach in which infrequent words are replaced by ‘*’ and then matched against candidate definitions [18].
- **WCL 1:** Implementation of the Word-Class Lattice model where a lattice is learned for each cluster of sentences [18].
- **WCL 3:** Implementation of the Word-Class Lattice model where a lattice is learned for each of the components of a definition, namely *Definiendum*, *Definitior* and *Definiens* [18].
- **PMA:** Pattern Mining Approach based on *Ngram*, subsequence and dependency subtrees patterns [38].
- **Our proposal:** An implementation of our approach with the highest scoring setup.

In the light of these scores, it seems reasonable to argue that a classification approach for DE can improve substantially by including features that account for the morphosyntactic structure of the sentence. Moreover, deep analyses of syntactic trees and the relation among dependents contributes decisively to DE. While syntactic and graph-based information is exploited in both [18] and [38], our results suggest that an approach exclusively reliant on these features can perform in a competitive fashion.

	NaïveBayes			VPerceptron			SVM			LogisticR			DTrees			RandomF		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
S_1	81.9	78	75.9	78.9	85.2	84.4	85.9	85.3	85.4	83.4	82.7	84.7	85.7	85.3	85.2	83.4	82.9	82.2
S_2	75.7	75.9	75.5	80.5	79.4	79.5	82	81.2	81.3	82.2	81.5	81.6	80.8	79.4	79.6	79.9	79.4	79.5
S_3	53.1	58.6	49	56.9	59.8	52.2	56.9	59.8	52.2	56.9	59.8	52.2	55	59.2	49.3	56.9	59.8	52.2

Table 2. Scores of our approach after being tested with several learning algorithms.

The highest scoring approach in terms of Precision is achieved by the WCL systems, with almost 1. The highest score in terms of Recall (83.5) and F-Measure (91.30), on the other hand, is the PMA system. After reviewing our own results, two main conclusions can be reached: (1) Our method improves substantially in terms of Recall and F-Measure with respect to the WCL approach, although it performs below the PMA approach in these metrics in about 10%². (2) Although our approach does not outperform the aforementioned approaches, it does provide an insight towards highly informative features that could potentially be used in the future together with shallower features.

	Our proposal	WCL 1	WCL 2	PMA	Star Patterns	Bigrams
Precision	85.9	99.8	99.8	89.1	86.7	66.7
Recall	85.3	42.1	60.7	93.5	66.1	82.7
F-Measure	85.4	59.2	83.5	91.3	75.1	73.9

Table 3. Comparative table of results between our approach and the reported scores in Navigli and Velardi (2010) and Yan et al. (2012).

6 Conclusions and Future Work

An approach to the task of DE based on morphosyntactic information derived from dependency relations has been described. From the initial hypothesis that syntactic features are highly discriminatory in predicting whether a sentence is or not a definition, we have justified the feature design and selection. Next, dataset on which the experiments have been carried out has been presented to the reader, followed by the experimental setup and the results, which seem to be encouraging enough to argue that features extracted from dependency relations can constitute the core of a successful DE system.

The main findings of this work can be summarized as follows:

- **Bag-of-Subtrees:** These features alone have proven to be uninformative enough as to constitute the backbone of a DE system. Drops in up to 30% in performance were identified after removing all the other features.
- **Subtree chain combinations:** Our experiments show that features that look at a subtree’s children or the cartesian product of certain subtrees can contribute to improving Precision in a DE system. This is particularly significant in certain algorithms and setups, like Naïve Bayes, where an increase of 7% was achieved.

As for potential lines of future work, we are currently investigating four possible methods for improving the performance of DE systems. On one hand, we are

² An implementation of this method and an evaluation of the contribution of tree-based ngrams is expected to be carried out in the future.

interested in testing the extent to which semantic roles can help discriminating clusters of words as being part of a *definiens*, which we believe could be used as a previous step to DE. On the other hand, we are also looking at the potential of sequential labelling for DE. Previous work [11] [8] showed that this approach has several advantages over full-sentence classification. E.g. it allows identifying cross-sentence definitions, helps pruning out noise in very long sentences which encode short definitions, and helps identifying not only full definitions but also their components (e.g. *definiens*, *definiendum* or its genus, where present). Moreover, moving out of a domain like Wikipedia would constitute a real-world challenge for this approach and would be appropriate for testing its generalization potential. And finally, exhaustive error analysis would be advisable in order to understand the reasons behind some of the misclassified instances during our experiments. This information could be used for designing more efficient feature sets.

Acknowledgments. We would like to express our gratitude to the reviewers for their helpful comments. We also acknowledge partial support from project Dr. Inventor (FP7-ICT-2013.8.1 611383), programa Ramón y Cajal 2009 (RYC-2009-04291), and project number TIN2012-38584-C06-03 from Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovación, Spain.

References

1. Afzal, N., Mitkov, R., Farzindar, A.: Unsupervised relation extraction using dependency trees for automatic generation of multiple-choice questions. In: Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence. pp. 32–43. Canadian AI’11, Springer-Verlag, Berlin, Heidelberg (2011)
2. Bohnet, B.: Very high accuracy and fast dependency parsing is not a contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics. pp. 89–97. COLING ’10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
3. Bontas, E.P., Mochol, M.: Towards a cost estimation model for ontology engineering. In: Eckstein, R., Tolkendorf, R. (eds.) Berliner XML Tage. pp. 153–160 (2005)
4. Borg, C., Rosner, M., Pace, G.: Evolutionary algorithms for definition extraction. In: Proceedings of the 1st Workshop in Definition Extraction (2009)
5. Cui, H., Kan, M.Y., Chua, T.S.: Generic soft pattern models for definitional question answering. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 384–391. ACM (2005)
6. Degórski, L., Marcińczuk, M., Przepiórkowski, A.: Definition extraction using a sequential combination of baseline grammars and machine learning classifiers. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008. ELRA, Marrakech (2008)
7. Del Gaudio, R., Batista, G., Branco, A.: Coping with highly imbalanced datasets: A case study with definition extraction in a multilingual setting. Natural Language Engineering pp. 1–33 (2013)

8. Espinosa, L.: Towards definition extraction using conditional random fields. In: Proceedings of RANLP 2013 Student Research Workshop. pp. 63–70 (2013)
9. Fahmi, I., Bouma, G.: Learning to identify definitions using syntactic features. In: Proceedings of the Workshop on Learning Structured Information in Natural Language Applications. pp. 64–71 (2006)
10. Hacioglu, K.: Semantic role labeling using dependency trees. In: International Conference on Computational Linguistics (COLING) (2004)
11. Jin, Y., Kan, M.Y., Ng, J.P., He, X.: Mining scientific terms and their definitions: A study of the ACL anthology. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 780–790. Association for Computational Linguistics, Seattle, Washington, USA (2013)
12. Malaisé, V., Zweigenbaum, P., Bachimont, B.: Detecting semantic relations between terms in definitions. In: Ananadiou, S., Zweigenbaum, P. (eds.) International Conference on Computational Linguistics (COLING 2004) - CompuTerm 2004: 3rd International Workshop on Computational Terminology. pp. 55–62. COLING, Geneva, Switzerland (2004)
13. Meyer, I.: Extracting knowledge-rich contexts for terminography. Recent advances in computational terminology 2, 279 (2001)
14. Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
15. Monachesi, P., Westerhout, E.: What can NLP techniques do for eLearning? In: International Conference on Informatics and Systems (INFOS08). pp. 150–156 (2008)
16. Muresan, A., Klavans, J.: A method for automatically building and evaluating dictionary resources. In: Proceedings of the Language Resources and Evaluation Conference (LREC) (2002)
17. Nakamura, J.i., Nagao, M.: Extraction of semantic information from an ordinary english dictionary and its evaluation. In: Proceedings of the 12th Conference on Computational Linguistics - Volume 2. pp. 459–464. COLING '88, Association for Computational Linguistics, Stroudsburg, PA, USA (1988)
18. Navigli, R., Velardi, P.: Learning word-class lattices for definition and hypernym extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1318–1327. ACL '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
19. Navigli, R., Velardi, P., Ruiz-Martínez, J.M.: An annotated dataset for extracting definitions and hypernyms from the web. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). pp. 3716–3722. European Language Resources Association (ELRA), Valletta, Malta (2010)
20. Nivre, J.: Dependency grammar and dependency parsing. Tech. rep., Växjö University (2005)
21. Park, Y., Byrd, R.J., Boguraev, B.K.: Automatic Glossary Extraction: Beyond Terminology Identification. In: Proceedings of the 19th International Conference on Computational Linguistics. pp. 1–7. Association for Computational Linguistics, Morristown, NJ, USA (2002)
22. Przepiórkowski, A., Spousta, M., Simov, K., Osenova, P., Lemnitzer, L., Kubo, V., Wójtowicz, B.: Towards the automatic extraction of definitions in Slavic. In: Proceedings of the BSNLP workshop at ACL 2007. pp. 43–50 (2007)
23. Rebeyrolle, J., Tanguy, L.: Repérage automatique de structures linguistiques en corpus: le cas des énoncés définitoires. Cahiers de grammaire 25, 153–174 (2000)

24. Reiplinger, M., Schäfer, U., Wolska, M.: Extracting glossary sentences from scholarly articles: A comparative evaluation of pattern bootstrapping and deep analysis. In: *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. pp. 55–65. Association for Computational Linguistics, Jeju Island, Korea (2012)
25. Sagghion, H., Gaizauskas, R.: Mining on-line sources for definition knowledge. In: *17th FLAIRS*. Miami Beach, Florida (2004)
26. Sánchez, A., Márquez, J.: Hacia un sistema de extracción de definiciones en textos jurídicos. In: *Actas de la 1er Jornada Venezolana de Investigación en Lingüística e Informática*. pp. 1–10 (2005)
27. Sarmiento, L., Maia, B., Santos, D., Pinto, A., Cabral, L.: Corpógrafo V3 From Terminological Aid to Semi-automatic Knowledge Engineering. In: *5th International Conference on Language Resources and Evaluation (LREC'06)*. Geneva (2006)
28. Seppälä, S.: A proposal for a framework to evaluate feature relevance for terminographic definitions. In: *Proceedings of the 1st Workshop on Definition Extraction*. pp. 47–53. WDE '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
29. Sierra, G., Alarcón, R., Aguilar, C., Barrón, A.: Towards the building of a corpus of definitional contexts. In: *Proceeding of the 12th EURALEX International Congress*, Torino, Italy. pp. 229–40 (2006)
30. Stevenson, M., Greenwood, M.A.: Comparing information extraction pattern models. In: *Proceedings of the Workshop on Information Extraction Beyond The Document*. pp. 12–19. IEBeyondDoc '06, Association for Computational Linguistics, Stroudsburg, PA, USA (2006)
31. Storrer, A., Wellinghoff, S.: Automated detection and annotation of term definitions in German text corpora. In: *Conference on Language Resources and Evaluation (LREC)*. pp. 275–295 (2006)
32. Sudo, K., Sekine, S., Grishman, R.: An improved extraction pattern representation model for automatic ie pattern acquisition. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*. Sapporo, Japan (2003)
33. Szpektor, I., Tanev, H., Dagan, I., Coppola, B.: Barcelona
34. Walter, S., Pinkal, M.: Automatic extraction of definitions from German court decisions. In: *Proceedings of the workshop on information extraction beyond the document*. pp. 20–28. Association for Computational Linguistics (2006)
35. Westerhout, E., Monachesi, P.: Extraction of Dutch definitory contexts for elearning purposes. *Proceedings of the Computational Linguistics in the Netherlands (CLIN 2007)*, Nijmegen, Netherlands pp. 219–34 (2007)
36. Westerhout, E., Monachesi, P.: Creating glossaries using pattern-based and machine learning techniques. *Proceedings of the 6th Conference on Language Resources and Evaluation (LREC)* pp. 3074–3081 (2008)
37. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann (2005)
38. Yan, Y., Hashimoto, C., Torisawa, K.: Pattern mining approach to unsupervised definition extraction. In: *Speech Processing Society 18th Annual Meeting (in Chinese)* (2005)
39. Zhang, C., Jiang, P.: Automatic extraction of definitions. In: *2nd IEEE International Conference on Computer Science and Information Technology*. pp. 364–368. Ieee (2009)