Dependency-Based Text Compression for Semantic Relation Extraction

Marcos Garcia
Center for Research in
Information Technologies (CITIUS)
University of Santiago de Compostela
marcos.garcia.gonzalez@usc.es

Pablo Gamallo
Center for Research in
Information Technologies (CITIUS)
University of Santiago de Compostela
pablo.gamallo@usc.es

Abstract

The application of linguistic patterns and rules are one of the main approaches for Information Extraction as well as for high-quality ontology population. However, the lack of flexibility of the linguistic patterns often causes low coverage. This paper presents a weakly-supervised rule-based approach for Relation Extraction which performs partial dependency parsing in order to simplify the linguistic structure of a sentence. This simplification allows us to apply generic semantic extraction rules, obtained with a distant-supervision strategy which takes advantage of semi-structured resources. The rules are added to a partial dependency grammar, which is compiled into a parser capable of extracting instances of the desired relations. Experiments in different Spanish and Portuguese corpora show that this method maintains the high-precision values of rule-based approaches while improves the recall of these systems.

1 Introduction

In recent years, the interest in obtaining structured data from unstructured resources has been increased, namely due to the exponential growth of information in the Web. Regarding this objective, Relation Extraction (RE) aims to automatically identify semantic relations between entities. For instance, from the sentence "Nick Cave was born in the small town of Warracknabeal", a RE system may identify that Warracknabeal is the birthplace of Nick Cave.

The obtained data are arranged in machine-readable formats ("Nick Cave `hasBirthplace` Warracknabeal"), and then incorporated into databases and ontologies, used to improve applications such as Question Answering engines or Information Retrieval systems.

RE systems usually need a set of sentences containing instances of a semantic relation (e.g., `hasBirthplace`). These sentences are processed in order to provide a rich *linguistic space* with different knowledge (tokens, lemmas, PoS-tags, syntactic dependencies, etc.). This knowledge is used to extract semantic relations by (i) training machine-learning classifiers or by (ii) applying on large corpora lexico-syntactic patterns (LSP) derived from the linguistic space.

Relation Extraction approaches rely on the assumption that lexico-syntactic regularities (e.g., LSP) may characterize the same type of knowledge, such as semantic information. However, one of the main problems of these strategies is the low coverage of LSP, which varies with small differences in punctuation, adjective or adverb modification, etc. For instance, the previous example sentence could be represented in a great variety of manners:

- "Nick Cave was born in the small town of Warracknabeal"
- "Nick Cave was born in the town of Warracknabeal"
- "Nick Cave was born in Warracknabeal"
- "Nick Cave, born in the small town of Warracknabeal"

Both machine learning and pattern-matching approaches attempt to avoid this problem by using

Proceedings of the Workshop on Information Extraction and Knowledge Acquisition, pages 21–28,
Hissar, Bulgaria, 16 September 2011.

larger training data or by applying syntactic parsers that identify the constituents of a sentence as well as their functions. However, obtaining large collections of high-quality training data for different semantic relations is not always feasible, since a lot of manual effort is needed. Furthermore, parsers for other languages than English often perform very partial analyses, or are not freely available.

In this paper, we introduce a rule-based approach for RE that overcomes the low coverage problem by simplifying the linguistic structures: we perform a sort of sentence compression technique that uses partial dependency parsing to remove some satellite elements from the input of the extraction rules.

In order to obtain high-quality extraction rules, we use a distant-supervision strategy that takes advantage of semi-structures resources, such as Wikipedia infoboxes or Freebase:[1] First, large sets of semantically related pairs are used for automatically extracting and annotating sentences containing instances of the desired relation. Then, we transform these sentences into LSP, which are generalized through a longest common string algorithm. Finally, the generic patterns are converted into syntactico-semantic rules and added to a dependency grammar.

We performed several experiments with different semantic relations in Portuguese and Spanish, using encyclopedic and journalistic corpora. The results show that dependency-based text compression allows us to improve the recall without losing the high precision values of pattern-matching techniques.

This paper is organized as follows: Section 2 introduces some related work. Section 3 presents the motivation of our Relation Extraction method. Then, Sections 4 and 5 show the strategy for extracting patterns as well as the method for transforming them into semantic rules. In Section 6, some experiments are performed. Finally, Section 7 reports the conclusions of our work.

2 Related Work

In this section we briefly introduce some related work concerning text compression methods as well as strategies for semantic Relation Extraction.

In recent years, several approaches have been proposed for sentence compression, whose aim is to reduce the size of a text while preserving its essential information (Chandrasekar et al., 1996). There are statistical methods (with different degree of supervision) for sentence compression, which require training corpora in order to learn what constituents could be removed from the original input (Clarke and Lapata, 2006). Cohn and Lapata (2009) present Tree-to-Tree Transducer, a state-of-the-art sentence compression method which transforms a source parse tree into a compressed parse tree. We have to note that our approach differs from common sentence compression strategies in a key point: it is not centered in maintaining the grammaticality of a sentence, but just in simplifying its structure and keeping its essential information.

Regarding Relation Extraction, Hearst (1992) was the first one to experiment a pattern-based strategy for the identification of semantic relations, using a small set of initial patterns to get hyperonymy relations by means of a bootstrapping technique. In Brin (1998), a similar method is applied, but it only selects those patterns that show a good performance. Other works make use of Question-Answering pair examples to automatically extract patterns (Ravichandran and Hovy, 2002). A novelty of this system lies in the application of a suffix tree, leading to discover generalized patterns by calculating their common substrings.

In the previously cited work, the learning process starts with patterns that have high precision but low recall. So, recall is increased by automatically learning new patterns. By contrast, in Pantel and Pennacchiotti (2006), the starting point are patterns with high recall and low precision. The goal is to exploit these patterns by filtering incorrect related pairs using the Web. There are also interesting works using more supervised strategies for domain-specific corpora: in Aussenac-Gilles and Jacques (2006), it is described a method and a tool to manually define new specific patterns for specialized text corpora.

Recently, distant-supervision and self-supervised approaches take advantage of large amounts of freely available structured data, in order to automatically obtain training corpora to build extraction systems (Mintz et al., 2009; Hoffman et al., 2010).

Other works perform extraction in a different way. Open Information Extraction is a new paradigm that attempts to extract a large set of relational pairs with-

[1] `www.wikipedia.org` and `www.freebase.com`

out manually specifying semantic relations (Etzioni et al., 2008). *woe* is an Open Information Extraction method that takes advantage of the high quality semi-structures resources of Wikipedia (Wu and Weld, 2010). Finally, Bollegala's Relational Duality (Bollegala et al., 2010) applies a sequential co-clustering algorithm in order to cluster different LSP for extracting relations.

3 Motivation

The method presented in this paper follows a common statement which suggests that some linguistic constructs reliably convey the same type of knowledge, such as semantic or ontological relations (Aussenac-Gilles and Jacques, 2006; Aguado de Cea et al., 2009). Furthermore, it is based on the following assumption:

> *Semantic relations can be expressed in the same simple way as syntactic dependencies*

A semantic relation found in a sentence can be usually represented by a dependency link between two entities, even if there are items of extra information that can make the sentence very complex. This extra information does not express the target relation, but it may extend the meaning of the related entities or introduce knowledge not relevant for the relation. Among the most frequent patterns expressing relations, we can find variations of the same *original* pattern, which differ by the existence of modifiers, coordination, etc. Since these simple patterns have high precision, it is crucial to find a way of making them still more generic to increase coverage. For this purpose, we follow a two-step strategy:

1. Sentence compression: We use a partial grammar that establishes syntactic dependencies between items of extra information (modifiers, adjuncts, punctuation…). The grammar maintains only the dependency Heads and therefore allows us to obtain a sort of simplified linguistic structure.

2. Pattern extraction: We extract LSP, which are then simplified by means of a longest common string algorithm. These simplified patterns are transformed into generic semantic rules and added to our dependency grammar.

The combination of both standard dependency rules and generic semantic rules for RE allows the system to increase coverage without losing precision.

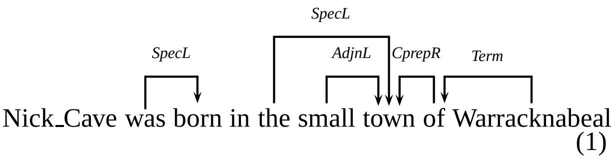
4 Partial Parsing for Sentence Compression

One of the main processes of our strategy attempts to simplify linguistic structures in order to easily extract their information. For this purpose, we use an open-source suite of multilingual syntactic analysis, DepPattern (Gamallo and González, 2011). The suite includes basic grammars for five languages as well as a compiler to build parsers from each one. The parser takes as input the output of a PoS-tagger, in our case, FreeLing (Padró et al., 2010), which also lemmatizes the sentences and performs Named Entity Recognition and Classification (NER/NEC).

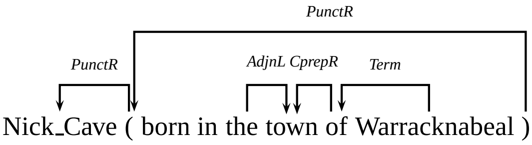
The basic grammars of DepPattern contain rules for many types of linguistic phenomena, from noun modification to more complex structures such as apposition or coordination. However, for our simplification task, only some types of dependencies are required, in particular those that compress the sentences maintaining their basic meaning. Following other strategies for sentence compression (Molina et al., 2010), we modified the default grammar by making use of rules that identify the following satellites and subordinate constituents:

- Punctuation (quotation marks, commas, brackets, etc.).
- Common noun and adjective coordination.
- Noun, Adverb, and Adjectival Phrases.
- Prepositional complements, verbal periphrasis and apposition.
- Negative sentences (where the verb inherits the negative tag).

After running the parser, all the Dependents identified by these rules are removed. That is, we obtain a compressed structure without satellites, modifiers, etc. In 1 and 2 we can see two examples of our partial parsing. The elements at the tail of the arrows are the Dependents, while those at the front of the arrows are the Heads.



(1)



(2)

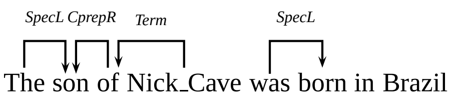
Taking into account that only the Heads (that are not Dependents) are maintained, the compression process on the two initial sentences will produce an unique simplified structure (note that the Heads of location phrases ("town of NP", "region of NP", etc.) inherit the location information provided by the dependent proper nouns, so in the examples, "town" represents a specific location):

<**Nick_Cave** *born in* **town**>

Generic semantic rules are then applied on these structures. For instance:

> if a personal name is the Head, a location noun is the Dependent, and the verb "to be born" is a Relator, then a `hasBirthplace` relation is identified.

This rule can be proposed to cover both the previous examples as well as many others. Moreover, our parsing also prevents from applying the previous extraction rule on sentences such as 3, where the Head of the first Noun Phrase is not the personal name, but a common noun ("son").



(3)

<**son** *born in* **Brazil**>

This way, in this type of sentences (or in negative ones, where the verb has a *negative* tag), our semantic rule will not extract the incorrect pair "Nick Cave `hasBirthplace` Brazil" (but we will be able to know the birthplace of "the son of Nick Cave").

The grammar formalism also allows the parser to maintain the Dependents of a rule after the rule application. Therefore, if we want to add several sets of rules for extracting various relations, the system will only need a single pass over the corpus.

In sum, the sentence compression performed by partial parsing simplifies the linguistic structures maintaining their basic information. This way, the addition of generic semantic rules (converted from LSP) at the end of a depedency grammar allows the parser to increase the coverage of the extraction.

5 Obtaining the Patterns and Rules

This section presents the distant-supervision method for extracting the lexico-syntactic patterns as well as the strategy for generating the generic rules.

5.1 Pattern Extraction

Following the assumption that most instances of a semantic relation are represented by similar LSP, we intend to obtain examples of those patterns and extract from them their *original* structures (without extra information), then transformed into semantic rules. In order to automate this process, we use the following strategy:

We get a large set of entity pairs of a desired relation from (semi)structured resources. For instance, for the `hasBirthplace` relation we get pairs from Wikipedia infoboxes (e.g., "Nick Cave - Warracknabeal", "Josep Guardiola - Sampedor", etc.). Note that the attributes of many relations are language-dependent (e.g., "Nick Cave" `hasProfession`: English: "singer/songwriter"; Spanish: "cantante/cantautor"; Portuguese: "cantor/cantautor", etc.), so the use of resources like Freebase is not always feasible. If we do not have a large amount of pairs, we manually introduce a small set of pairs regarding a particular relation.

These pairs are used to select from the unstructured text of Wikipedia sentences that contain both a named entity and an attribute of the relation. If the two terms match a known pair of the initial list, the example is annotated as positive. Otherwise, it is annotated as negative. Note that if we have a large set of pairs, the method does not require bootstrapping. However, If we only have a small set of initial pairs, a bootstrapping process is required (we use this strategy if the number of positive sentences is less than n, where n was empirically set to 200). Then, each selected sentence is tokenized, lemma-

Sentence: Nick Cave was born in the town of Warracknabeal.

Polarity: **Nick Cave** `hasBirthplace` **Warracknabeal**, true.

Pattern: <**X** be_V born_V in_PRP DA town_N of_PRP **Y**>

Figure 1: Example of a Sentence with the Polarity label of the related terms and its Pattern (V means verb, DT determiner, PRP preposition and N common noun).

tized and PoS-tagged. We also apply a NEC, in order to semantically classify the named entities.

Finally, the two target entities are replaced by both **X** and **Y**, standing for the first and the second entities of the pair, respectively. Only the context between the two entities are considered. To represent this context, we only take into account lemmas of verbs, common nouns and prepositions. We have observed in preliminary experiments that the performance of the patterns decreased when either these types of lemmas were removed or all lemmas including grammatical words (stop words), adjectives and proper names were retained. It follows that verbs, common nouns and prepositions are critical pieces of information to define the lexico-syntactic contexts of the target terms. Figure 1 contains an example of a pattern associated to the relation `hasBirthplace` (Table 1 also shows a set of extracted patterns in Portuguese).

All the process is performed without human revision. Note that this method may lead us to annotate *false positives* or *false negatives*. However, a manual evaluation on 1000 patterns show that this method has a precision of about 80%.

5.2 Pattern Generalization

We use the following method for making generic patterns, then transformed into high-precision rules:

1. First, we take all the patterns of type "**X**[...]**Y**" and select the most precise ones according to their confidence value. This value is obtained as follows: we calculate the positive and negative frequencies of each pattern; then we subtract the negative frequency from the positive, and sort the patterns by this value. Finally, the top *n* most confident patterns are selected (where $n = 20$ in our experiments). The same process is made for "**Y**[...]**X**" patterns.

2. Then, we apply a generalization algorithm for extracting the longest common string (*lcs*) from these patterns. In order to generalize two patterns, we check first if they are similar and then all those units that they do not share are removed. The similarity, noted $Dice\_lcs$, between two patterns p_1 and p_2 is defined using the longest common string and Dice metric as follows:

$$Dice\_lcs(p_1, p_2) = \frac{2 * lcs(p_1, p_2)}{length(p_1) + length(p_2)} \quad (4)$$

   where $lcs(p_1, p_2)$ is the size of the longest common string between patterns p_1 and p_2, and $length(p_i)$ represents the size of pattern p_i. It means the similarity between two patterns is a function of their longest common string and their lengths.

   After computing the similarity between two patterns p_1 and p_2, the *lcs* is extracted if and only if p_2 is the most similar pattern to p_1 and the similarity score is higher than a particular threshold (0.75 in our tests). The longest common string of two patterns is considered as the generalized pattern out of them.

3. We filter out those generalized patterns that are not in the best initial 20 patterns, so we automatically obtain a few set of very confident patterns (see Table 1 for an example).

4. All these generic patterns are added as blocks of rules into a grammar, which already has a set of dependency rules for text compression. The new semantic rules take the first entity **X** as the Head, and the second one **Y** as the Dependent of the relation. This process is made manually.

5. Finally, the grammar is compiled into a parser, which is applied on a corpus to obtain triples "**X** `relation` **Y**".

Table 1 shows an example of pattern generalization, with the best extracted patterns, the generic one automatically obtained as well an extraction rule.

Extracted Patterns: <**X** nascer_V em_PRP **Y**>,
<**X** nascer_V em_PRP a_DA cidade_N de_PRP **Y**>,
<**X** nascer_V em_PRP NP Fc **Y**>,
<**X** Fc_V nascer_V em_PRP **Y**>,
<**X** nascer_V CC residir_V em_PRP **Y**>,
[...]

Generic Pattern: <**X** nascer_V em_PRP **Y**>
Rule: **N**<**tp:P**> V<l:nascer> [P<l:em>] **N**<**tp:L**>

Table 1: Example of pattern generalization for the `hasBirthplace` relation in Portuguese (*nascer* means "to be born", *cidade* means "city" and *residir*, "to live").

In sum, the application of the longest common string algorithm on the best extracted patterns allows us to obtain a small set of high-quality rules in a weakly-supervised way. These rules, added at the end of a partial dependency grammar, extract instances of pairs belonging to the initial relation.

6 Experiments

We carried out three major experiments in order to know the performance of our RE method. First, we compared the rule-based approach to two baselines in a manually revised corpus containing examples of the relation `hasProfession` in Spanish. We also compared the performance of the system using a large amount of initial pairs (see Section 5.1) as well as with a small set of seed pairs.

Second, we applied a parser with the obtained extraction rules for the biographical relations `hasProfession` and `hasBirthplace` on the whole Spanish and Portuguese Wikipedias.

Finally, we applied the same Portuguese parser on a journalistic corpus, in order to know the performance on the system in different text genres.

6.1 Initial Data

We first obtained about $10,000$ pairs for each relation and language (Portuguese and Spanish) from the Wikipedia infoboxes. Then, we identified near $20,000$ sentences containing a personal name and (i) an occupation noun (`hasProfession`) or (ii) a location (`hasBirthplace`), which were automatically classified as positive or negative using the distant-supervision strategy described in Section 5.1. Finally, we randomly selected two sets of $2,000$ sentences for each relation and language as well as a small set of 200 for the relation `hasProfession`. The latter set was selected for evaluating the use of a small input.

For testing, we randomly selected $1,000$ sentences of `hasProfession` (different from the previous sets), which were manually revised.[2]

6.2 Results

Our first experiment evaluates the performance of the rule-based method compared to two baselines (in Spanish): *Baseline_1* performs a pattern-matching approach applying on the test set the whole positive sentences (except for the proper nouns, replaced by a PoS-tag) from the initial $2,000$ set. *Baseline_2* uses the $2,000$ initial sentences to train a Support Vector Machine classifier, representing each instance with the `token_TAG` elements as features. For this purpose, we used the WEKA implementation of the SMO algorithm (Witten and Frank, 2005).

To evaluate the rule-based system, we performed two experiments: the first one extracted the rules from the initial 200 sentences (*Rule_1*, with only 2 extraction rules) while the second one used the $2,000$ set of sentences (*Rule_2*, with 8 rules). The test only contains the 15 most frequent occupations found in the Wikipedia infoboxes, so the evaluation only takes into account the extraction containing these 15 nouns.

Table 2 shows the results of the four described methods over the test set. Precision is the number of correct positive decisions divided by the number of positive decisions (true and false positives). Recall is the number of correct positive decisions divided by the total of positive examples found in the test.

The pattern-matching baseline (*Baseline_1*) has a precision of 100%, but its f-score is merely 10% due to its low recall values. *Baseline_2* performs better, but it produces many false positives, so its precision values do not achieve 45%.

Both rule-based methods perform clearly better than the proposed baselines. *Rule_1*, with only two generic rules, achieves over 55% recall, maintaining the same precision as the pattern-matching models. The use of more data allowed us to add a set of 8 generic rules, so the *Rule_2* method increased its re-

[2] Both training and testing sets will be avaliable at `http://gramatica.usc.es/pln/`

Model	Precision	Recall	F-score
Baseline_1	100%	5.8%	10.1%
Baseline_2	44.51%	42.54%	43.5%
Rule_1	99.02%	55.8%	71.38%
Rule_2	99.16%	65.2%	**78.7%**

Table 2: Precision, Recall and F-score of the Baselines and the two rule-based models for the `hasProfession` relation in Spanish. Test set of $1,000$ sentences.

Language	Relation	Precision	Pairs
Spanish	`hasProf.`	85.35%	$241,323$
	`hasBirth.`	95.56%	$13,083$
Portuguese	`hasProf.`	93.86%	$17,281$
	`hasBirth.`	90.34%	$5,762$

Table 3: Precision and unique extracted pairs for each relation in the whole Spanish and Portuguese Wikipedias.

call in more than 10% without losing precision.

Since the test sentences used in these experiments were filtered with a small list of frequent occupation nouns, we performed other extractions in order to know the performance of our system in real text conditions. So we used the *Rule_2* method to parse the whole Spanish and Portuguese Wikipedias. For this purpose, we extracted seven `hasProfession` rules for Portuguese. Moreover, we add the `hasBirthplace` rules for each language obtained from the initial $2,000$ sets of this relation (four different rules were added for each language). Semantic information obtained from the NEC was used only in those `hasBirthplace` rules that did not have verb lemmas (such as *nacer/nascer*, "to be born").

Before evaluating the extraction in the whole corpora, we automatically remove some noise by eliminating tokens with less than 3 characters or with numbers. `hasProfession` pairs were filtered with the occupation nouns obtained from the Spanish and Portuguese Wikipedia infoboxes (about 500 and 250, respectively). To evaluate the `hasBirthplace` relation, the complete output of the extraction was used. We randomly selected and revised samples of 50 pairs from each rule, and calculate the weighted average of the extraction.

Table 3 shows the results of the two extractions over the Spanish and Portuguese Wikipedias. Only a single parsing was performed for each language (with both `hasProfession` and `hasBirthplace` extraction rules). Note that the corpora have about 3.2 and 1.8 gigabytes for Spanish and Portuguese, respectively.

In Spanish, almost $241,000$ unique pairs of `hasProfession` related entities were extracted, and more than $13,000$ different instances of `hasBirthplace`. Precision values for the first relation were worse than those obtained in the previous experiment (85% *vs* 99%). However, a deep evaluation of the errors shows that many of them were produced in previous processing steps (namely the identification of proper nouns), so the precision of these rules is likely to be better. `hasBirthplace` had better precision results (95%), but the amount of extracted pairs was noticeably lower.

In Portuguese, the system extracted about $17,000$ and $5,700$ `hasProfession` and `hasBirthplace` unique pairs, respectively. The differences between the Portuguese and the Spanish extractions have probably several reasons: on the one hand, the size of the Spanish corpus is almost the double. On the other hand, the number of occupation nouns used as a filter was also half in the Portuguese experiments. However, the extractions in Portuguese maintain high-precision values (90-93%).

Note that both `hasBirthplace` and `hasProfession` relations extract biographical data, so it is expected that encyclopedic resources such as Wikipedia contain many instances of these relations. Nevertheless, as we intend to perform extractions on texts of different genres, we applied the same Portuguese parser on a journalistic corpus from Público, a general-purpose Portuguese newspaper (with about 1.2 gigabytes).

In Table 4 we can see the results on the Público newspaper (evaluated in the same way as Wikipedia extractions). The first impression of these data is that the extraction doubles the number of instances with respect to the parsing of Wikipedia (which has a similar size). Precision values are between 6% and 9% lower, achieving 84% in both semantic relations. However, in a quick review of the extracted data, we also noted that many instances were incorrect due to the previous errors cited above.

Relation	Precision	Pairs
`hasProfession`	84.54%	$41,669$
`hasBirthplace`	84.67%	$11,842$

Table 4: Precision and unique extracted pairs for each relation in the Portuguese newspaper Público.

7 Conclusions

This paper presents a novel weakly-supervised approach for semantic Relation Extraction in different languages. We apply a sort of text compression strategy by means of partial dependency parsing which simplifies the linguistic structures, thus allowing the extraction rules to increase their coverage.

In order to (semi)automatically obtain these rules, we first extract lexico-syntactic patterns using a distant-supervision strategy. These patterns are generalized by a longest common string algorithm and finally transformed into semantic rules added at the end of a formal grammar.

Several experiments in different languages and corpora showed that this method maintains the high-precision values of pattern-matching techniques, while the recall is significantly improved.

In future work, we will carry out further experiments with other relations as well as in different corpora. Moreover, we will analyze the performance of the method with different Named Entity Classifiers, in order to avoid some noise during the extraction. Finally, we intend to take advantage of some anaphora and coreference resolution methods that might allow us to extract a large number of instances and to make a fusion process easier.

Acknowledgments

This work has been supported by the MICINN, within the project with reference FFI2010-14986.

References

Aguado de Cea, G., Gómez Pérez, A., Montiel-Ponsoda, E. and Suárez-Figueroa, M. C. 2009. Using Linguistic Patterns to Enhance Ontology Development. In: *Proceedings of KEOD*, 206–213.

Aussenac-Gilles, N. and Jacques, M.-P. 2006. Designing and Evaluating Patterns for Ontology Enrichment from Texts. In *Proceedings of EKAW*, 158–165.

Bollegala, D. T., Matsuo, Y. and Ishizuka, M. 2010. Relational duality: unsupervised extraction of semantic relations between entities on the web. In: *Proceedings of IW3C2*, 151–160.

Brin, S. 1998. Extracting patterns and relations from the world wide web. In: *WebDB Workshop at EDBT*, 172–183.

Chandrasekar, R., Doran, C. and Srinivas, B. 1996. Motivations and methods for text simplification. In: *Proceedings of COLING*, 2, 1041–1044.

Clarke, J. and Lapata, M. 2006. Models for Sentence Compression: A Comparison across Domains, Training Requirements and Evaluation Measures. In: *Proceedings of COLING/ACL*, 377–384.

Cohn, T. and Lapata, M. 2009. Sentence Compression as Tree Transduction. *Journal of Artificial Intelligence Research*, 34: 637–674.

Etzioni, O., Banko, M., Soderland, S. and Weld, D. S. 2008. Open Information Extraction from the Web. In: *ACM* 51, 12, 68–74.

Gamallo P. and González, I. 2011. A Grammatical Formalism Based on Patterns of Part-of-Speech Tags. *International Journal of Corpus Linguistics*, 16: 1, 45–71.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of COLING*, 2, 539–545.

Hoffmann, R., Zhang, C. and Weld, D.S. 2010. Learning 5000 Relational Extractors In: *Proceedigns of ACL*, 286–295.

Mintz, M., Bills, S., Snow, R. and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In: *Proceedings of ACL/IJCNLP*, 2.

Molina, A., da Cunha, I., Torres-Moreno, J-M. and Velazquez-Morales, P. 2010. La compresión de frases: un recurso para la optimización de resumen automático de documentos. *Linguamática*, 2: 3, 13–27.

Padró, Ll., Collado, M., Reese, S., Lloberes, M. and Castellón, I. 2010. FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In: *Proceedings of LREC*.

Pantel, P. and Pennacchiotti, M. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In: *Proceedings of COLING/ACL*, 113–120.

Ravichandran, D. and Hovy, E. 2002. Learning surface text patterns for a question answering system. In: *Proceedings of ACL*, 41–47.

Witten, I. H. and Frank, E. 2005. Data mining: practical machine learning tools and techniques with Java implementations. Elsevier Inc., San Francisco.

Wu, F. and Weld, D. 2010. Open information extraction using Wikipedia. In: *Proceedings of ACL*, 118–127.