A Weakly-Supervised Rule-Based Approach for Relation Extraction

Marcos Garcia and Pablo Gamallo *

Center for Research in Information Technologies (CITIUS)
University of Santiago de Compostela

Resumen Rule-based approaches for information extraction usually achieve good precision values, even if they often need a lot of manual effort to be implemented. In this paper, we present a novel rule-based strategy for semantic relation extraction that takes advantage of partial syntactic parsing in order to simplify the linguistic structures containing instances of semantic relations. We also propose a distant supervision strategy that automatically extracts generic lexico-syntactic patterns by means of semi-structured resources such as Wikipedia infoboxes. These generic patterns are then transformed into extraction rules that are used to update a partial dependency grammar. Several evaluations of this method show that it improves the recall while maintaining high-precision values. Experiments were performed on Spanish texts.

Keywords: information extraction, relation extraction, ontologies, lexico-syntactic patterns, text compression

1. Introduction

Information Extraction (IE) systems attempt to automatically obtain structured knowledge from unstructured text, such as the Web or other large corpora. Relation Extraction (RE) is a subtask of IE that aims to identify semantic relations between entities. For instance, from

the sentence "López Bouza nació en la localidad gallega de Ferrol" (*López Bouza was born in the Galician town of Ferrol*), a RE system may identify the birthplace of López Bouza (Ferrol).

The obtained data are then arranged to be incorporated into machine readable databases and ontologies which, in turn, are used to improve applications such as Question Answering engines or Information Retrieval systems.

Most of RE approaches need an initial set of sentences containing instances of a semantic relation. Such sentences are automatically obtained from seed entity pairs (e.g., "López Bouza `hasBirthplace` Ferrol"). The initial set of sentences is usually increased by applying bootstrapping strategies. These sentences provide a rich space with different linguistic knowledge (tokens, lemmas, PoS-tags, syntactic dependencies, etc.) used to build systems capable of extracting new instances of the semantic relations.

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

There are two main strategies for extracting relations: (i) machine learning approaches, which train classifiers by representing elements of the linguistic space as sets of features; (ii) pattern-matching techniques, which transform the linguistic space into lexico-syntactic patterns (LSP) applied on large corpora.

RE systems rely on the intuition that syntactic regularities (e.g., LSP) may characterize the same type of semantic information. However, one of the main problems of these strategies is that small variations in punctuation, adjective modification, etc., would prevent from finding appropriate patterns. For instance, the previous example pair could be contained in a great variety of sentences (these as well as the remainder examples of this paper will be in Spanish):

- "López Bouza nació en la localidad gallega de Ferrol"
- "López Bouza nació en Ferrol"
- "López Bouza nacé en el municipio de Ferrol"
- "López Bouza, nacido en la localidad coruñesa de Ferrol"

Both machine learning and pattern-matching techniques avoid this problem by (i) using larger sets of training patterns or (ii) applying parsers that identify the constituents of a sentence as well as their syntactic functions. However, obtaining large collections of high-quality training data is not always feasible, since a lot of manual effort is needed. Furthermore, parsers for other languages than English often perform very partial analysis, or are not freely available.

In this paper, we introduce a novel RE system that simplifies the linguistic structures by performing partial parsing. This simplification allows generic LSP to improve their recall in the extraction of semantically related pairs of entities.

The patterns are automatically obtained and generalized by means of a longest common string algorithm. Then, they are added as syntactico-semantic rules into a dependency grammar. We evaluate this method for two different semantic relations on a manually revised corpus as well as on the whole Spanish Wikipedia. The results show that the use of partial parsing allows the system to improve the extraction recall while maintaining precision.

This paper is organized as follows. Section 3 introduces some related work. Then, Section 3 describes our rule-based approach for RE. Section 3 shows the results of several tests and, finally, Section 5 reports the conclusions of our work.

2. Related Work

In recent years, many techniques have been applied in order to extract semantically related pairs. Supervised methods use high-quality linguistic data to characterize their training examples or to manually define extraction patterns. These methods achieve good results, but with a lot of effort. Other works take advantage of weakly-supervised strategies and bootstrapping, requiring only a small number of initial labeled training pairs or sentences. In this section, we briefly describe some weakly-supervised methods, focusing on pattern-based approaches.

Hearst was the first one to experiment a pattern-based strategy for the identification of semantic relations [6]. She made use of a small set of initial patterns to get hyperonymy

relations, increasing then the set of patterns with a bootstrapping technique. Other works make use of Question Answering pair examples to automatically extract patterns [13]. A novelty of this method lies in the application of a suffix tree, allowing the system to discover generalized patterns by calculating their common substrings. Another paper that carries out generalizations of patterns, by computing the edit distance, is [14], whose goal is the automatic expansion of WordNet.

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 [12], 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. More supervised strategies manually define specific patterns for specialized text corpora, such as [2].

More recent works perform extraction in a different way. Open IE is a new paradigm that attempts to extract a large set of relational pairs without manually specifying semantic relations [4]. *woe* is an Open IE method that takes advantage of the high quality semi-structures resources of Wikipedia [16].

Concerning text simplification, dependency rules and algorithms are used for simplifying complex sentences in order to easily access their information [3, 7].

3. The Method

In this section we will introduce the assumptions underlying our strategy as well as the Relation Extraction method itself.

3.1. Motivation

Our strategy follows a common statement which suggests that some linguistic constructs reliably convey the same type of knowledge, such as semantic or ontological relations [2, 1]. 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, etc.). 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 syntactic dependency rules and generic semantic rules for relation extraction allows the system to increase coverage without losing precision. In the remaining of this section, we describe the two tasks of our strategy: sentence compression by partial parsing, and pattern extraction.

3.2. Simplifying the Structure through Partial Parsing

In order to perform partial dependency parsing, we use an open-source suite of multilingual syntactic analysis, DepPattern [5]. 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 [11], which includes a lemmatizer, and a Named Entity Recognizer.

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 structure. Following other strategies for sentence compression [10], we modified the default grammar by making use of rules that identify satellites and subordinate constituents:

- Punctuation (quotation marks, commas, brackets, etc.).
- Noun and adjective coordination.
- Noun, Adverb, and Adjectival Phrases.
- Prepositional complements and verbal periphrasis.
- Apposition.

Then, all the Dependents identified by these rules are removed, so we obtain a compressed structure without satellites, modifiers, etc. In Examples 1 and 2 we can see two instances of our partial parsing. The elements at the tail of the arrows are the Dependents, while those at the front of the arrow are the Heads.

López Bouza nació en la localidad gallega de Ferrol. (1)

López Bouza nacíaa en el municipio de Ferrol. (2)

Taking into account that only the Heads (that are not Dependents) are maintained, the compression process will produce very similar simplified structures (note that the Heads of location structures inherit this tag, so in these examples “localidad” and “municipio” —synonyms for *town*— are location nouns):

<**López Bouza** *nació en* **localidad**.>
<**López Bouza** *nacía en* **municipio**.>

Then, we apply generic semantic rules on these structures, for instance:

> if a proper noun is the Head, a location noun is the Dependent, and the verb "nacer *en*" (*to be born* in) is a Relator, then a `hasBirthplace` relation is identified.

This rule can be proposed to cover both the previous simplified examples as well as many others, e.g., "López Bouza nació en Ferrol", etc. Let us note that our parsing simplification also prevents from applying the semantic rule defined above on sentences such as the Example 3, where the Head of the first Noun Phrase is "hermano" (*brother*) and not the proper noun.

El hermano de López Bouza nacÃa en Barcelona. (3)

<**hermano** *nacía en* **Barcelona**.>

This way, in this type of sentences (or in negative ones), our semantic rule will not extract the incorrect pair "López Bouza `hasBirthplace` Barcelona".

In sum, adding generic semantic rules (converted from LSP) at the end of a partial dependency grammar allows the system to get high-quality rules. Note that the coverage of these rules is much larger than the patterns themselves, since they take as input the structures previously simplified by dependency rules.

3.3. Obtaining the Patterns and Rules

Pattern Extraction: Following our assumption that most instances of a semantic relation are represented by similar LSP, our aim is to obtain examples of those patterns and then transform them into semantic rules. In order to automate this process, we use one of the following strategies:

1. If there are seed pairs of entities of the desired relation in (semi)structured resources, we use a distant supervision approach [9]: We get a large set of pairs from Wikipedia infoboxes. For instance, for the relation `hasBirthplace`, we get pairs such as "Fernando Pessoa - Lisboa", "Andrés Iniesta - Fuentealbilla", etc., with about 95 % of precision.
2. If we do not have a large amount of pairs for a particular relation, we manually introduce a small set of pairs of this relation relation.

We use these pairs to select sentences that contain both a named entity and an location from the free text of Wikipedia. If the two terms match a known pair of the list, the example is annotated as positive. Otherwise, it is negative. If we use the second strategy, a bootstrapping process will be required if the number of positive sentences is less than n (where n was empirically set to 200).

Each selected sentence is tokenized, lemmatized, and PoS-tagged. Finally, the two target entities are replaced by both **X** and **Y** (the first and the second entities of the pair, respectively). Only the context between the two entities are considered. 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, 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.

This process is performed automatically, so it may lead us to annotate false positives or false negatives. However, the revision of test sets showed that this method has between 80 % and 92 % of precision, depending on the relation.

Figure 1 contains an example of a pattern for the relation `hasBirthplace`.

Sentence: Andrés Iniesta nació en la localidad de Fuentealbilla.
Polarity: "Andrés Iniesta `hasBirthplace` Fuentealbilla", **true**.
Pattern: <**X** nacer_V en_PRP DA localidad_N de_PRP **Y**>

Figura 1. Example of a Sentence with its Polarity label and its Pattern (V means verb, DA article, PRP preposition and N common noun).

Pattern Generalization: In order to make more generic patterns which are transformed into high-precision rules, we use the following method:

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 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 [14]. 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 *lcs* 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 of p_1 and the similarity score is higher than a particular threshold (0.75 in our tests). The *lcs* of two patterns is considered as the generalized pattern out of them.

3. We filtered out those patterns that are not in the best initial 20 patterns. This strategy allows us to obtain a few set of very confident patterns.
4. The simplified patterns are added as blocks of rules into a grammar, which already has a set of standard dependency rules. The new semantic rules take the first entity **X** as the Head, and the second one **Y** as the Dependent of the relation. If two different rules only differ in one or two tokens, the grammar formalism allows us to declare these tokens as optional in the rules, so we can merge two rules into a new one. This last task is made manually.
5. Finally, the grammar is compiled into a parser, which is applied on a corpus to obtain triples "**X** `relation` **Y**".

Extracted Patterns: <**X** nacer_V en_PRP **Y**>, <**X** haber_V nacer_V en_PRP **Y**>
<**X** nacer_V en_PRP el_DA ciudad_N de_PRP **Y**>, <**X** nacer_V en_PRP NP Fc **Y**>,
<**X** nacer_V en_PRP W en_PRP **Y**>, <**X** nacer_V CC residir_V en_PRP **Y**>,
<**X** Fc CS haber_V nacer@V CC crecer_V en_PRP **Y**>, <**X** Fc nacer_V en_PRP **Y**>
Generic Pattern: <**X** nacer_V en_PRP **Y**>

Cuadro 1. Example of pattern generalization for the `hasBirthplace` relation in Spanish.

Table 1 shows an example of pattern generalization, with the best extracted patterns as well as the generic one obtained by means of the *lcs* algorithm. This pattern will be transformed into a semantic rule, which will be used for RE.

In sum, the application of the *lcs* algorithm on the best patterns allows us to obtain a small set of high-coverage rules in a weakly-supervised way. In the following section, the results of several test using this method are analyzed.

4. Experiments

We carried out two major experiments in order to know the performance of our RE method in real text. First, we compared the rule-based approach to two baselines in a manually revised corpus containing examples of the relation `hasProfession`. We also compared the two strategies of learning rules described in Section 3.3 (with a large amount of sentences as well as using a small set). Second, we apply a parser with automatically obtained rules for `hasProfession` and `hasBirthplace` relations in the whole Spanish Wikipedia (May 2010).

In order to extract the sentences containing the related entities, we first obtain about $10,000$ pairs for each relation from the Spanish Wikipedia infoboxes. Then, we identified near $20,000$ sentences containing a named entity as an occupation noun (`hasProfession`) or a location (`hasBirthplace`), automatically classified as positive or negative. Finally, we randomly selected two sets of $2,000$ sentences for each relation 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. All the sets have a ratio of 50 %/50 % of positive and negative examples.

For testing, we randomly selected 1,000 sentences of the `hasProfession` relation (different from the previous sets), and manually revised their classification.[1]

4.1. Results

Our first experiment evaluates the performance of the rule-based method compared to two baselines: *Baseline_1* performs a pattern-matching approach using the tokens of the positive sentences from the initial 2,000 set (except for the proper nouns, where the lemmas were replaced by a PoS-tag). *Baseline_2* uses the 2,000 initial sentences to train a Support Vector Machine classifier, using the `token_TAG` elements as features. For this purpose, we used the WEKA implementation of the SMO algorithm [15].

To evaluate the rule-based approaches, we extracted the best patterns from the initial 200 sentences (*Rule_1*, with only 2 extraction rules) as well as from the set of 2,000 sentences (*Rule_2*, with 8 rules). The test set only contains the 15 most frequent occupations found in the Spanish Wikipedia infoboxes, so the evaluation only takes into account the extraction containing the same 15 nouns.

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 %**

Cuadro 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.

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 set.

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 similar precision values than the pattern-matching models. *Rule_2* (with eight rules) increased its recall in about 10 % without losing precision. This metrics are not easily comparable to other systems, since we do not know similar experiments in Spanish. However, some of the results reported in [8] suggest that our method performs similar than other state-of-the-art systems for this kind of extraction.

In order to know the performance of our system in real text conditions, we used the *Rule_2* method to parse the whole Spanish Wikipedia. For this purpose, we also add the `hasBirthplace` rules obtained from the initial 2,000 set of this relation. The

[1] Training and testing sets will be available at `http://gramatica.usc.es/pln/`

extraction rule method produced four different basic rules from this collection. However, due to its similarity, they were unified into two rules. Note that only a single parsing was performed (with both `hasProfession` and `hasBirthplace` extraction rules). Before evaluating the extraction in the whole corpus, we automatically remove some noise by eliminating tokens with less than three characters or with numbers. We also filtered the `hasProfession` pairs with about 500 occupation nouns obtained from the Spanish Wikipedia infoboxes.

		Generic Rules							
Relation		1	2	3	4	5	6	7	8
`hasProfession`	Prec.	75.51 %	100 %	86 %	93.48 %	46.34 %	68.89 %	86.96 %	84.44 %
	Pairs	100,117	35,107	231,512	58,477	1,063	9,271	34,727	18,635
`hasBirthplace`	Prec.	95.45 %	97.78 %						
	Pairs	12,400	683						

Cuadro 3. Precision and unique extracted pairs for rule in the whole Spanish Wikipedia. Rule numbers correspond to their frequency position in the extraction from the initial 2,000 patterns, so `hasProfession` rules 1 and 2 are those used in the *Rule_1* model.

In order to obtain the precision values, we randomly extracted and revised samples of 50 pairs from each extraction rule (Table 3). `hasProfession` rules extracted about 535,000 pairs (250,000 unique). As we can see, there are noticeable variation in the performance of each extraction rule, namely in terms of quantity. In spite of this, note that most rules have a high precision (except rules 5 and 6, which also have low coverage), so the weighted average is of 85.35 %. Besides, we have to say that most errors were produced by previous steps of the analysis (namely the tokenizer and the Named Entity Recognizer). On the other hand, `hasBirthplace` rules extracted almost 13,500 unique pairs, with very high precision values (weighted average of 95.56 %).

5. Conclusions

In this paper we introduced a novel rule-based approach for Relation Extraction, which requires little manual effort. We follow the assumption that some linguistic structures convey the same kind of knowledge, such as semantic relations.

In order to simplify/compress many linguistic structures belonging to the same semantic relation, we apply partial dependency parsing focused on the identification of satellite constituents. Then, we automatically extract a set of LSP using a distant supervision strategy. These patterns are simplified/generalized by means of a longest common string algorithm, transformed into extraction rules and, finally, added to a formal grammar.

The performed experiments showed that this method maintains the high-precision values of pattern-matching strategies. In addition, due to the increase in recall, the overall performance significantly improves the extraction.

In future work, we will carry out further experiments with other relations as well as in different languages. Moreover, we will analyze the performance of the system with

different Named Entity Recognizers and Classifiers, in order to avoid some noise in the extraction.

Referencias

1. Aguado de Cea, G., Gómez Pérez, A., Montiel-Ponsoda, E., Suárez-Figueroa, M. C.: Using Linguistic Patterns to Enhance Ontology Development. In: KEOD (2009).
2. Aussenac-Gilles, N. and Jacques, M.-P.: Designing and Evaluating Patterns for Ontology Enrichment from Texts. In Proceedings of EKAW 2006, pp. 158–165 (2006).
3. Chandrasekar, R., Doran, C., Srinivas, B.: Motivations and methods for text simplification. In: Proceedings of COLING, 2, pp. 1041–1044 (1996).
4. Etzioni, O., Banko, M., Soderland, S., Weld, D. S.: Open Information Extraction from the Web. In: ACM 51, 12, pp. 68–74 (2008).
5. Gamallo P., González, I.: A Grammatical Formalism Based on Patterns of Part-of-Speech Tags. International Journal of Corpus Linguistics, 16: 1, 45–71 (2011).
6. Hearst, M. A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of COLING, 2, pp. 539–545 (1992).
7. Klebanov, B., Knight, K., Marcu, D.: Text Simplification for Information-Seeking Applications. In: On the Move to Meaningful Internet Systems. LNCS. 3290, pp. 735–747 (2004).
8. Mann, G., Yarowsky, D.: Multi-Field Information Extraction and Cross-Document Fusion. In: Proceedings of ACL, pp. 483–490 (2005).
9. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of ACL/IJCNLP (2009).
10. Molina, A., da Cunha, I., Torres-Moreno, J-M., Velazquez-Morales, P.: La compresión de frases: un recurso para la optimización de resumen automático de documentos. Linguamática, 2: 3, 13–27 (2010).
11. Padró, Ll., Collado, M., Reese, S., Lloberes, M., Castellón, I.: FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In: Proceedings of LREC. La Valletta, Malta (2010).
12. Pantel, P. and Pennacchiotti, M.: Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In: Proceedings of COLING/ACL, pp. 113–120 (2006).
13. Ravichandran, D. and Hovy, E.: Learning surface text patterns for a question answering system. In: Proceedings of ACL, pp. 41–47 (2002).
14. Ruiz-Casado, M., Alfonseca, E., and Castells, P.: Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets. In: Proceedings of AWIC, pp. 380–386 (2005)
15. Witten, I. H. and Frank, E.: Data mining: practical machine learning tools and techniques with Java implementations. Elsevier, San Francisco (2005).
16. Wu, F., Weld, D.: Open information extraction using Wikipedia. In: Proceedings of ACL, pp. 118–127, Stroudsburg (2010).