Improving Relation Extraction by Using an Ontology Class Hierarchy Feature

Pedro H.R. Assis[1(✉)], Marco A. Casanova[1], Alberto H.F. Laender[2], and Ruy Milidiu[1]

[1] Department of Informatics, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brazil
{passis,casanova,milidiu}@inf.puc-rio.br
[2] Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil
laender@dcc.ufmg.br

Abstract. Relation extraction is a key step to address the problem of structuring natural language text. This paper proposes a new ontology class hierarchy feature to improve relation extraction when applying a method based on the distant supervision approach. It argues in favour of the expressiveness of the feature, in multi-class perceptrons, by experimentally showing its effectiveness when compared with combinations of (regular) lexical features.

Keywords: Relation extraction · Distant supervision · Semantic Web · Machine learning · Natural language processing

1 Introduction

A considerable fraction of the information available on the Web is under the form of natural language, unstructured text. While this format suits human consumption, it is not convenient for data analysis algorithms, which calls for methods and tools to structure natural language text. Among the many key problems this task poses, *relation extraction*, i.e., the problem of finding relationships among entities present in a natural language sentence, stands out.

The most successful approaches to address the relation extraction problem apply supervised machine learning to construct classifiers using features extracted from hand-labeled sentences of a training corpus [5,10]. However, supervised methods suffer from several problems, such as the limited number of examples in the training corpus, due to the expensive cost of manually annotating sentences. Such limitations hinder their use in the context of Web-scale knowledge bases. Distant supervision, an alternative paradigm introduced by Mintz et al. [9], addresses the problem of creating examples, in sufficient number, by automatically generating training data with the help of a sample database.

J. Wang et al. (Eds.): WISE 2015, Part II, LNCS 9419, pp. 241–249, 2015.
DOI: 10.1007/978-3-319-26187-4_20

In this paper, we first discuss how to apply the distant supervision approach to develop a multi-class perceptron[1] for relation extraction. Then, we present new *semantic features*, defined based on a pair of entities e_1 and e_2 identified in the sentence. The semantic features associate classes C_1 and C_2 to the sentence, where C_1 and C_2 are derived from the class hierarchy of an ontology and the original classes of e_1 and e_2 in the hierarchy. The main contribution of the paper is the proposal of these semantic features.

Finally, we describe experiments to evaluate the effectiveness of our semantic based features, using a corpus extracted from the English Wikipedia and instances of the DBpedia Ontology. We conducted two types of experiments, adopting the automatic held-out evaluation strategy and human evaluation. In the held-out evaluation experiments, the multi-class perceptron identified, with an F-measure greater than 70 %, a total of 88 relations out of the 480 relations featured in the version of the DBpedia adopted. In the human evaluation experiments, it achieved an average accuracy greater than 70 % for 9 out of the top 10 relations, in the number of instances, selected for manual labeling. An early and short version of these results appeared in [2].

This paper is structured as follows. Section 2 discusses related work. Section 3 describes the approach adopted to construct multi-class perceptron for relation extraction and the definition of the ontology classes hierarchy feature. Section 4 contains the experimental results. Finally, Sect. 5 presents the conclusions and suggestions for future work.

2 Related Work

Soderland et al. [11] introduced supervised-learning methods as approaches for information extraction. They are the most precise methods for relation extraction [5,10], but they are not scalable to the Web due to the expensive cost of production and the dependency on an annotated corpus for the specific application domain. In order to address the scalability problem in relation extraction frameworks, weak supervision methods were introduced, based on the idea of using a database with structured data to heuristically label a text corpus [4,13,14].

Mintz et al. [9] coined the term distant supervision to replace the term weak supervision. They applied Freebase facts to create relation extractors from Wikipedia, achieving an average precision of approximately 67.6 % for the top 100 relations. The popularity of distant supervision methods increased rapidly since its introduction. Unfortunately, depending on the domain of the relation database and the text corpus, heuristics can lead to noisy data and poor extraction performance.

Finally, classifiers can be improved with the help of Semantic Web resources and, conversely, new Semantic Web resources can be generated by using relation extraction classifiers. For example, Gerber et al. [6] used DBpedia as background knowledge to generate several thousands of new facts in DBpedia from Wikipedia

[1] *Perceptron* is a linear classifier for supervised machine learning. It is an assembly of linear-discriminant representations in which learning is based on error-correction.

articles, using distant supervision methods. For relation extraction they used a pattern matching approach. In this work, instead of relying on the generation of relation patterns, we used DBpedia as background knowledge to generate an annotated dataset to construct a multi-class perceptron for relation extraction.

3 The Distant Supervision Approach

We transform the relation extraction problem into a classification problem by treating each relation r as a class $\mathbf{r}$ of a multi-class perceptron. To construct the perceptron, we feed a machine learning algorithm with sentences in a corpus C, together with their feature vectors, where the sentences are heuristically annotated with relations using the distant supervision approach. In this paper, we adopt a non-memory-based machine learning method, called Multinomial Logistic Regression [8], which computes a multi-class perceptron. This section covers the major points of the approach, referring the reader to [1] for the full details.

3.1 Distant Supervision

The approach we adopt to generate a dataset is based on distant supervision [9]. The main assumption is that a sentence might express a relation if it contains two entities that participate in that relation.

Formally, given an ontology O, we say that e_i is an entity *defined in* O iff there is a triple of the form $(e_i, \text{rdf:type}, K_i)$ in O such that K_i is a class in the vocabulary of O. The *relation database* of O is the set R_O such that a triple $(e_1, r_i, e_2) \in O$ is in R_O iff e_1 and e_2 are entities defined in O and r_i is an object property in the vocabulary of O. For example, if *"Barack Obama"* and *"United States"* are entities in O and there is a triple $t =$ (*"Barack Obama", "president of", "United States"*), then $t \in R_O$.

Let C be a corpus of sentences each of which is annotated with two entities defined in O. Suppose that a sentence $s \in C$ is annotated with entities e_1 and e_2 and that there is a triple (e_1, r, e_2) in R_O. Then, we consider that s is *heuristically labeled* as an example of the relation r. For example, suppose that R_O contains the triple: *(Led Zeppelin, genre, Rock Music)*, where the rock band *Led Zeppelin* and the music genre *Rock Music* are defined in O. Then, every sentence annotated with *Led Zeppelin* and *Rock Music* is a prospective example of the relation *genre*, such as: "***Led Zeppelin*** *is a british rock band that plays* ***rock music***."

The approach is applicable for inverse relations if they are explicitly declared in the ontology O. They will be simply treated as new classes.

3.2 Features

We associate a feature vector with each sentence s in the corpus C. Feature vectors will have dimension 12, comprising 10 lexical features, as in [9], and two features based on the class structure of the ontology O.

For *lexical features*, let s be a sentence in a corpus C annotated with two entities e_1 and e_2. We break s into five components, $(w_l, e_1, w_m, e_2, w_r)$, where w_l comprehends the subsentence to the left of the entity e_1, w_m the subsentence between the entities e_1 and e_2 and w_r the subsentence to the right of e_2. For example, the sentence s_A "*Her most famous temple, the* ***Parthenon***, *on the Acropolis in* **Athens** *takes its name from that title.*" is represented as ("Her most famous temple, the", **Parthenon**, ", on the Acropolis in", **Athens**, " takes its name from that title."). Lexical features contemplate the sequence of words in w_l, w_m, and w_r and their part-of-speech; but not all the words in w_l and w_r are used. Indeed, let $w_l(1)$ and $w_l(2)$ denote the first and the first two rightmost words in w_l, respectively. Analogously, let $w_r(1)$ and $w_r(2)$ denote the first and the first two leftmost words in w_r, respectively. In the example, the corresponding sequences of length 1 and 2 are: $w_l(1)$ = "the", $w_l(2)$ = "temple, the", $w_r(1)$ = "takes" and $w_r(2)$ = "takes its". The part-of-speech tags cover 9 lexical categories: NOUN, VERB, ADVERB, PREPosition, ADJective, NUMbers, FOReign words, POSSessive ending and everything ELSE (including articles).

For *class-based features*, we propose to use as a feature of an entity e (and of the sentences where it occurs) the class that best represents e in the class structure of the ontology O. We claim that the chosen class must not be too general, since we want to avoid losing the specificities of the semantics of e that are not shared with the other entities of the superclasses. On the other hand, a class that is too specific is also not a good choice. Very specific classes restrict the accuracy of classifiers, since they probably contain fewer entities than more general classes. In other words, the number of entities in a class is likely to be inversely proportional to the class specificity.

Therefore, we propose to use as a feature of an entity e (and of the sentences where it occurs) the class associated with e that intuitively lies in the mid-level of the ontology class structure. For example, suppose we have the entity *Barack Obama*, with class hierarchy *President* $\subset$ *Politician* $\subset$ *Office_holder* $\subset$ *Person* $\subset$ *Agent* $\subset$ *owl:Thing*. We have to choose one class to represent the entity *Barack Obama*. If we choose the class *Agent*, for example, which is too general, all relations involving a president will be assign to every example of agents in our dataset, which therefore not a good choice. On the other hand, if we choose the class *President*, which is too specific, we will be missing several relations shared by politicians or office holders. Therefore, we choose the class at the middle level of the hierarchy, which in this example is *Office_holder*.

More precisely, given an ontology O, the *class structure* of O is the directed graph $G_O = (V_O, E_O)$ such that V_O is the set of classes defined in O and there is an edge $< C, D >$ in E_O iff there is a triple *(C, owl:SubClassOf, D)* in O. We assume that G_O is acyclic and that G_O has a single sink, the class *owl:Thing*. This assumption is consistent with the usual practice of constructing ontologies and the definition of *owl:Thing*. By analogy with trees, the *height* of G_O is the length of the longest path from a source of G_O to *owl:Thing* and the *level* of a class C in G_O is the length of the shortest path in G_O from C to *owl:Thing*.

We also assume that O is equipped with a service that, given an entity e, classifies e into a single class C_e. Assume that the shortest path in G_O from C_e to *owl:Thing* is $(C_k, \cdots, C_i, \cdots, C_0)$, where $C_k = C_e$ and $C_0 = owl{:}Thing$. Then, we define the *class-based feature* of e as the class C_i, where $i = min(k, h/2)$, where h is the height of G_O. Note that we take the minimum of k and $h/2$ since the level of C_k may be smaller than half of the height of G_O.

Finally, let s be a sentence in the corpus C, annotated with two entities e_1 and e_2. We define the *class-based features* of s as the class-based features of e_1 and e_2.

4 Experiments

We adopted a version of DBpedia [3] as our ontology, which features 359 classes, organized into hierarchies, 2,350,000 instances and more than 480 different relations. We used all Wikipedia articles in English as a source of unstructured text. We annotated a Wikipedia article A with an entity e from DBpedia if there is a link in the text of A pointing to the article corresponding to e. For sentence boundary detection, we used the algorithm proposed by Gillick [7]. We also applied heuristics in order to increase the number of acceptable sentences. We annotated references to the main subject of an article by string matching between the article text and the article title. Also, for sentences with more than two instances annotated, we considered combinations of all pairs of instances.

Applying all strategies described above, we generated a corpus of 2,276,647 sentences with annotated entities, for which we obtained lexical and class-based features as described in Sects. 3.2 and 4. We used the Stanford Part of Speech Tagger [12] and the WSJ 0.18 Bidirectional model for POS features to extract the lexical features, but we simplified the POS tags into 9 categories, as already indicated in Sect. 3.2.

4.1 Held-Out Evaluation

We ran experiments to assess the impact of the class-based features by training the Multinomial Logistic Regression classifier [8] using only lexical features, only class-based features and both sets of features. Half of the sentences for each relation were randomly chosen not to be used in the training step. They are later used in the testing step.

For this kind of extraction task, final users usually consider an acceptable performance if it predicts classes with an F-measure greater than 70 %. Therefore, the comparison between the various options took into account the number of classes for which the perceptron achieved an F-measure greater than 70 %. Table 1 show the top 10 classes for each combination of features, with the classes identified by their suffixes, since they all share the same prefix in their URI: http://dbpedia.org/ontology. Also, Table 1 shows that class-based features were able to predict over 6 times more classes than our baseline (lexical features only)

Table 1. Top 10 classes for a perceptron trained with different feature set.

Features	No.	Class	Precision	Recall	F-measure
Lexical	1	/targetSpaceStation	1.00	1.00	1.00
	2	/department	0.98	0.86	0.92
	3	/discoverer	1.00	0.81	0.90
	4	/militaryBranch	0.94	0.83	0.88
	5	/notableWine	0.99	0.75	0.85
	6	/programmeFormat	0.87	0.77	0.82
	7	/type	0.69	0.83	0.75
	8	/license	0.98	0.58	0.73
	9	/sport	0.81	0.63	0.71
	10	/composer	0.95	0.54	0.69
		average:	**0.921**	**0.760**	**0.825**
		number of classes > 70% F-measure:		**6**	
Class-based	1	/areaOfSearch	1.00	0.98	0.99
	2	/ground	0.96	1.00	0.98
	3	/mission	0.97	1.00	0.98
	4	/politicalPartyInLegislature	1.00	0.95	0.97
	5	/precursor	0.99	0.96	0.97
	6	/sport	0.96	0.97	0.97
	7	/targetSpaceStation	0.94	1.00	0.97
	8	/discoverer	0.93	1.00	0.96
	9	/drainsTo	0.97	0.93	0.95
	10	/isPartOfAnatomicalStructure	0.91	1.00	0.95
		average:	**0.963**	**0.979**	**0.969**
		number of classes > 70% F-measure:		**60**	
Lexical and Class-based	1	/areaOfSearch	1.00	0.97	0.98
	2	/ground	0.97	1.00	0.98
	3	/mission	0.99	0.96	0.97
	4	/sport	0.97	0.97	0.97
	5	/targetSpaceStation	1.00	0.93	0.97
	6	/academicDiscipline	0.93	0.99	0.96
	7	/discoverer	0.99	0.93	0.96
	8	/locatedInArea	0.93	0.98	0.96
	9	/programmeFormat	0.93	0.99	0.96
	10	/politicalPartyInLegislature	1.00	0.91	0.95
		average:	**0.971**	**0.963**	**0.966**
		number of classes > 70% F-measure:		**88**	

and the inclusion of lexical features can improve the previous result in 32 %, predicting a total of 88 classes with more than 70 % of F-measure.

Although, in general, there is a considerable gain by using both sets of features, the perceptron trained using both sets of features had a worse performance than that trained using only class-based features for some classes. For example, `/aircraftFighter` is identified with a F-measure of 50 % using both sets of features, whereas it was identified with 77 % using only class-based features.

Table 2. Average accuracy for the top 10 relations in examples in our dataset for human evaluation of a sample of 100 predictions.

Relation	Number of instances	Average accuracy
http://dbpedia.org/ontology/country	607,380	73 %
http://dbpedia.org/ontology/family	159,717	75 %
http://dbpedia.org/ontology/isPartOf	139,694	90 %
http://dbpedia.org/ontology/birthPlace	138,797	76 %
http://dbpedia.org/ontology/genre	109,813	77 %
http://dbpedia.org/ontology/location	96,516	76 %
http://dbpedia.org/ontology/type	72,942	80 %
http://dbpedia.org/ontology/order	53,421	81 %
http://dbpedia.org/ontology/occupation	48,859	87 %
http://dbpedia.org/ontology/hometown	34,010	68 %

This shows that for some classes, our lexical features reduces the generalization of our model of classification, but overall they increase the robustness of predictions for the majority of classes.

4.2 Human Evaluation

For the human evaluation experiments, we also separated the sentences, annotated with pairs of entities, into training and testing data. We randomly chose half of the sentences not to be used in the training step, for each relation (in this section we again use the term "relation" instead of "class"). For each of the top 10 relations (in the number of instances in our dataset), we extracted random samples of 100 sentences from the remaining sentences and forwarded to two evaluators to manually label the sentences with relations. Finally, we compared the manually labeled sentences with the labeling obtained by a perceptron trained using both lexical and class-based features, as shown in Table 2, where the average accuracy is percentage of the sentences that the automatic labeling coincided with the manual labeling, for each relation. Note that the average accuracy ranged from 90 % for http://dbpedia.org/ontology/isPartOf to 68 % for http://dbpedia.org/ontology/hometown.

5 Conclusions

In this paper, we introduced a feature defined by ontology class hierarchies to improve relation extraction methods based on the distant supervision approach.

To demonstrate the effectiveness of class-based features, we presented experiments involving articles in the English Wikipedia and triples from DBpedia. We first heuristically labeled a corpus of sentences with relations, using the distant supervision method. We then used the class-based features, combined with

common lexical features adopted for relation extraction, to train a multi-class perceptron. The held-out experiments demonstrated a substantial gain in how many relations could be identified (with an F-measure greater than 70 %), when the class-based features are adopted. We also conducted a human evaluation experiment to further assess the accuracy of the perceptron.

As future work, we plan to explore how sensitive the perceptrons are to the choice of the classes that annotate a sentence and define our semantic feature. Also, we intend to extend the feature vector extracted from sentences by adding more lexical features, such as dependencies path. Finally, we intend to improve the annotation of self-links (match between the article text and its title) by using co-reference resolution, synonyms, pronouns, etc.

Acknowledgments. This work was partly funded by CNPq, under grants 312138/2013-0 and 303332/2013-1, and by FAPERJ, under grant E-26/201.337 /2014.

References

1. Assis, P.H.R.: Distant supervision for relation extraction using ontology class hierarchy-based features. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro (2014)
2. Assis, P.H.R., Casanova, M.: Distant supervision for relation extraction using ontology class hierarchy-based features. In: Poster and Demo Track of the 11th Extended Semantic Web Conference (2014)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: a nucleus for a web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
4. Craven, M., Kumlien, J.: Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (1999)
5. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 363–370 (2005)
6. Gerber, D., Ngonga Ngomo, A.C.: Bootstrapping the linked data web. In: Proceedings of the 1st Workshop on Web Scale Knowledge Extraction, ISWC 2011 (2011)
7. Gillick, D.: Sentence boundary detection and the problem with the U.S. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Short Papers), pp. 241–244 (2009)
8. McCullagh, P., Nelder, J.A.: Generalized Linear Models (1989)
9. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: vol. 2, pp. 1003–1011. ACL (2009)

10. Nguyen, T.D., yen Kan, M.: Keyphrase extraction in scientific publications. In: Proceedings of International Conference on Asian Digital Libraries, pp. 317–326 (2007)
11. Soderland, S.: Learning information extraction rules for semi-structured and free text. Mach. Learn. **34**, 233–272 (1999)
12. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 63–70 (2000)
13. Wu, F., Weld, D.S.: Autonomously semantifying wikipedia. In: Proceedings of the 16th ACM Conference on Information and Knowledge Management, pp. 41–50 (2007)
14. Wu, F., Weld, D.S.: Automatically refining the wikipedia infobox ontology. In: Proceedings of the 17th International World Wide Web Conference (2008)