Distant Supervision for Relation Extraction using Ontology Class Hierarchy-Based Features

Pedro H. R. Assis and Marco A. Casanova

Department of Informatics, PUC-Rio
Rio de Janeiro/RJ, Brazil
{passis,casanova}@inf.puc-rio.br

Abstract. Relation extraction is a key step in the problem of structuring natural language text. This paper demonstrates a multi-class classifier for relation extraction, constructed using the distant supervision approach, along with resources of the Semantic Web. In particular, the classifier uses a feature based on the class hierarchy of an ontology that, in conjunction with basic lexical features, improves accuracy and recall. The paper contains extensive experiments, using a corpus extracted from the Wikipedia and the DBpedia ontology, to demonstrate the usefulness of the new feature.

Keywords: relation extraction, distant supervision, Semantic Web, machine learning, natural language processing

1 Introduction

A large amount of the data on the Web is stored in natural language format or unstructured text. While this format provides information targeting towards human consumption, several algorithms for data analysis are not applicable since they require structured data.

In order to render a structure from natural language text, a key problem is relation extraction, namely, the problem of finding relationships between entities present in the text. The most successful approaches to the relation extraction problem apply supervised machine learning to compute classifiers using features extracted from hand-labeled sentences comprising a training corpus [1,2,4]. However, supervised methods create several problems, such as the limited number of examples in the training corpus, due to expensive cost of production, and the domain dependency on corpus annotations. Such limitations prevent using supervised machine learning to construct web-scale knowledge bases.

An alternative paradigm for relation extraction was introduced in [5]. The distant supervision approach addresses the problem of creating a considerable number of examples by automatically generating training data from heuristically matching a database relation to text. Recent approaches to relation extraction use resources of the Semantic Web to improve accuracy of the classifiers and, conversely, to generate new Semantic Web resources [3].

In this paper, we demonstrate a multi-class classifier for relation extraction, constructed using the distant supervision approach, along with resources of the Semantic Web. In particular, the classifier uses a feature based on the class hierarchy of an ontology that, in conjunction with basic lexical features, improves accuracy and recall.

We conducted two types of experiments, adopting the automatic held-out evaluation strategy and human evaluation (we recall that the term *held-out evaluation* refers to experiments where part of the data is held out for testing and the rest of the data is used to train a classifier). In the held-out evaluation experiments, the multi-class classifier identified a total of 88 relations, out of the 480 relations featured in the version of the DBPedia adopted, with an F-measure greater than 70%, whereas 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.

2 Heuristic Labeling, Lexical Features and Class-based Features

Heuristic Labeling. Let O be an ontology, defined as a set of RDF triples. We define a subset $T \subseteq O$ such that $t_i = (e_1, r_i, e_2) \in T$ iff r_i is an object property of O and there are triples $(e_1, \text{rdf:type}, K_1)$ and $(e_2, \text{rdf:type}, K_2)$ in O, where K_1 and K_2 are classes of O.

Let C be a corpus of n sentences and assume that each sentence is annotated with two entities defined in O. A sentence s is *heuristically labeled* with a relation r_i iff s is annotated with entities e_1 and e_2 and $(e_1, r_i, e_2) \in T$. For example, suppose that the triple *(Led Zeppelin, genre, Heavy Metal Music)* is in T and assume that both entities are instances of classes of the ontology O. Consider the sentence *"**Led Zeppelin** is a british rock band that plays **heavy metal music**"*, where the text in boldface are annotated with references to the entities "Led Zeppelin" and "Heavy Metal Music". Then, we label this sentence as an example of the relation *genre*.

Lexical Features. Each labeled sentence in C is described by a 12-dimension *feature vector*. Out of the 12 dimensions, 10 are *lexical* and 2 are *class-based*, defined in this and the next subsections.

Let s be a sentence and divide s into 5 components $s = (w_l, e_1, w_m, e_2, w_r)$, where w_l comprehends the subsentence to the left of the entity e_1, w_m represents the subsentence between the entities e_1 and e_2 and w_r comprehends the subsentence to the right of e_2. The *lexical features* of s contemplate the sequence of words in w_l, w_m, and w_r. Not all words in w_l and w_r are used, though. In fact, let $w_l(1)$ and $w_l(2)$ denote the first and the first and the second rightmost words in w_l, respectively, and let $w_r(1)$ and $w_r(2)$ denote the first and the first and the second leftmost words in w_r, respectively. Table 1 defines the 10 lexical features adopted and illustrates them with the sentence

s_A = "*Her most famous temple, the* ***Parthenon****, on the Acropolis in* ***Athens*** *takes its name from that title.*"

Table 1. Lexical features and examples

Dimension	Description	Example from s_A
f_1	The sequence of words of w_m	", on the Acropolis in"
f_2	Part-of-speech tags of w_m	PREP ELSE NOUN PREP
f_3	The sequence of words of $w_l(1)$	"the"
f_4	Part-of-speech tags of $w_l(1)$	ELSE
f_5	The sequence of words of $w_l(2)$	"temple, the"
f_6	Part-of-speech tags of $w_l(2)$	NOUN ELSE
f_7	The sequence of words of $w_r(1)$	"takes"
f_8	Part-of-speech tags of $w_r(1)$	VERB
f_9	The sequence of words of $w_r(2)$	"takes its"
f_{10}	Part-of-speech tags of $w_r(2)$	VERB ELSE

Class-based Feature. One of the main contributions of this paper is to use as a feature of an entity e the class that best represents e in the class hierarchy of an ontology. The chosen class must not be too general, in a sense that we want to avoid loosing specificities of the semantics of e that are not shared with other entities that belong to the upper classes. On the other hand, a class which is too specific is not a good choice as well. Very specific classes restrict the accuracy of classifiers since there are more entities for a more general class. Therefore, we propose to use as a feature for e the class associate with e that intuitively lies in the mid-level of the tree.

More precisely, let H be a tree representing an ontology class hierarchy and assume that h is the height of H. Let C_k be the class of entity e that the entity annotation tool returns (we assume that the tool returns only one class). Assume that the path in H from the root to C_k is $C_0, ...C_i, ..., C_k$. Then, the *class-based feature* of entity e is the class C_i, where $i = min(k, h/2)$. Note that we take the minimum of $h/2$ and k since the level of C_k may be smaller than half of the height h of H.

3 Experiments

In this work, we adopted DBpedia as our source of relation instances and the English Wikipedia as a source of unstructured text. We created the annotations of the sentences extracted from Wikipedia by matching links to others articles, occurring in the text, to entities in DBpedia, discarding any imprecision in our results due to ambiguity on entity recognition. We selected only sentences in Wikipedia that contained at least two annotations, thereby generating a corpus of nearly 2.2 million sentences. From these annotated sentences, we extracted feature vectors that were used as input to a Logistic Regression classifier.

We conducted held-out evaluation experiments and human evaluation experiments. Recall that held-out evaluation refers to experiments where part of the data is held out for testing and the remaining is used for training a classifier. We ran held-out experiments with classifiers constructed using only lexical features, only class-based features and both sets of features to measure the impact of the class-based feature proposed in this work. We compared the performance of the classifiers thus obtained by counting the number of classes each classifier identified with F-measure greater than 70%.

We considered as baseline the number of classes with F-measure greater than 70% that the classifier trained only with lexical features identified. In our experiments, such classifier identified 9 classes. The classifier trained using only the class-based feature proposed in this work identified a total of 60 classes with F-measure greater than 70%. The classifier trained using both lexical and class-based features identified a total of 88 classes, again with F-measure greater than 70%. Compared to the baseline, it achieved an almost 10-fold increase in the number of classes identified with F-measure greater than 70%. Table 2 shows the top 10 classes.

Table 2. Top 10 relations for a classifier trained with lexical and class-based features.

Class	Precision	Recall	F-measure
/areaOfSearch	1.00	0.97	0.98
/ground	0.97	1.00	0.98
/mission	0.99	0.96	0.97
/sport	0.97	0.97	0.97
/targetSpaceStation	1.00	0.93	0.97
/academicDiscipline	0.93	0.99	0.96
/discoverer	0.99	0.93	0.96
/locatedInArea	0.93	0.98	0.96
/programmeFormat	0.93	0.99	0.96
/politicalPartyInLegislature	1.00	0.91	0.95

For the human evaluation, we extracted random samples of 100 sentences for each of the top 10 relations in the number of examples in our dataset. Those samples were forwarded to two evaluators. Table 3 shows the accuracy of each prediction of the samples, carried out manually.

Table 3. Average accuracy for the top 10 relation in examples in our dataset for human evaluation of a sample of 100 predictions.

Relation	Accuracy
http:/dbpedia.org/ontology/country	0.73%
http:/dbpedia.org/ontology/family	0.75%
http:/dbpedia.org/ontology/isPartOf	0.90%
http:/dbpedia.org/ontology/birthPlace	0.76%
http:/dbpedia.org/ontology/genre	0.77%
http:/dbpedia.org/ontology/location	0.76%
http:/dbpedia.org/ontology/type	0.80%
http:/dbpedia.org/ontology/order	0.81%
http:/dbpedia.org/ontology/occupation	0.87%
http:/dbpedia.org/ontology/hometown	0.68%

4 Demonstration

To demonstrate the multi-class classifier for relation extraction, we created a tool that accepts a sentence, annotated with the URIs of two DBpedia instances, extracts all features described in Section 2, generates a feature vector that is used as an input to the classifier, and returns a relation between the two instances. A demonstration video can be watched at
`https://www.youtube.com/watch?v=jwMXkHeUwhM`

References

1. Curran, J., Clark, S.: Language independent NER using a maximum entropy tagger. Proc. 7th Conf. Nat. Lang. Vol. 4, 164–167 (2003)
2. Finkel, J., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. Proc. 43rd An. Meet. Assoc. Comp. Ling., 363–370, (2005)
3. Gerber, D., Ngomo, A.-C.: (2011) Bootstrapping the Linked Data Web, 1st Work. Web Scale Know. Extraction ISWC (2011)
4. Nguyen, T., Kan, M.: Keyphrase Extraction. Scientific Publications, 317–326 (2007).
5. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data Proc. Joint Conf. 47th An. Meet. ACL and 4th Int. Joint Conf. Nat. Lang. Proc. AFNLP: Vol 2, 1003–1011 (2009)