3. The Relation Extraction Problem

This chapter first defines the problem of relation extraction and then present
the solution proposed. Section 4.1 gives a formal description of the approach
applied in this dissertation. Then, Section 4.2 states the formal definition of the
problem discussed in this dissertation. Finally, Section 4.3 discusses the features
used to build the multi-class classifier for relation extraction.

3.1. Approach

The approach we propose is based on that used by Mintz et al. (2009). This
approach, called distant supervision, is a heuristic that automatically labels the
relation between annotated entities on sentences based on a database relation. In
this dissertation, the heuristically labeled dataset is used for the computation of
relation extractors.

Our approach uses a dataset $T$ containing triples $t_i = (e_1, r_i, e_2)$, where $e_1$ and
$e_2$ are entities and $r_i$ is a relation. In our architecture, this dataset is extracted from
Semantic Web resources. Consider an ontology $O$ defined by a set of triples. We
define the set $T \subseteq O$ such that a triple $t_i = (e_1, r_i, e_2)$ is in $T$ iff $r_i$ is an object
property and $(e_1, \text{rdf:type}, K_1)$ and $(e_2, \text{rdf:type}, K_2)$ are triples in $O$, where $K_1$ and
$K_2$ are classes of $O$. In our approach, we use triples from $T$ to heuristically label
sentences based on the annotated entities.

Consider a text corpus $C = (s_1, ... , s_n)$ constituted by $n$ sentences. We also
consider that every sentence is annotated with two entities defined in $O$. Suppose
that a sentence $s_i$ is annotated with entities $e_1$ and $e_2$. If there is a triple $(e_1, r_i, e_2)$ in
$T$, the sentence $s_i$ is heuristically labeled as an example of the relation $r_i$. 
Then, a feature vector $v_j$ is computed using $s_j$. This feature vector is used to train a multi-class classifier using a supervised machine learning algorithm. In this case, $v_j$ is an instance of the class $p$.

For example, consider the following triple of the relation genre in our triple set $T$:

$(\text{Led Zeppelin, genre, Heavy Metal Music})$

that indicates that the rock band “Led Zeppelin” plays music of the genre “heavy metal”. Assume that both entities are instances of classes of the same ontology $O$.

Using this approach, every sentence that mentions “Led Zeppelin” and “heavy metal music” is a prospective example of the relation genre. In this example, consider the sentence:

“Led Zeppelin is a british rock band that plays heavy metal music.”

This sentence is annotated referencing the entities “Led Zeppelin” and “Heavy Metal Music”. Heuristically, we label this sentence as an example of the relation genre. As discussed in Section 4.3, we can extract features from this sentence to define a feature vector that composes an instance of the class genre for a supervised learning algorithm.

We are now able to give a formal definition of the relation extraction task used in this work.

### 3.2. Extraction Task Definition

Let $C = (s_1, ..., s_n)$ be a corpus containing $n$ sentences of unstructured text, each sentence annotated with two named entities, defined in an ontology $O$. Also, let $T$ be defined as in Section 4.1. For each sentence $s_j$ in $C$ with named entities $e_1$ and $e_2$ such that there is a triple $t_j = (e_1, r_j, e_2)$ in $T$, heuristically label $s_j$ as an example of the relation $r_j$. Compute a classifier $f$, using heuristically labeled sentences as a training set, such that it takes as an input a new sentence $s$ annotated with two named entities and returns the relation between such instances expressed in $s$. 
3.3. Features

This section describes the dimensions of the feature vector \( v_j \) extracted from a sentence \( s_j \) from a corpus \( C \). We assume that each sentence from \( C \) is annotated with two entities \( e_1 \) and \( e_2 \). Feature vectors will have dimension 12 (10 lexical features and 2 features based on the ontology hierarchical class tree) and will feed a supervised machine learning algorithm in order to calculate a multi-class classifier \( f \). This section is the main contribution of this dissertation, which is the proposal of a feature that is based on the class hierarchy of an ontology.

3.3.1. Natural Language Processing Based Features

For each sentence \( s_j \) from a textual corpus \( C = (s_1, \ldots, s_n) \), as defined in the previous section, we define several lexical features. Most of them were explored in Mintz et al. (2009).

Let \( s \) be a sentence annotated with two entities \( e_1 \) and \( e_2 \). We break \( s \) into five 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 \( e_2 \). For example, consider the following sentence:

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

According to the representation given above, this sentence can be represented as:

\[
s = ("Her most famous temple, the ", \text{Parthenon}, ", on the Acropolis in ", \text{Athens}, ", takes its name from that title."
\]

Lexical features contemplate the sequence of words in \( w_l \), \( w_m \), and \( w_r \), 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 second rightmost words in \( w_l \), respectively. Analogously, let \( w_r(1) \) and \( w_r(2) \) denote the first and second leftmost words in \( w_r \), respectively. In the example, the corresponding sequences of length 1 and 2 are:

\[
\begin{align*}
w_l(1) & = "the" \quad \text{and} \quad w_l(2) = "temple, the" \\
w_r(1) & = "takes" \quad \text{and} \quad w_r(2) = "takes its"
\end{align*}
\]
We define 10 dimensions based on lexical features, as listed in Table 1, which also includes examples using the sentence \( s \) above.

### Table 1. Lexical features and examples

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
<th>Example in ( s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>The sequence of words of ( w )</td>
<td>“, on the Acropolis in”</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>Part-of-speech tags of ( w )</td>
<td>ELSE CONPREP ADVERB NOUN CONPREP</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>The sequence of words of ( w )</td>
<td>“the”</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>Part-of-speech tags of ( w )</td>
<td>ADVERB</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>The sequence of words of ( w )</td>
<td>“temple, the”</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>Part-of-speech tags of ( w )</td>
<td>NOUN ELSE ADVERB</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>The sequence of words of ( w )</td>
<td>“takes”</td>
</tr>
<tr>
<td>( f_8 )</td>
<td>Part-of-speech tags of ( w )</td>
<td>VERB</td>
</tr>
<tr>
<td>( f_9 )</td>
<td>The sequence of words of ( w )</td>
<td>“takes its”</td>
</tr>
<tr>
<td>( f_{10} )</td>
<td>Part-of-speech tags of ( w )</td>
<td>VERB NOUN</td>
</tr>
</tbody>
</table>

#### 3.3.2. Ontology Class Hierarchy Based Feature

One of the main contributions of this dissertation is to use as a feature of an entity \( e \) the class that best represents \( e \) in the class hierarchy of an ontology. We claim that the chosen class must not be too general, in a sense that we want to avoid losing specificities of the semantics of \( e \) that are not shared with other entities of upper classes. For example, choosing the class \( \text{Person} \) for the entity \( \text{Barack Obama} \) is not a good choice since it is most likely that relations involving that entity are more specific for subclass like \( \text{Politician} \).

On the other hand, a class which is too specific is also not a good choice. Very specific classes in this sense restrict the accuracy of classifiers since less entities than a more general class. In other words, the number of entities is inversely proportional to the class specificity. Using the same example, if the entity \( \text{Barack Obama} \) is associated with the class \( \text{President} \), then we will probably
associate *Barack Obama* with a smaller number of relations than if *Barack Obama* is associated with the class *Politician*, for example, which clearly has more entities generating more examples of similar relations. Therefore, we propose to use as a feature for an entity $e$ the class associated with $e$ that intuitively lies in the mid-level of the ontology class hierarchy.

More precisely, we assume that the topology of the classes of $O$ is a tree $H$, which implies that there must be a single root node or class that is a superclass of all classes. In fact, this node is always represented by the class *owl:Thing*.

Assume that $h$ is the height of $H$. Let $C_k$ be the class of entity $e$ that the annotation returns. Assume that the path in $H$ from the root to $C_k$ is $C_0, \ldots, C_i, \ldots, C_k$. Then, we take as a feature of entity $e$ the class $C_i$ where $i = \min(k, h/2)$. Note that we must take the minimum of $h/2$ and $k$ since the level of $C_k$ may be smaller than half of the height of $H$.

Consider the example in Figure 16 which depicts a hierarchy tree from a subset the classes from DBpedia\(^7\). Considering only the classes in the tree depicted in Figure 16, we have that the longest is from the root “owl:Thing” to the class “Eurovision Song Contest Entry”. A cut in tree at the height $h/2$ is showed in Figure 17 which represents the subset of classes that can be chosen for entities. According to the cut, an instance of the class “Eurovision Song Contest Entry” will be represented by the class “Musical Work”, an instance of the class “Language” will be represented by “Language”, since it is a leaf in the cut tree, and an instance is defined only as an instance of the class “Work” will be represented by the class “Work”.

\(^7\) [http://dbpedia.org](http://dbpedia.org)
Consider the sentence \( s \) used as example in Section 3.3.1 with annotated entities \( e_1 = \text{Parthenon} \) and \( e_2 = \text{Athens} \). The features based on the ontology class hierarchy extracted from \( s \) is showed in Table 2.

\[
\begin{align*}
\text{s} &= \text{“Her most famous temple, the Parthenon, on the Acropolis in Athens takes its name from that title.”}
\end{align*}
\]

Table 2. Features of \( s \) based on Ontology Class Based Hierarchy Features

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
<th>Example in ( s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{11} )</td>
<td>Hierarchy Class based features of</td>
<td><a href="http://dbpedia.org/ontology/Building">http://dbpedia.org/ontology/Building</a></td>
</tr>
<tr>
<td>( f_{12} )</td>
<td>Hierarchy Class based features of</td>
<td><a href="http://dbpedia.org/ontology/AdministrativeRegion">http://dbpedia.org/ontology/AdministrativeRegion</a></td>
</tr>
</tbody>
</table>
3.4. Summary

This chapter presented the formal definition of the problem discussed in this dissertation and describes our approach to the relation extraction problem. In particular, it covered the features extracted from resources used by our approach, classified into lexical features and ontology class hierarchy based features. The last one is one of the main contributions of this work.