Final Report
A SVM Model for Relation Classification of Noun Phrases
based on the NELL Database

Shen Chen Xu
Computer Science Department
Carnegie Mellon University
shenchex@cs.cmu.edu

Xiaohua Yan
Language Technologies Institute
Carnegie Mellon University
xiaohuay@cs.cmu.edu

Wei Zhang
Language Technologies Institute
Carnegie Mellon University
weizhan1@cs.cmu.edu

Abstract

The Never Ending Language Learning (NELL) project has produced considerable corpus statistics
describing subject-verb-object triples, as well as instances of hundreds of relations. This paper
presents RECSVM, a relation classification model for noun phrase (NP) pairs based on the NELL
knowledge base, which is capable of predicting whether a given pair of noun phrases satisfies a
certain relation in NELL, but it is also capable of determining the most probable relation given the
input NP pair. Features employed to build the SVM model include categories of NPs, distributions
of word occurrences, contexts of NPs as well as results of shallow parsing, which are proved to be
effective in classifying the relations of noun phrase pairs.

1 Introduction and Motivation

Open Information extraction (IE) is the process of generating relational data from natural language texts without
requiring a pre-specified vocabulary. Particularly, Open IE systems have achieved notable success in massive, open-
domain corpora such as the Web and Wikipedia [1]-[3]. The output of Open IE systems can be utilized to support
tasks such as learning selectional preferences [4] and acquiring common sense knowledge [5]-[6]. For example, the
NELL (Never Ending Language Learning) research project is a computer system that operates 24 hours per day, for
years, to continuously extract facts from the web [6]. NELL is given as input an initial ontology that specifies the
semantic relations (e.g. animalEatFood(animal,food)) it must extract from the web. New instances of these categories
(e.g. city (“Pittsburgh”)) and relations (e.g. teamPlaysInCity (“Steelers”,“Pittsburgh”)) are extracted by NELL using a
large-scale multitask, semi-supervised learning method. Since NELL started its operation in January 2010, it has built
up a database containing over 700,000 instances with a precision of approximately 0.85.

A particularly interesting problem is that, given a new NP pair, how we can determine the relations that the NP pair
satisfies. For instance, given a pair of noun phrases such as (cat, fish), or (cat, house), the relation classifier should
predict whether the NP pair satisfies the relation “animalEatFood”. Unfortunately, there is not such a prediction
functionality in the architecture of NELL.

The above problem can be regarded as a variation of the supervised relation extraction problem, which has been
investigated by many researchers [7]-[9]. Relation extraction, typically involved in annotating the unstructured text
with entities as well as relations between entities, can be formulated as a classification problem in a discriminative
framework. Given a set of training examples labeled positive or negative, a general classifier for the input sentence
$S = w_1 w_2 \ldots e_1 \ldots e_2 \ldots w_{n-1} w_n$ can be illustrated as

$$f_R(T(S)) = \begin{cases} +1, & \text{if } e_1 \text{ and } e_2 \text{ are related by } R \\ -1, & \text{otherwise} \end{cases}$$
where \( f_R(\cdot) \) can be a discriminative classifier and \( T(S) \) can be a set of features extracted from the sentence. Figure 1 shows the general architecture of feature-based supervised approaches for relation extraction. Though these approaches perform well for pre-defined relations and are of relatively low computational complexity, the major limitation of them lies in the fact that it is difficult to extend the trained classifiers to new relation types due to the lack of labeled data, and also the manually defined relations cannot be canonicalized. However, harnessing the power of large-scale knowledge bases such as NELL, supervised approaches become promising again with the availability of huge amounts of extracted relations and labeled instances. In the setting of our problem, the inputs of the classifier would be NP pairs rather than sentences, and hence it is challenging to extract features that effectively characterize the NP pairs.

Figure 1: General architecture of feature-based supervised relation extraction

This paper presents \textsc{RecSvm}, a SVM-based relation classification model, which is not only able to predict whether a given pair of noun phrases satisfies a certain relation, but it is also capable of producing the most probable relation given the input NP pair. Namely, given the NP pair (e.g. \(<\text{cat}, \text{fish}>\)), our classifier will output a positive answer to the relation (e.g. “animalEatFood”). Moreover, it is also able to output the most probable relation that \(<\text{cat}, \text{fish}>\) satisfies. The major contribution of our work is the proposed multi-class SVM model that is derived from individually trained binary classifiers for each relation, which achieves an overall accuracy of 0.87 in the case study of classifying three selected relations in NELL’s database.

The rest of this paper is organized as follows. Section 2 explains in more detail about the data we used and the definitions of proposed features that characterize noun phrases. Section 3 presents the experiment results on three selected relations. An in-depth analysis of the proposed method and comparison of performance of proposed feature sets are presented in Section 4. Finally, several conclusions are made in Section 5 along with expectations for future work.

2 Methods

In this section, we explain the methods used in developing the classification model and describe in more detail the data we used, from which we extracted 4 distinct features that characterize relations in different aspects. The approach of individually training binary classifiers for each relation as well as a combined multi-class SVM model, \textsc{RecSvm}, is also introduced.

2.1 Data

The data we used mainly come from the NELL database [10] and ClueWeb09 [11], covering information about different noun-phrases, the relations in between, and the contexts (patterns) that help feature these noun phrases. The format of each data type is described as follows.

2.1.1 SVO triples

The subject-verb-object (SVO) triples data are constructed by parsing 50m Web documents from ClueWeb09 (890m sentences, 16B tokens) using the MALT dependency parser, and then extracting SVO triples from these parsed sentences. According to the SVO triples, it is easy to extract and count the hits of individual NP pairs (e.g. \(<\text{cat, fish}>, 300\), co-occurrence of all the \(<\text{individual NP, context}> \) pairs (e.g. \(<\text{fish, is eaten}>, 200\)) as well as \(<\text{NP pair, context} \) pairs (e.g. \(<\text{fish, cat, is eaten by}>, 100\).
2.1.2 NELL’s beliefs

This type of data describes the beliefs learned by the NELL project. We tailored the data downloaded from the NELL database to meet our needs, each entry of which is of the following format,

\[ \text{<entity relation value>} \]

Namely, entity here denotes the subject of a SVO triplet while value being the object, and relation describes the relation given the noun phrases. For example, a valid entry in our tailored data would be

\[ \text{<book:a_device_of_death bookwriter writer:christopher_bulis>} \]

2.2 Features

2.2.1 Category feature

Based on NELL’s beliefs, the categories of NPs in the pair are major indicators that describe certain relations, and it is not hard to observe that NP pairs that satisfy the same relation can fall into different categories. For instance, \(<\text{female:beyonce actor:starred_in_movie movie:cadillac_records}>\) and \(<\text{actor:will_smith actor:starred_in_movie movie:independence_day}>\) are both valid entries for the relation \(<\text{actor:starred_in_movie}>\) even though the category of beyonce is not actor. Hence, we define the category feature \(F_1\) of a training relation to be an indicator vector of all the possible categories that each noun phrase of the NP pair falls into. For example, given the pair \(<\text{beyonce,cadillac_records}>\), the dimensions that correspond to words such as female, singer and movie would be marked as 1 in the \(F_1\) vector, as shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>female</th>
<th>male</th>
<th>singer</th>
<th>movies</th>
<th>writer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>(0)</td>
<td>(1)</td>
<td>(1)</td>
<td>(0)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Category feature of the NP pair \(<\text{beyonce,cadillac_records}>\)

The effectiveness of the category feature is discussed in Section 4, which turns out to be the most powerful one conditional on the situation that the categories of both items of the NP pair can be found in the NELL database. However, this is not necessarily the case. Whenever there is no matching category found for a certain NP, the category feature may lose its power as a result. To address this problem, we turned to other ontologies, such as WordNet, as a complementary source that maintains a broader coverage concerning the categories of noun phrases. Nevertheless, the results still might suffer if we use the categories in WordNet indiscriminately. The performance of the WordNet ontology for our task is discussed in Section 4. Also, the limitations of the category feature lead us to think of other possible features independent on the category definitions.

2.2.2 Directed correlation of NP pairs

To classify a given NP pair whose relation label is unknown, it is intuitive to first determine how “related” the NPs are. We define the “directed correlation” (DC) feature of NP pairs as a measure of . Here by “directed” we want to discriminate the order of the NPs appeared in the pair to make sure the direction of subject-verb-object is not reversed. Assume \(P(NP_1)\) and \(P(NP_2)\) to be the probabilities of the occurrence of each noun phrase, and \(P(NP_1, NP_2)\) be the probability of their co-occurrence. The directed correlation of the NP pair \(<NP_1, NP_2>\) can be denoted as

\[ DC(NP_1, NP_2) = \frac{P(NP_1, NP_2)}{P(NP_1)P(NP_2)} \]

It is clear that the value of \(DC\) would be equivalent to 1 if \(NP_1\) is independent from \(NP_2\). For the purpose of avoiding the denominator to be zero, we can write the smoothed \(DC\) as
$DC(NP_1, NP_2) = \frac{P(NP_1, NP_2) + \frac{1}{M}}{P(NP_1)P(NP_2) + \frac{1}{N^2}}$

where $M$ denotes the total number of NP pairs while $N$ being the total number of individual noun phrases. In order to adapt this feature to fit our learning algorithm, the value of this feature is computed by

$$F_2 = \begin{cases} [00], & DC \leq 1 - \delta \\ [01], & 1 - \delta < DC \leq 1 + \delta \\ [10], & DC > 1 + \delta \end{cases}$$

where the optimal value of $\delta$ is learned from the data. The definition of DC allows the possibility that DC alone is not confident enough to tell whether the NP pair is correlated, which means other features may be required to further classify it. In the meantime, $F_2$ consistently keeps the features to be 0-1 vectors, which is crucial to reduce the computational complexity of our final model.

### 2.2.3 Context feature

As stated in Section 1, the classic supervised approaches for relation extraction obtained effective features from sentences where the NPs appear. Namely, the context serves as an informative feature that describes NP pairs [12]. However, in our task the context in which either or both items of the NP pair occur is not directly accessible. Instead, we made use of the counts of <individual NP, contexts> and <NP pair, contexts> pairs to augment the feature vector in the following way: as long as a certain verb or noun appears in the contexts related to either or both items of the NP pair, the corresponding dimension of the feature vector would be marked as 1. In other words, the corresponding feature $F_3$ would be a fairly long vector where the value of 1 indicates the existence of corresponding verbs or nouns ever appeared in related contexts (patterns) given an individual NP or a NP pair. For example, given the NP pair <beyonce, cadillac_records>, we would look at all the contexts that are associated with cat, fish or both. The dimensions corresponding to verbs and nouns such as star and movie would hopefully be marked as 1 in the feature vector. Figure 2 visualizes the process of extracting the context feature for the pair <beyonce, cadillac_records>.

![Figure 2: Process of extracting the context feature for <beyonce, cadillac_records>](image)

Also worth mentioning is that in order to fit the $F_3$ vector, all the contents in contexts would be pre-processed by stemming. Moreover, regarding to the order of noun phrases in the pair, we can incorporate position information in this feature by making distinctions on the words adjacent to $NP_1$ and $NP_2$. Consider the example of “$NP_1$ acted in the new movie $NP_2$”. The words “acted” and “movie” are given a special flag “_” to denote their contiguity with the NPs, and the resulting feature vector is displayed in Table 2.

<table>
<thead>
<tr>
<th>. act in the movie</th>
<th>movie sing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
An alternative way of defining the context feature of NP pairs is to only extract the related verbs in the context as for a certain relation. In other words, we can get a “bag of verbs” rather than a “bag of words” for each relation. The performances of these two definition schemes are compared in Section 4.

2.2.4 Shallow parsing feature

The POS (Part of Speech) feature of relations for training is also employed to help characterize the relations. To be more specific, as for a training relation (e.g. actorStarredInMovie), the POS tags along with the positions of each POS are obtained and accordingly comprise the feature vector $F_3$ by marking the corresponding dimensions as 1. Similar with $F_3$, we also give special flags to POS tags that are contiguous with $NP_1$ and $NP_2$. An example of how we extract the shallow parsing feature is presented in Figure 3.

![Figure 3: Process of extracting the shallow parsing feature for “NP_1 acted in the movie NP_2”](image)

After we have obtained the feature vectors $F_1, ..., F_4$, the next step would be to evaluate the performance of each feature as well as the combinations of them, which is carried out in Section 4. The final input vector of our proposed model would be the concatenations of the relation label and the optimal combination of features. In other words, each row of the input will be of the form $LF_1F_2F_3F_4$, where $L$ denotes the label of the training relation.

2.3 The REC SVM model

So far, we have described the ways of extracting and digitalizing different features (all of them are transformed into 0-1 vectors) from the data to be incorporated into our classifier. SVM is employed to perform the task of predicting whether a NP pair satisfy a certain relation as well as outputting the most probable relation it satisfies. Generally our classification model can be divided into two stages:

1. According to the relation definitions available from NELL, target relations and labeled instances are selected for training. Given the training dataset $D$ for relation $R_j$, which is defined by

   $D = \{(x_i^{R_j}, y_i^{R_j}) | x_i^{R_j} \in B^p, y_i^{R_j} \in \{-1, +1\}\}

   where $B^p$ denotes the $p$-dimensional 0-1 feature space spanned by $F_1, F_2, F_3$ and $F_4$. A binary SVM classifier, $C_j$, is trained for the given relation $R_j$. Assume in total there are $M$ relations selected for training, the resulting $M$ binary classifiers would be capable of predicting whether the test point $X$ satisfies the relation $R_j$, which is our first goal, by feeding $X$ to the $C_j$ classifier.

2. In this stage, each training example is first used as a test point to be fed into each of the $M$ classifiers, and the resulting 0-1 vectors along with the original label of the training example is utilized to constitute a new training example. In this way, the produced $M$ binary classifiers are combined are combined to produce new inputs for non-parametric classification algorithms such as K nearest neighbors. Figure 4 illustrates the process of generating the final classification model.

In the next section, experiments are carried out on the proposed model, whose final prediction is based on the nearest neighbor(s). That is, given a new test point, we feed it to each of the binary classifiers and use the resulting vector (which is all +1 and -1) to find the nearest neighbor(s) based on the new training examples.

3 Experiment

For the purpose of testing the feasibility of the proposed method, and at the same time evaluating the performance of individual features and their combinations, experiments were performed on 2600 training samples with 807 testing samples (each is of the format <subject, object>), both of which are all labeled into 3 relations that are
denoted by <actorstarredinmovie>, <directordirectedmovie>, <bookwriter> respectively. We chose these 3 relations because the first and second relations are quite similar while the third one is less relevant with the previous 2 relations. Using the RECSVM model to output the most probable relation(s) for each testing example, classification accuracy for each relation is displayed in Table 3.

4 Results and discussions

4.1 NELL vs. Wordnet: effectiveness of F₁

As can be observed from Table 3, the category feature based on NELL’s ontology achieves much higher than that based on WordNet. It is not surprising that the NELL-based F₁ feature achieves classification accuracy as high as 98.2% (for the relation actorStarredInMovie), because all the NP pairs used as training and testing samples are extracted from NELL’s database, and they will both have corresponding categories pre-defined for the feature vector. Nevertheless, if given a input NP pair there are no matching categories for each item, the category feature based on NELL alone will be almost useless. On the other hand, although WordNet is believed to be a more comprehensive source of word categories, it does not fit well to our task if we use its categories alone. This is because WordNet is constructed with each single word as a unit, while most of the NPs in the real word consist of multiple words, especially in relations related to names of entities such as actors, books and movie directors. For example, given the noun phrase harry_porter, it is difficult for WordNet to tell that the appropriate category for this NP is harry_porter rather than name. Therefore, the best practice of utilizing the benefits of word categories of both NELL and WordNet would be to combine them for our purpose. In other words, given a NP pair, it is reasonable to first turn to NELL for their categories and then look up and infer the categories of the NP pair from WordNet when there is no matching result found in NELL, which is one of our future works.
4.2 Evaluation of individual features

Apart from the category feature $F_1$, the performance of the other 3 features are also examined in our experiments. The effectiveness of the $DC$ feature of NP pairs proposed in Section 2.2.2 relies largely on the assumption that the counts of occurrences NPs can be regarded as random variables and thus $DC$ is approximately equivalent to 1 if the items in the pair are not related. According to Table 3, this feature does not work very well on relation classification, which is expected since all the training and testing instances are more or less related. This feature has less to do with the characterization of relations. However, it still has its strengths in serving as a “filter” that eliminates NP pairs that are hardly relevant.

The context-based feature, $F_3$ and $F_4$, are showed to perform well, which is grounded on the fact that there are always some patterns that characterize the relations we aim to classify. Notably, the performance of context features based on all non-verb words gained lower accuracy than that are based on only verbs. We attribute this result to the fact that feature space of non-verb words in the context is very sparse, and classification accuracy increased as we made the feature space more compact by extracting only verbs from the context.

Meanwhile, shallow parsing feature alone achieved classification accuracy of more than 70%, which confirms our belief that POS tags are in some degree the abstract characteristics of relations and is therefore appropriate for the task of relation extraction.

4.3 Evaluation of feature combinations

After experimenting with all the features alone, we finally examined the performance of feature combinations. For example, the combination of $F_1$(NELL),$F_2$, $F_3$(Verb)$F_4$ achieved classification accuracy of 91.2% in average while the combination of $F_3$(Verb)$F_4$ achieved accuracy of 83.8%, which is even higher than the combination of $F_1$(WordNet)$F_2$, $F_3$(Verb)$F_4$ (77.3% in average). To conclude, the $F_1$(NELL)$F_2$, $F_3$(Verb)$F_4$ is the best feature set for our proposed model according to the experiment results.

5 Conclusions

With the availability of large-scale knowledge base such as NELL, correct and efficient relation classification becomes possible. In particular, we presented RecSvm, a SVM-based relation classification model, which is able to overcome several shortcomings of classical supervised approaches for relation extraction with the help of comprehensive ontologies like NELL and WordNet. We show an overall performance of more than 90% classification accuracy using the features we proposed, including the categories of NPs and contexts related to them. We envision a more competent relation classification system based on the proposed model as NELL grows larger and more efficient feature sets are extracted.

References


