An Ensemble Approach for Entity Type Prediction over Linked Data

Guangyuan Piao and John G. Breslin

Insight Centre for Data Analytics
National University of Ireland Galway
IDA Business Park, Lower Dangan, Galway, Ireland
{guangyuan.piao}@insight-centre.org

Abstract. This paper provides an overview of the approach done for the JIST 2015 Challenge on Entity Type Prediction over Linked Data. Our approach uses the Random Forest as the classification method with 458 selected features. The result shows that our approach can achieve an F-score of 0.969 in terms of 10-fold cross-validation on the provided training set.

Keywords: Linked Data, Type Prediction, JIST2015

1 Introduction of the Challenge

The challenge is held in conjunction with the 5th Joint International Semantic Technology Conference. The main task of the challenge is to predict labels of entities/resources in Zhishi.me[1], which is one of the most important and interlinked Chinese datasets on the Web of Data. 1,897 entity URLs are provided and 1,397 of them are provided with label information. Table 1 shows 10 labels and the distribution of them in the labeled samples provided by the challenge. The task is the classification of the 500 unlabeled entities. Four different types of information related to the 1,897 entities are provided: (1) abstracts of entities; (2) infobox properties; (3) external links and (4) related pages.

Table 1. The labels and the distribution of them in the labeled samples

insect (124), university (157), game (143), politician (134), city (139), song (139), novel (150), scene (130), cartoon (134), actor (147)

Therefore, our dataset consists of all aforementioned information related to entities and the dataset is stored in a Sesame Native Store [2]. In total, there are 20,020 triples in the dataset.

[1] http://zhishi.apexlab.org/

2 Overall Approach

In this section, we describe the features and the classification method we used for the prediction task in this challenge.

2.1 Features for Predicting Entity Type

We combine features from three different feature sets for describing an entity:

- all distinct *properties* of entities in the dataset
- semantic similarities between the entity and all labels (i.e., `insect`, `novel` etc.)
- a bag of *Named Entities* (NEs) created from all *abstracts* of entities in the dataset

All distinct properties. The features in the first feature set are binary, i.e., the value of a property is 1 if the given entity has the property, and 0 if not.

Semantic similarities. The equation for measuring the semantic similarity between an entity e_i and a label l_u is defined as below:

$$sim(e_i, l_u) = \frac{\sum_{e_j \in l_u} RESIM(e_i, e_j)}{|l_u|} \quad (1)$$

l_u denotes one of the 10 labels in Table 1 and $|l_u|$ denotes the number of entities of label l_u. The semantic similarity between a given entity and an entity of a specific label is measured using `RESIM` [7]. `RESIM` is a similarity method for measuring the semantic similarity between two entities/resources in a Linked Data graph by taking direct and indirect links between them into account. Using equation (1), 10 semantic similarities can be measured for a given entity with respect to 10 labels. For instance, for an entity e_1 and a label `insect`, we measured the similarity between e_1 and each entity of label `insect`, and then measured the average similarity over all entities of the label. In the same way, for an entity e_1, we can get 10 semantic similarity features for 10 different labels. These features reflect the relatedness between the given entity and each label.

A bag of Named Entities. Named Entities (NEs) were extracted from a given abstract using Stanford Named Entity Recognizer (NER) [3]. First, all abstracts in the dataset were segmented using Chinese Stanford Word Segmenter [6]. The Chinese Dictionary used in this segmenter contains 423,200

墨西哥国立自治大学（以下简称墨国大）创建于1551年，是墨西哥和拉丁美洲地区历史最悠久、规模最大的综合性大学，也是世界上规模最大的高等学府之一。	墨西哥,国立,自治,大学,简称,墨,国大,创建,墨西哥,拉丁美洲,地区,历史,最,悠久,规模,最,大,综合性,大学,世界,上,规模,最,大,高等,学府

Fig. 1. An example of named entities in a sequence (right) for an abstract (left)

⋯	墨西哥	⋯	夏季	⋯	大学	⋯
⋯	1.692	⋯	0	⋯	1.192	⋯

Fig. 2. An example of representing an abstract by a set of weighted named entities

unique words. Based on these segmented words, we then used the Stanford NER to retrieve all NEs. Stop words and NEs appeared in all abstracts of entities less than 10 times were removed. By doing so, an abstract a consists of a set of NEs $\{ne_1, ne_2 \ldots, ne_n\}$ in a sequence where n is the total number of NEs in a (see Fig. 1). In addition, $a \in A$ can be represented by a set of weighted NEs $\{w(ne_1), w(ne_2), \ldots, w(ne_m)\}$ where A denotes the set of all abstracts in the dataset and m denotes the total number of distinct NEs in A. The weights of n NEs appeared in an abstract a are calculated by the equation (2). On the other hand, the weights of NEs that did not appear in the abstract are zero (see Fig. 2).

$$w(ne_i, a) = \sum_{ne_i \in a} 1 - \frac{pos(ne_i, a)}{n} \tag{2}$$

$pos(ne_i, a)$ denotes the position of ne_i in the NEs of a. As we can see from the equation (2), the weight of an entity not only takes the number of entity's appearances in an abstract into account but also incorporates the position of the entity in the abstract. This means that entities appeared at the beginning of an abstract and appeared frequently in the abstract can have higher weights. We also investigated the effect of removing NEs with a length smaller than 2 and found it did not improve the performance.

All numeric features (semantic similarity features and NE features) were normalized in the [0,1] interval. Furthermore, irrelevant features might harm a classifier's performance. In order to filter out irrelevant features, we used the `GainRatioAttributeEval` method in Weka [4], an attribute/feature selection method, with a threshold 0.3. Table 2 depicts the number of all features and selected ones after the feature selection in each feature set.

Table 2. The number of features and selected features after feature selection

Feature set	**# of features**	**# of selected features**
All properties of entities	553	154
Semantic similarities for each label	10	7
A bag of Named Entities (NEs)	1,335	297
Total	1,888	458

Table 3. Performances of different classifiers

Classifier	Precision	Recall	F-score
Decision Tree	0.942	0.942	0.942
Support Vector Machine	0.920	0.910	0.912
Random Forest	**0.970**	**0.969**	**0.969**
Stacking with Random Forest	0.949	0.948	0.948

Table 4. Performances of using all features and selected ones

Random Forest	Precision	Recall	F-score
All features	0.961	0.961	0.961
Selected features	**0.968**	**0.968**	**0.968**

2.2 Prediction Strategies

We use the Random Forest, which is an ensemble approach to form a "strong learner" from a group of "weak leaners", as the classification method since it performed best in our experiment. We compared following classifiers in this experiment:

- Decision Trees [9], using the confidence factor 0.25 for pruning
- Support Vector Machine [8], using the complexity parameter C = 1.0
- Random Forest [1], using 100 trees
- *stacking* [10] with Random Forest (with 90 trees and a max depth = 10) using aforementioned classifiers as base classifiers

The experiment was conducted using Weka [4] and we carried out a 10-fold cross-validation to investigate the performance of each classifier with different parameter settings.

3 Results

Table 3 shows the results of different classifiers based on the selected features (see Table 2). As shown in Table 3, Random Forest performs best in terms of 10-fold cross-validation on the provided dataset compared to other approaches. We noticed that stacking with Random Forest (using several base classifiers) did not improve the classification performance. The results of Random Forest using all features and selected ones are presented in Table 4. From the results, it can be observed that the classifier is improved using selected features compared to using all of them.

4 Conclusion

Overall, our approach using Random Forest as the classification method with selected features can achieve an F-score of 0.969 on the provided training set. Therefore, we used this ensemble approach to predict the labels in the provided test set for our final submission (`result.dat`).

Acknowledgments. This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 (Insight Centre for Data Analytics).

References

1. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
2. Broekstra, J., Kampman, A., Van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: The Semantic WebISWC 2002, pp. 54–68. Springer (2002)
3. 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. Association for Computational Linguistics (2005)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11(1), 10–18 (2009)
5. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial intelligence 97(1), 273–324 (1997)
6. Manning, C.: A Conditional Random Field Word Segmenter
7. Piao, G., showkat Ara, S., Breslin, J.G.: Computing the Semantic Similarity of Resources in DBpedia for Recommendation Purposes. In: Semantic Technology
8. Platt, J.: Fast training of support vector machines using sequential minimal optimization. Advances in kernel methodssupport vector learning 3 (1999)
9. Quinlan, J.R.: C4. 5: programs for machine learning. Elsevier (2014)
10. Wolpert, D.H.: Stacked generalization. Neural networks 5(2), 241–259 (1992)