NLP4Types: Predicting Types Using NLP

Idafen Santana-Perez* and Mariano Rico

Ontology Engineering Group,
Universidad Politécnica de Madrid, Madrid, Spain
{isantana, mariano.rico}@fi.upm.es

Abstract. Type inference for resources in Knowledge Graphs is a widely studied problem, for which different approaches have been proposed, including reasoning, statistical analysis, and the usage of the textual information related to the resources. We focus on the latter, exploiting text classification techniques for predicting semantic types from textual descriptions. In this paper we introduce NLP4Types, an online tool that combines different standard NLP techniques and classifiers for predict types based on DBpedia abstracts, as well as to collect feedback from the users for those predictions.

Keywords: DBpedia, Natural Language Processing, Data Quality, Linked Data

1 Introduction

Type statements, that is, assertions of types for entities, are the most basic and fundamental piece of information for semantic resources. This information can be generated by different means, including manual, automated and semi-automated approaches. In this paper we cover DBpedia, which is generated automatically from the information contained in Wikipedia, using a set of translation mappings, from the entries in tabular format contained the infoboxes of each page. As not all pages contain infoboxes it is not always possible to generate type information. According to our calculation, around a 16% of resources from Wikipedia do not have any type mapped to DBpedia. We have also to take into account that, even in those cases in which this information can be generated, it is not always complete or correct, as mappings are defined manually and collaboratively by users.

In this paper we explore how textual abstracts can be exploited, using NLP techniques, to classify entries into the DBpedia ontology. We combine document-to-term matrix and Named Entity Recognition to train a model that we later use to predict types from free text on our tool. We have evaluated our model using K-fold evaluation and a well-known gold standard, obtaining high results. The final result of this process is NLP4Types[1], an online tool that allows user to explore type predictions and to collect feedback from them.

*This work was partially funded by projects RTC-2016-4952-7, TIN2013-46238-C4-2-R and TIN2016-78011-C4-4-R, from the Spanish State Investigation Agency of the MINECO and FEDER Funds.

[1]http://nlp4types.linkeddata.es

2 Related Work

Typing resources on large datasets is a widely studied problem that has been addressed during last decade, being SDType [7] the most prominent system. SDType exploits the statistical information of property distribution to infer new typing statements. Other approaches have been introduced, exploiting different NLP-based techniques for type assignment based on text [3, 2]. In [5] a hierarchy of Support Vector Machines (hSVM) is introduced for applying lexico-syntactic patterns using a bag-of-words model, extracted from short abstracts and Wikipedia categories. This work extends the Linked Hypernym Dataset Framework [4], by the same authors, for extracting these pattern-based structures. These works introduce also the LHD Gold Standard dataset, which we use in this paper, to measure the performance of our system and compare it to other existing tools. This gold standard has been produced, as reported by authors, using experts to assign types to a subset of the English DBpedia resources. We have used it to evaluate our system, as it provides means for comparing our contribution to both, hSVM and SDType.

3 Text Classification

We have implemented a pipeline in which different NLP techniques are combined, for a total of seven steps. These are the main features of these steps: **A) Get Abstract text:** get the text from available abstracts. All those resources that do not have an abstract are discarded as they can not be used to train our system. **B) Named Entity Recognition:** using DBpedia Spotlight [1] the system detects the Named Entities (NE) on the text, obtaining their types. The surface form of those entities is simply ignored and only the types (e.g. Person, Organization) are used, adding them as new words to the text. **C) Text pre-process:** apply several text normalization techniques on the abstracts (i.e. stop words, lemmatization, and stemming). **D) Data Vectorization:** translate textual data into a vector space model, using a Bag of Words approach. We apply a TF-IDF metric to get a more discriminatory score. **E) Training:** train the classifier, using the vectorized data generated before. We have used a Support Vector Machine classifier, as they has been proven to be efficient, performing at the state-of-the-art level[2]. **F) Prediction:** predict types either from test data, reserved during the training phase, or from new unseen data. We use these training and prediction steps when validating our approach against the gold standard. **G) Evaluation:** evaluating the results obtained, comparing how the predictions fit the labelled data. To reduce overfitting, we apply a 5-fold evaluation and the aforementioned gold standard.

[2] `https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-and-machine-learning-on-documents-1.html`

The system has been implemented in Python, including NLTK [6] and scikit-learn [8] libraries for NLP and machine learning processing. The code is available online.[3]

4 Evaluation

We have selected four main metrics for evaluating our results. In a general classification problem, if the predicted label is not the same as the expected one, the prediction is computed as an error. However, when working with labels structured in a hierarchy, more flexible evaluation metrics can be defined to take this into account. This is discussed in [5], where authors use the hierarchical precision, recall, and F-measure to evaluate the performance of different systems over the same gold standard used in our evaluation. Thus, we use these metrics, plus the regular accuracy, to evaluate our system.

We have executed our evaluation using the English version of the DBpedia dataset, from the 2016-10 release, which contains a total of 3.048.742 resources with both type and abstract. Resources that are typed only with owl:Thing are not considered, as inferring this type is trivial and does not add any information for the classification problem. We have executed a 5-fold evaluation over the DBpedia resources, to obtain performance results. We have also evaluated it using the aforementioned gold standard, using only the resources with an abstract associated. From the total of 2.092 resources, 1.825 meet this requirement.

The results obtained are depicted in Table 1. As we can see, the more data we are able to use for training the system, the more precise it gets, obtaining around a 95% of hierarchical F-measure when using the full training set. By using the gold standard dataset, we can compare our system to hSVM and SDType[4]. As we can see, in general, our system outperforms both hSVM and SDType over the gold standard resources.

Table 1: Results of 5-Fold and Gold Standard (GS) evaluations

Resources	Acc.	hPrec	hRecall	hF-measure	GS Acc.	GS hPrec	GS hRecall	GS hF-meas.
NLP4Types	0,835	0,952	0,949	0,950	0,449	0,827	0,822	0,825
hSVM	-	-	-	-	0,548	0,890	0,665	0,761
SDType	-	-	-	-	0,338	0,809	0,641	0,715

5 NLP4Types

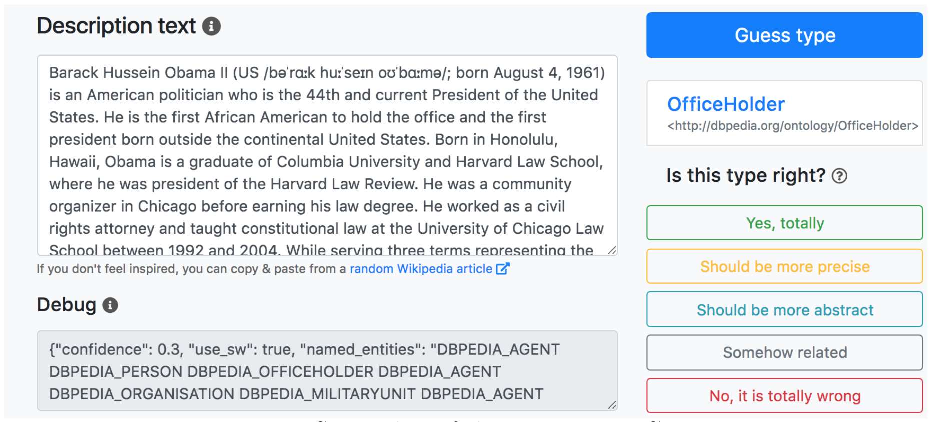
Based on the NLP pipeline introduced above, the online interface of NLP4Types allows user to predict types from any free-text sample. The current version of the tool includes a model trained with all the resources from DBpedia 2016-10,

[3] `https://github.com/idafensp/NLP4Types`

[4] We have included only the highest results reported, shown in Table 6 of the cited paper by Kliegr et. al. [5]

which is used to predict types. Currently only types belonging to the DBpedia ontology are predicted.

Once a prediction is obtained, the user can evaluate the result and provide feedback. As shown in Figure 1, five different criteria are provided, to specify whether the prediction is wrong or right, or how it should be improved. Once the user selects one, it is asked for some extra feedback, including the expected type, user expertise and text source. The main goal of this tool is to collect this feedback and allow to analyze and improve the system and its evaluation



Barack Hussein Obama II (US /bəˈrɑːk huːˈseɪn oʊˈbɑːmə/; born August 4, 1961) is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was president of the Harvard Law Review. He was a community organizer in Chicago before earning his law degree. He worked as a civil rights attorney and taught constitutional law at the University of Chicago Law School between 1992 and 2004. While serving three terms representing the

If you don't feel inspired, you can copy & paste from a random Wikipedia article

```
{"confidence": 0.3, "use_sw": true, "named_entities": "DBPEDIA_AGENT
DBPEDIA_PERSON DBPEDIA_OFFICEHOLDER DBPEDIA_AGENT
DBPEDIA_ORGANISATION DBPEDIA_MILITARYUNIT DBPEDIA_AGENT
```

Fig. 1: Screenshot of the NLP4Types GUI

The system and interface described in this paper are the first steps on exploring how NLP techniques can be applied to infer types in semantic Knowledge Graphs. They set the foundation for future work in which the feedback collected and the different techniques applied will provide a better insight on the problem and its solutions.

References

1. J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *I-Semantics 2013*, 2013.
2. S. Faralli and S. P. Ponzetto. A hearst-like pattern-based approach to hypernym extraction and class induction. In *Semantic Web Evaluation Challenge*, 2016.
3. A. Gangemi, A. G. Nuzzolese, et al. Automatic Typing of DBpedia Entities. In *International Semantic Web Conference Proc.*, pages 65–81, 2012.
4. T. Kliegr. Linked hypernyms: Enriching dbpedia with targeted hypernym discovery. *Journal of Web Semantics*, 31:59 – 69, 2015.
5. T. Kliegr and O. Zamazal. LHD 2.0: A text mining approach to typing entities in knowledge graphs. *Journal of Web Semantics*, 39:47–61, 2016.
6. E. Loper and S. Bird. Nltk: The natural language toolkit. ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
7. H. Paulheim and C. Bizer. Type Inference on Noisy RDF Data. In *The Semantic Web – ISWC 2013*, pages 510–525, 2013.
8. F. Pedregosa, G. Varoquaux, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.