Concept Extraction Challenge: University of Twente at #MSM2013

Mena B. Habib
Database Chair
University of Twente
Enschede, The Netherlands
m.b.habib@ewi.utwente.nl

Maurice van Keulen
Database Chair
University of Twente
Enschede, The Netherlands
m.vankeulen@utwente.nl

Zhemin Zhu
Database Chair
University of Twente
Enschede, The Netherlands
z.zhu@utwente.nl

ABSTRACT

Twitter messages are a potentially rich source of continuously and instantly updated information. Shortness and informality of such messages are challenges for Natural Language Processing tasks. In this paper we present a hybrid approach for Named Entity Extraction (NEE) and Classification (NEC) for tweets. The system uses the power of the Conditional Random Fields (CRF) and the Support Vector Machines (SVM) in a hybrid way to achieve better results. For named entity type classification we used AIDA [8] disambiguation system to disambiguate the extracted named entities and hence find their type.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing; I.7 [**Document and Text Processing**]: Miscellaneous

General Terms

Algorithms

Keywords

Named Entity Extraction, Named Entity Classification, Social Media Analysis, Twitter Messages.

1. INTRODUCTION

Twitter is an important source for continuously and instantly updated information. The huge number of tweets contains a large amount of unstructured information about users, locations, events, etc. Information Extraction (IE) is the research field which enables the use of such a vast amount of unstructured distributed information in a structured way. Named Entity Recognition (NER) is a subtask of IE that seeks to locate and classify atomic elements (mentions) in text belonging to predefined categories such as the names of persons, locations, etc. In this paper we split the NER task into two separate tasks: Named Entity Extraction (NEE) which aims only to detect entity mention boundaries in text; and Named Entity Classification (NEC) which assigns the extracted mention to its correct entity type. For NEE, we used a hybrid approach of CRF and SVM to achieve better results. For NEC, we first apply AIDA disambiguation system [8] to disambiguate the extracted named entities, then we use the Wikipedia categories of the disambiguated entities to find the type of the extracted mention.

2. OUR APPROACH

2.1 Named Entity Extraction

For this task, we made use of two famous state of the art approaches for NER; CRF and SVM. We trained each of them in a different way as described below. The purpose of training is only for entity extraction rather recognition (extraction and classification). Results obtained from both are unionized to give the final extraction results.

2.1.1 Conditional Random Fields

CRF is a probabilistic model that is widely used for NER [5]. Despite the successes of CRF, the standard training of CRF can be very expensive [6] due to the global normalization. In this task, we used an alternative method called *empirical training* [9] to train a CRF model. The maximum likelihood estimation (MLE) of the empirical training has a closed form solution, and it does not need iterative optimization and global normalization. So empirical training can be radically faster than the standard training. Furthermore, the MLE of the empirical training is also a MLE of the standard training. Hence it can obtain competitive precision to the standard training. Tweet text is tokenized using special tweets tokenizer [1]. For each token, the following features are extracted and used to train the CRF: (a) The Part of Speech (POS) tag of the word provided by a special POS tagger designed for tweets [1]. (b) If the word initial character is capitalized or not. (c) If the word characters are all capitalized or not.

2.1.2 Support Vector Machines

SVM is a machine learning approach used for classification and regression problems. For our task, we used SVM to classify if a tweet segment is a named entity or not. The training process takes the following steps:

1. Tweet text is segmented using the segmentation approach as described in [4]. Each segment is considered a candidate for a named entity. A set of features is extracted for each segment and the SVM is trained to distinguish true positive entities from false positive ones. We enriched the segments by looking up a Knowledge-Base (KB) (here we use YAGO [3]) for entity mentions as described in [2]. The purpose of this step is to achieve high recall. To improve the precision a bit, we applied some filtering hypothesis (such as removing

Table 1: Extraction Results

	Pre.	Rec.	F1
Twiner Seg.	0.0997	0.8095	0.1775
Yago	0.1489	0.7612	0.2490
Twiner∪Yago	0.0993	0.8139	0.1771
Filter(Twiner∪Yago)	0.2007	0.8066	0.3214
SVM	0.7959	0.5512	0.6514
CRF	0.7157	0.7634	0.7387
CRF∪SVM	0.7166	0.7988	**0.7555**

Table 2: Extraction and Classification Results

	Pre.	Rec.	F1
CRF	0.6440	0.6324	0.6381
AIDA Disambiguation + Entity Categorization	0.6545	0.7296	**0.6900**

segments that are composed of stop words or having verb POS).

2. For each tweet segment, we extract the following set of features in addition to those features mentioned in section 2.1.1: (a) The joint and the conditional probability of the segment obtained from Microsoft Web N-Gram services [7]. (b) The stickiness of the segment as described in [4]. (c) The segment frequency over around 5 million tweets. (d) If the segment appears in WordNet. (e) If the segment appears as a mention in Yago KB. (f) AIDA disambiguation system score for the disambiguated entity of that segment (if any).

   The selection of the SVM features is based on the claim that disambiguation clues can help in deciding if the segment is a mention for an entity or not [2].

3. An SVM with RBF kernel is trained whether the segment represents a mention of NE or not.

We take the union of the CRF and SVM results, after removing duplicate extractions, to get the final set of annotations. For overlapping extractions we select the entity that appears in Yago, then the one having longer length.

2.2 Named Entity Classification

The purpose of NEC is to assign the extracted mention to its correct entity type. For this task, we first use the prior type probability of the given mention in the training data. If the extracted mention is out of vocabulary (does not appear in training set), we apply AIDA disambiguation system on the extracted mentions. AIDA provides the most probable entity for the mention. We get the Wikipedia categories of that entity from the KB to form an entity profile. Similarly, we use the training data to build a profile of Wikipedia categories for each of the entity types (PER, ORG, LOC and MISC).

To find the type of the extracted mention, we measure the document similarity between the entity profile and the profiles of the 4 entity types. We assign the mention to the type of the most similar profile.

If the extracted mention is out of vocabulary and is not assigned to an entity by AIDA we try to disambiguate the last token of it. If all those methods failed to find entity type for the mention we just assign "MISC" type.

3. EXPERIMENTAL RESULTS

In this section we show our experimental results of the proposed approaches on the training data. All our experiments are done through a 4-fold cross validation approach for training and testing. We used Precision, Recall and F1 measures as evaluation criteria for those results. Table 1 shows the NEE results along the extraction process phases. **Twiner Seg.** represents results of the tweet segmentation algorithm described in [4]. **Yago** represents results of the surface matching extraction as described in [2]. **Twiner∪Yago** represents results of merging the output of the two aforementioned methods. **Filter(Twiner∪Yago)** represents results after applying filtering hypothesis. The purpose of those steps is to achieve as much recall as possible with reasonable precision. **SVM** is trained as described in section 2.1.2 to find which of the segments represent true NE. **CRF** is trained and tested on tokenized tweets to extract any NE regardless of its type . **CRF∪SVM** is the unionized set of results of both **CRF** and **SVM**. Table 2 shows the final results of both extraction with **CRF∪SVM** and entity classification using the method presented in section 2.2. It also shows the **CRF** results when trained to recognize (extract and classify) NE. We considered it as our baseline. Our method of separating the extraction and classification outperforms the baseline.

4. CONCLUSION

In this paper, we present our approach for the IE challenge. We split the NER task into two separate tasks: NEE which aims only to detect entity mention boundaries in text; and NEC which assigns the extracted mention to its correct entity type. For NEE we used a hybrid approach of CRF and SVM to achieve better results. For NEC we used AIDA disambiguation system to disambiguate the extracted named entities and hence find their type.

5. REFERENCES

[1] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proc. of the 49th ACL conference*, HLT '11, pages 42–47, 2011.

[2] M. B. Habib and M. van Keulen. Unsupervised improvement of named entity extraction in short informal context using disambiguation clues. In *Proc. of the Workshop on Semantic Web and Information Extraction (SWAIE)*, pages 1–10, 2012.

[3] J. Hoffart, F. M. Suchanek, K. Berberich, E. L. Kelham, G. de Melo, and G. Weikum. Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In *Proc. of WWW 2011*, 2011.

[4] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. Twiner: named entity recognition in targeted twitter stream. In *Proc. of the 35th ACM SIGIR conference*, SIGIR '12, pages 721–730, 2012.

[5] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. of the 7th HLT-NAACL conference*, CONLL '03, pages 188–191, 2003.

[6] C. Sutton and A. McCallum. Piecewise training of undirected models. In *Proc. of UAI*, pages 568–575, 2005.

[7] K. Wang, C. Thrasher, E. Viegas, X. Li, and B.-j. P. Hsu. An overview of microsoft web n-gram corpus and applications. In *Proc. of the NAACL HLT 2010*, pages 45–48, 2010.

[8] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum. Aida: An online tool for accurate disambiguation of named entities in text and tables. *PVLDB*, 4(12):1450–1453, 2011.

[9] Z. Zhu, D. Hiemstra, P. M. G. Apers, and A. Wombacher. Closed form maximum likelihood estimator of conditional random fields. Technical Report TR-CTIT-13-03, University of Twente, 2013.