Named Entity Extraction and Linking Challenge: University of Twente at #Microposts2014

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 is a potentially rich source of continuously and instantly updated information. Shortness and informality of tweets are challenges for Natural Language Processing (NLP) tasks. In this paper we present a hybrid approach for Named Entity Extraction (NEE) and Linking (NEL) for tweets. Although NEE and NEL are two topics that are well studied in literature, almost all approaches treated the two problems separately. We believe that disambiguation (linking) could help improving the extraction process. We call this potential for mutual improvement, the reinforcement effect. It mimics the way humans understand natural language. Furthermore, our proposed approaches handles uncertainties involved in the two processes by considering possible alternatives.

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 Linking, Social Media Analysis, Twitter Messages.

1. INTRODUCTION

Named Entity Extraction (NEE) is a subtask of IE that aims to locate phrases (mentions) in the text that represent names of persons, organizations or locations regardless of their type. It differs from the term Named Entity Recognition (NER) which involves both extraction and classification into set of predefined classes. Named Entity Linking (NEL) (aka Named Entity Disambiguation) is the task of exploring which correct person, place, event, etc. is referred to by a mention. Wikipedia articles or Knowledge bases (KB) that is derived from Wikipedia are widely used as entities' references. NEE & NEL in tweets are challenging. The informal language of tweets plus their shortness make NEE & NEL processes more difficult.

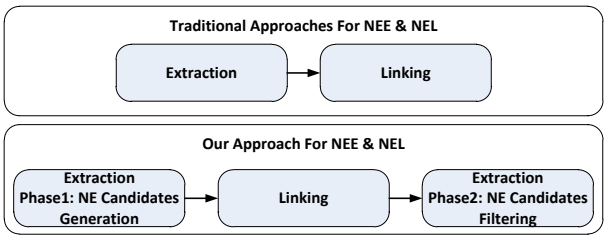

Figure 1: Traditional approaches versus our approach for NEE & NEL.

According to a literature survey, almost no research tackled the combined problem of NEE & NEL. Researchers either focus on NEE or NEL but not both. Systems that do NEL like AIDA [7], either require manual annotations for NE or use some off-the-shelf extraction models like Stanford NER [2]. Here, we present a combined approach for NEE and NEL for tweets with an application on #Microposts 2014 challenge [1]. Although the logical order for such system is to do extraction first then the disambiguation, we start with an extraction phase which aims to achieve high recall (find as much NE candidates as possible). Then we apply disambiguation for all the extracted mentions. Finally, we filter those extracted NE candidates into true positives and false positives using features derived from the disambiguation phase in addition to other word shape and KB features. The potential of this order is that the disambiguation step gives extra information about each NE candidate that may help in the decision whether or not this candidate is a true NE. Figure 1 shows our system architecture versus traditional one.

2. OUR APPROACH

2.1 NE Candidates Generation

For this task, we unionize the output of the following candidates generation methods:

- **Tweet Segmentation**: Tweet text is segmented using the segmentation algorithm described in [6]. Each segment is considered a NE candidate.
- **KB Lookup**: We scan all possible n-grams of the tweet against the mentions-entities table of DBpedia. N-grams that matches a DBpedia mention are considered NE candidates.
- **Regular Expressions**: We used regular expressions to extract numbers, dates and URLs from the tweet text.

2.2 NE Linking

Our NEL approach is composed of three steps; matcher, feature extractor, and SVM ranker.

- **Matcher**: This module takes each extracted mention candidate and looks for its Wikipedia reference candidates on DBpedia. Furthmore, for those mention candidates which don't have reference candidates in DBpedia, we use Google Search API to find possible Wikipedia pages for these mentions. This search helps to find references for misspelled or concatenated mentions like '*justinbieber*' and '*106andpark*'.
- **Feature Extractor**: This module is responsible for extracting a set of contextual and URL features for each candidate Wikipedia page as described in [3]. These features give indicators on how likely the candidate Wikipedia page could be a representative to the mention.
- **SVM Ranker**: After extracting the aforementioned set of features, SVM classifier is trained to rank candidate Wikipedia pages of a mention. For the challenge, we pick the page on the 1st order as a reference for the mention. The DBpedia URI is then generated from the selected Wikipedia URL.

2.3 NE Candidates Filtering

After generating the candidates list of NE, we apply our NE linking approach to disambiguate each extracted NE candidate. After the linking phase, we use SVM classifier to predict which candidates are true positives and which ones are not. We use the following set of features for each NE candidate to train the SVM:

- **Shape Features**: If the NE candidate is initially or fully capitalized and if it contains digits.
- **Probabilistic Features**:
  - The joint and the conditional probability of the candidate obtained from Microsoft Web N-Gram services.
  - The stickiness of the candidate as described in [6].
  - The candidate's frequency over around 5 million tweets[1].
- **KB Features**:
  - If the candidate appears in WordNet.
  - If the candidate appears as a mention in DBpedia KB.
- **Disambiguation Features**:
  - All the features used in the linking phase as described in [3]. We used only the feature set for the first top ranked entity page selected for the given NE candidate.

2.4 Final NE Set Generation

Beside the SVM, we also train a CRF model for NEE. We used the CRF model described in [4]. To generate the final NE set, we take the union of the CRF annotation set and SVM results, after removing duplicate extractions, to get the final set of annotations. We tried two methods to resolve overlapped mentions. In the first method (used in UTwente_Run1.tsv), we select the mention that appears in Yago KB [5]. If both mentions appear in Yago or both don't, we select the one with the longer length. In the second method (used in UTwente_Run2.tsv), we select only the mention with the longer length among the two overlapped mentions. The results shown in the next section are the results of the first method.

The idea behind this unionization is that SVM and CRF work in a different way. The former is a distance based classifier that uses numeric features for classification which CRF can not handle, while the latter is a probabilistic model that can naturally consider state-to-state dependencies and feature-to-state dependencies. On the other hand, SVM does not consider such dependencies. The hybrid approach of both makes use of the strength of each.

[1] `http://wis.ewi.tudelft.nl/umap2011/` + TREC 2011 Microblog track collection.

3. EXPERIMENTAL RESULTS

In this section we show our experimental results of the proposed approaches on the challenge training data [1] in contrast with other competitors. All our experiments are done through a 4-fold cross validation approach for training and testing. Table 1 shows the results of '**Our Linking Approach**' presented in section 2.2, in comparison with two modes of operation of AIDA [7]. The first mode is '**AIDA Cocktail**' which makes use of several ingredients: the prior probability of an entity being mentioned, the similarity between the context of the mention in the text and an entity, as well as the coherence among the entities. While the second mode is '**AIDA Prior**' which makes use only of the prior probability. The results show the percentage of finding the correct entity of the ground truth mentions. Table 2 shows the NEE results along the extraction process phases in contrast with '**Stanford NER**' [2]. Finally, table 3 shows our final results of both extraction and entity linking in comparison with our competitor ('**Stanford + AIDA**') where '**Stanford NER**' is used for NEE and '**AIDA Cocktail**' is used for NEL.

Table 1: Linking Results

	Percentage
Our Linking Approach	**70.98%**
AIDA Cocktail	56.16%
AIDA Prior	55.63%

Table 2: Extraction Results

	Pre.	**Rec.**	**F1**
Candidates Generation	0.120	**0.945**	0.214
Candidates Filtering (SVM)	**0.722**	0.544	0.621
CRF	0.660	0.568	0.611
Final Set Generation	0.709	0.706	**0.708**
Stanford NER	0.716	0.392	0.507

Table 3: Extraction and Linking Results

	Pre.	**Rec.**	**F1**
Extraction + Linking	**0.533**	**0.534**	**0.534**
Stanford + AIDA	0.509	0.279	0.360

4. REFERENCES

[1] A. E. Cano, G. Rizzo, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie. #microposts2014 neel challenge: Measuring the performance of entity linking systems in social streams. In *Proc. of the #Microposts2014 NEEL Challenge*, 2014.

[2] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL 2005*, pages 363–370, 2005.

[3] M. B. Habib and M. van Keulen. A generic open world named entity disambiguation approach for tweets. In *Proc. of KDIR 2013*, pages 267–276, 2013.

[4] M. B. Habib, M. van Keulen, and Z. Zhu. Concept extraction challenge: University of twente at #msm2013. In *#MSM*, pages 17–20, 2013.

[5] 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.

[6] 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 SIGIR 2012*, pages 721–730, 2012.

[7] 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.