MSM2013 IE Challenge NERTUW : Named Entity Recognition on tweets using Wikipedia

Sandhya Sachidanandan, Prathyush Sambaturu, and Kamalakar Karlapalem

IIIT-Hyderabad
{sandhya.s,prathyush.sambaturu}@research.iiit.ac.in
kamal@iiit.ac.in

Abstract. We propose an approach to recognize named entities in tweets, disambiguate and classify them into four categories namely person, organization, location and miscellaneous using Wikipedia. Our approach annotates the tweets on the fly, ie, it does not require any training data.

Keywords: named entity recognition, entity disambiguation, entity classification

1 Introduction

A significant amount of tweets generated each day, discusses about different types of popular entities which may be persons, locations, organizations etc. Most of the popular entities has a page in Wikipedia. Hence, Wikipedia can act as a useful source of information to recognize popular named entities in tweets. Moreover, Wikipedia contains huge number of names of different types of entities which will help us to recognize entities which does not have an explicit page in Wikipedia.

Tweets are of very short length. A tweet may or may not have enough context information to disambiguate the named entities in it. There would be a very small number of words in the tweet which supports the disambiguation of named entities which needs to be utilized efficiently. If the tweet do not have enough context to disambiguate the named entities in it, the popularity of each entity has to be leveraged in disambiguating it. Disambiguating an entity is essential to classify it correctly into location, person, organization or miscellaneous.

Our contributions are :- 1) An approach which utilizes the titles, anchors and infoboxes contained in Wikipedia and a little information from Wordnet and the context information in tweets to recognize, disambiguate and classify named entities in tweets. 2) Our approach does not require any training data and hence no human labelling effort is needed. 3) Along with the global information from Wikipedia, our approach utilizes the context information in the tweet by mapping them to their correct senses using a word sense disambiguation approach which is then used to disambiguate the named entities in the tweet. This will also help in disambiguating the words other than the named entities present in the tweet if any.

2 Approach

- Input tweet is split into ngrams. Link probability of each ngram is calculated as in [1], and those ngrams with link probability less than a threshold τ (experimentally set to 0.01) are discarded . Link probability of a phrase p is calculated as shown in Equation 1.

$$LProb(p) = \frac{n_a(p, W)}{n(p, T)} \tag{1}$$

  where, $n_a(p, W)$ is the number of times a phrase p is used as an anchor text in Wikipedia W and $n(p, T)$ is the number of times the phrase occur as text in a corpus T of around one million tweets. Each concept associated with a phrase, will get the same link probability $LProb(p)$.
- For each ngram, a set of Wikipedia article titles are obtained based on their lexical match. The Wikipedia article titles mapped to the longest matching ngrams are then treated as candidate entities for disambiguation. For each ngram that matched to the title of a disambiguation page in Wikipedia, all the articles related to the ngram are added.
- The candidate entities are then passed on to a Syntax analyser, which uses YAGO's *type* relation to extract WordNet synsets mapped to the candidate entities. With the synsets mapped to the candidate entities and all the synsets of verbs and common nouns associated with the tweet as vertices, a syntax graph is generated using WordNet. The idea behind creating the syntax graph, is to identify the candidate entities which are supported by the syntax of the text. Since, this should be accompanied by disambiguation of words in the text, we found the approach proposed in [3] to be appropriate. In order to identify the candidate entities supported by the syntax of the tweet, we modify [3] by adding words from WordNet which are mapped to the candidate entities, to the syntax graph being generated. If a candidate entity is supported by the syntax of the tweet, the words from WordNet mapped to it get connected to the correct sense of the words added from the tweet in the Syntax graph. A portion of the syntax graph generated for a tweet is shown in Figure 1.
- Page Rank algorithm [2] is then applied on the syntax graph, setting high prior probabilities for synsets of common nouns and verbs added from the tweet. The average of the score of all synsets mapped to a candidate entity is treated as its syntax score.
- With the candidate entities as vertices, a semantic graph is created. The similarity between each pair of candidate entities is calculated and an edge is added with the similarity score as weight if the score is greater than an experimentally set threshold. This makes the most related candidate entities

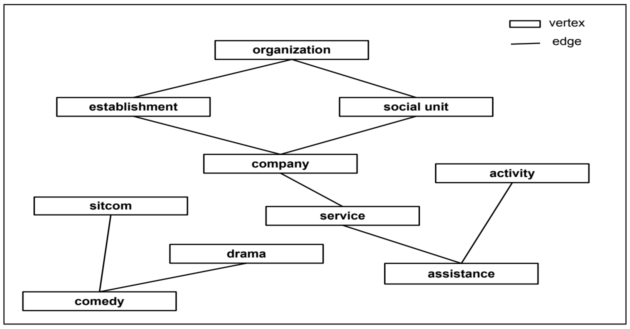

Fig. 1. A portion of the syntax graph created from the tweet - *How can #SMRT take in 161 million of profit and yet deliver sich a crapy service? why do we a company that puts....* The vertices include the words mapped to candidate entities by Yago along with all the senses of common nouns and verbs obtained from the tweet. The edges represents a relation between the vertices in WordNet.

connected in the resulting semantic graph, which may result in many connected components in the graph. An example of a so constructed semantic graph is shown in Figure 2.

- Weighted Page rank algorithm [4] is then applied on the semantic graph and the resulting scores assigned to the candidate entities is treated as the final score for ranking. The priors for each candidate entity is set as the linear combination of the following scores :-
  - Syntax score of each entity as calculated by the Syntax analyzer. This score represents the context information in the tweet.
  - Link probability of the ngram from which the candidate entity is generated.
  - Anchor probability of the candidate entity which is the number of times the entity is used as an anchor in Wikipedia. Both link probability and anchor probability represents the popularity of the candidate entity which plays a significant role in disambiguating the candidate entities in cases where a little or no context information is available in the tweet.
- ***Entity classification***: Each ngram which has a candidate entity in the semantic graph is considered as a named entity. For each ngram, the candidate entity with the highest page rank in the semantic graph is given to a named entity classifier, which uses the keywords present in the infobox of the Wikipedia page of the candidate entity to classify it as person, location, organization or miscellaneous. We extracted the unique keywords with maximum occurence, pertaining to each entity type provided in the training data to classify the named entities.

3 Error analysis and Discussion

- We use an automated and scalable approach to collect keywords from the infoboxes of Wikipedia pages to identify different entity types. Though it

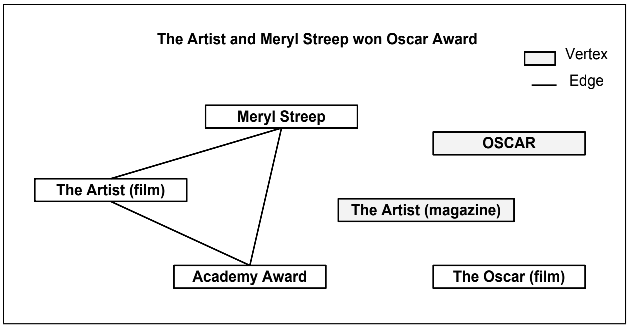

Fig. 2. A portion of semantic graph obtained from the tweet - *The Artist and Meryl Streep won oscar award*. The vertices represent the candidate entities, and edges represent their semantic relatedness.

is able to classify a significant number of entities correctly, it fails in cases where the articles do not contain infobox.

- Since not all entities are present in Wikipedia, we used a post processing step where we merge certain entities with the same type which occur adjacently in the tweet. More post processing can be done by merging adjacently located entities which are not of the same type and assign the most generic type to it which is not done.

References

1. E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM*, 2012.
2. R. Mihalcea, P. Tarau, and E. Figa. Pagerank on semantic networks, with application to word sense disambiguation. In *COLING*, 2004.
3. R. Navigli and M. Lapata. Graph connectivity measures for unsupervised word sense disambiguation. In *IJCAI*, pages 1683–1688, 2007.
4. W. Xing and A. A. Ghorbani. Weighted pagerank algorithm. In *CNSR*, pages 305–314, 2004.