Adapting AIDA for Tweets

Mohamed Amir Yosef, Johannes Hoffart, Yusra Ibrahim,
Artem Boldyrev, Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
{mamir|jhoffart|yibrahim|boldyrev|weikum}@mpi-inf.mpg.de

ABSTRACT

This paper presents our system for the "Making Sense of Microposts 2014 (#Microposts2014)" challenge. Our system is based on AIDA, an existing system that links entity mentions in natural language text to their corresponding canonical entities in a knowledge base (KB). AIDA collectively exploits the prominence of entities, contextual similarities, and coherence to effectively disambiguate entity mentions. The system was originally developed for clean and well-structured text (e. g. news articles). We adapt it for microposts, specifically tweets, with special focus on the named entity recognition and the entity candidate lookup.

Keywords

Entity Recognition, Entity Disambiguation, Social Media

1. INTRODUCTION

Microblogs present a rich field for harvesting knowledge, especially Twitter with more than 500 million tweets per day [5]. However, extracting information from short informal microposts (tweets) is a difficult task due to insufficient contextual evidence, typos, cryptic abbreviations, and grammatical errors. The MSM challenge adresses a fundamental task for knowledge harvesting, namely Named Entity Recognition and Disambiguation (NERD). The goal is to identify entity mentions in text and link them to canonical entities in (mostly Wikipedia-derived) KBs such as `www.yago-knowledge.org` or `dbpedia.org`. We participate in the MSM challenge with an adaptation of the existing AIDA [4] system, a robust NERD framework originally designed for handling input texts with clean language and structure, such as news articles. We adapt it to handle short microposts by adding additional components for named entity recognition, name normalization, and extended candidate entity retrieval. We also integrate data harvested from Twitter API into our model to cope with the context sparsity. Moreover, we tuned the AIDA algorithm parameters to accommodate the brief informal nature of tweets. In the following sections we will first briefly introduce AIDA, then present our approach for adapting AIDA to microblogs, and finally detail our experimental settings.

Published as part of the #Microposts2014 Workshop proceedings, available online as CEUR Vol-1141 (`http://ceur-ws.org/Vol-1141`)

#Microposts2014, April 7th, 2014, Seoul, Korea.

2. AIDA FRAMEWORK OVERVIEW

The AIDA framework deals with arbitrary text that contains mentions of named entities (people, music bands, universities, etc.), which are detected using the Stanford Named Entity Recognition (NER) [2]. Once the names are detected, the entity candidates are retrieve by a dictionary lookup, where the dictionary is compile from Wikipedia redirects, disambiguation pages, and link anchors. For the actual disambiguation, we construct a weighted mention-entity graph containing all mentions and candidates present in the input texts as nodes. The graph contains two kinds of edges: **mention-entity edges:** between mentions and their candidate entities, weighted with the *similarity* between a mention and a candidate entity, and **entity-entity edges:** between different entities with weights that capture the *coherence* between two entities.

The actual disambiguation in form of mention-entity pairs is obtained by reducing this graph into a dense sub-graph where each mention is connected to exactly one candidate entity. The *similarity* between a mention and a candidate entity is computed as a linear combination of two ingredients: 1) the prior probability of a entity given a mention, which is estimated from the fraction of a Wikipedia link anchor (the mention) pointing to a given article (the entity); 2) based on the partial overlap between mention's context (the surrounding text) and a candidate entity's context (a set of keyphrases gathered from Wikipedia). For entity-entity edges we harness the Wikipedia link structure to estimate *coherence* weights. We define the coherence between two entities to be proportional to the number of Wikipedia articles at which they were co-referenced [6]. More details on the features, algorithms and implementation of this approach are included in [4, 7].

3. ADAPTING AIDA FOR TWEETS

AIDA was geared for well-written and long texts, such as news articles. We made the following modifications to adapt it for tweets.

Named Entity Recognition. AIDA originally uses Stanford NER, with a model trained on newswire snippets, a perfect fit for news texts. However, it is not optimized for handling user generated content with typos and abbreviations. Hence, we employ two different components for

mention detection: The first is Stanford NER with models trained for caseless mention detection; the second is our in-house dictionary-based NER tool. The dictionary-based NER is performed in two stages:

1. Detection of named entity candidates using dictionaries of all names of all entities in our knowledge base, using partial prefix-matches for lookups to allow for shortening of names or little differences in the later part of a name. For example, we would recognize the ticker symbol "GOOG" even though our dictionary only contains "Google". The character-wise matching of all names of entities in our KB is efficiently implemented using a prefix-tree data structure.
2. The large number of false positives are filtered using a collection of heuristics, e. g. the phrase has to contain a NNP tag or it has to end with a suffix signifying a name such as "Ave" in "Fifth Ave".

Mention Normalization. The original AIDA did not distinguish between the textual representation of the mention, and its normalized form that should be used to query the dictionary. For example, the hashtag `"#BarackObama"` should be normalized to "Barack Obama" before matching it against the dictionary. Furthermore, many mentions of named entities are referred to in the tweet by their Twitter user ID, such as `"@EmWatson"` the Twitter account of the British actress "Emma Watson". Because the Twitter user IDs are not always informative we access the account metadata, which contains the full user name most of the time. In fact, we attach to each mention string a set of normalized mentions and use all of them to query the dictionary. For example `"@EmWatson"` will have the following normalized mentions {"EmWatson", "Em Watson", "Emma Watson"}, and each of them will be matched against the dictionary to retrieve the set of candidate entities. As the prior probability is on a per-mention basis, we compute the aggregate prior probability of an entity e_i given a mention m_i:

$$prior(m_i, e_i) = \max_{m' \in N(m_i)} prior(m', e_i) \quad (1)$$

where $N(m_i)$ is the set of normalized mentions of m_i. The maximum is taken in order not to penalize an entity if one of the normalized mentions is rarely used to refer to it.

Approximate Matching. This step is employed iff the previous normalization step did not produce candidate entities for a given mention. For example, it is not trivial to automatically split a hashtag like `"#londonriots"`, and hence its normalized mention set, {"londonriots"}, does not have any candidate entities. We address this by representing both the mention strings and dictionary keys as vectors of character-trigrams between which the cosine similarity is computed. We only consider the candidate entity if cosine similarity between the mention and candidate entity keys is above a certain threshold (experimentally determined as 0.6).

Parameter Settings. In our graph representation, the weight of a mention-entity edge is computed by a linear combination of different similarity measures. To estimate the constants of the linear combination, we split the provided tweets training dataset into `TRAIN` and `DEVELOP` chunks, using `TRAIN` for the estimation. We estimated further hyper-parameters for our algorithm (like the importance of mention-entity vs. entity-entity edges) on `DEVELOP`.

Unlinkable Mentions. Some mentions should not be disambiguated to a entity, even though there are candidates for it. This is especially frequent in the case of social media, where a large number of user names are ambiguous but do not refer to any existing KB entity – imagine how many Will Smiths exists besides the famous actor. We address this problem by thresholding on the disambiguation confidence as defined in [3], where a mention is considered unlinkable and thus removed if the confidence is below a certain threshold, estimated as 0.4 on `DEVELOP`.

4. EXPERIMENTS

We conducted our experiments on the dataset provided in [1]. We carried out experiments with three different setups. First we used Stanford NER trained for entity detection, along with mention prior probability and key-phrases matching for entity disambiguation. In the second experiment we added coherence graph disambiguation to the previous setting. The third setting is similar to the first one, but we use our dictionary-based NER instead of Stanford's for entity detection. Note that we automatically annotate all digit-only tokens as mentions using a regular expression, as all numbers were annotated in the training data. The results of running the three experiments on the testing dataset are correspondingly provided with the following ids: AIDA_1, AIDA_2 and AIDA_3.

During our experiments, our runs achieved around 51% F1 on the `DEVELOP` part of the training data, where a mention is counted as true positive only if both the mention span matches the ground truth perfectly and the entity label is correct.

5. CONCLUSION

AIDA is a robust framework that can be adapted to any type of natural language text, here we use it to disambiguate names to entities in tweets. We found that using a dictionary-based NER worked well for the sometimes ill-formatted inputs. An approximate candidate lookup crucially improves recall, which in combination with discarding low-confidence mentions improves the results.

6. REFERENCES

[1] A. E. Cano Basave, G. Rizzo, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie. Making Sense of Microposts (#Microposts2014) Named Entity Extraction & Linking Challenge. In #Microposts2014.

[2] J. R. Finkel, T. Grenager, C. Manning. Incorporating Non-local Information into Information Extraction systems by gibbs sampling. ACL 2005

[3] J. Hoffart, Y. Altun, G. Weikum. Discovering Emerging Entities with Ambiguous Names. WWW 2014

[4] J. Hoffart, M. A. Yosef, I. Bordino et al. Robust Disambiguation of Named Entities in Text. EMNLP 2011

[5] R. Holt. Twitter in numbers, March 2013.

[6] D. Milne, I. Witten. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. WIKIAI workshop at AAAI 2008

[7] M. A. Yosef et al. AIDA: An online tool for accurate disambiguation of named entities in text and tables. VLDB 2011