EDIUM: Improving Entity Disambiguation via User Modeling

Romil Bansal, Sandeep Panem, Manish Gupta and Vasudeva Varma

International Institute of Information Technology, Hyderabad
India

Abstract. Entity Disambiguation is the task of associating entity name mentions in text to the correct referent entities in the knowledge base, with the goal of understanding and extracting useful information from the document. Entity disambiguation is a critical component of systems designed to harness information shared by users on microblogging sites like Twitter. However, noise and lack of context in tweets makes disambiguation a difficult task. In this paper, we describe an **E**ntity **Di**sambiguation system, *EDIUM*, which uses **U**ser interest **M**odels to disambiguate the entities in the user's tweets. Our system jointly models the user's interest scores and the context disambiguation scores, thus compensating the sparse context in the tweets for a given user. We evaluated the system's entity linking capabilities on tweets from multiple users and showed that improvement can be achieved by combining the user models and the context based models.

Keywords: Entity Disambiguation, Knowledge Graph, User Modeling

1 Introduction

Named Entity Disambiguation (NED) is the task of identifying the correct entity reference from the knowledge bases (like DBpedia, Freebase or YAGO), for the given entity. In microblogging sites like Twitter, NED is an important task for understanding the user's intent and for topic detection and tracking, search personalization and recommendations.

In the past, many NED techniques have been proposed. Some utilize contextual information of an entity, while others use candidate's popularity for disambiguation. But tweets being short and noisy, lack sufficient context for these systems to disambiguate precisely. Due to this, the underlying user interests are modeled to disambiguate the entities [1–3]. However, creation of the user models might require some external knowledge (like Wikipedia edits [1]), which are computationally expensive or labor intensive. Many other researchers have also tried to model entity disambiguation through user interest models [2, 3]. Our approach is different from these approaches as it tries to combine contextual models and user models by analyzing the user's tweeting behavior. The user model is built by analyzing user's behavior on the previous tweets.

This approach can be used for modeling users and disambiguating entities in other streaming documents like emails or query logs as well. The next section describes the *EDIUM* system in detail.

2 The *EDIUM* System

EDIUM works by representing user's interests as a distribution over the semantic Wikipedia categories. *EDIUM* has three sub-systems: the context modeling system (Section 2.1), the user modeling system (Section 2.2) and the disambiguation system (Section 2.3). The user's interests are modeled based on the tweet categories by the user. These interests along with the local context are used by the disambiguation system for linking entities in the new tweets. The final results are fed back to the system for improving the user model.

Every tweet has multiple entity mentions and each mention can be aligned to multiple entities. Table 1 represents few notations used while describing the system.

Table 1. Notations used for Describing the System

Symbol	Description
C_i^j	j-th candidate entity for the i-th entity mention in a given tweet
c_i^j	Contextual similarity score for candidate entity C_i^j
$Par(C)$	Parent of C is set of all categories that are the immediate ancestor of category C in Wikipedia Category Network
$G(C)$	Grandparent of C is set of all categories such that $G(C) \subseteq Par(Par(C))$ and $G(C) \cap Par(C) = \emptyset$
$N_r(C)$	Set of all categories in the r-th neighborhood of category C in Wikipedia Category Network
IC_u^i	Set of all categories in the i-th interest cluster for user u
ic_u^i	Score for the interest cluster IC_u^i

2.1 The Contextual Modeling System

Context Model (CM) disambiguates the entities based on the text around the entities. Similarity between the text around the entity mention and the text on Wikipedia page of an entity is compared and an appropriate weightage for disambiguation is given to each candidate entity. The candidate entity with the maximum weightage is considered as the disambiguated entity for the given entity mention.

Many techniques have been proposed to disambiguate the entity mentions in the text [4–6] based on the context. We used existing entity linking systems like DBpedia Spotlight [4] and Wikipedia Miner [5] for linking and disambiguating the entities based on the context.

The contextual score $Score_C(C_i^j)$ is the candidate score normalized based on all the possible candidates for the given entity mention. We improved the referents' disambiguation scores by combining the context based scores with the user's interest based scores in an appropriate manner.

2.2 The User Modeling System

User Model (UM) understands the user's interests and behavior.

UM Creation: We used cluster-weighted models[1] for modeling the user's interests. The following assumptions were made while creating the user models.

- Users only tweet on topics that interest them.
- The amount of interest in a topic is proportional to the information shared by the user on the topic.

Based on these assumptions, we modeled each user into weighted clusters of semantic Wikipedia categories. Each cluster represents the user's interest in specific topic and weight represents the overall interest of the user in that topic.

The UM is updated for user's future tweets based on the categories in the user's current tweet. The tweet categories are extracted using the following steps.

1. Current tweet's entities are discovered via the disambiguation modeling (DM) system (Section 2.3).
2. The entities with sufficiently high confidence are shortlisted to prevent UM from learning incorrect information for the user. We considered only those entities where confidence (ratio of scores of the second ranked entity to the disambiguated entity) is atmost δ. [2]
3. Tweet categories for the shortlisted entities are extracted using Wikipedia. The score of each tweet category is equal to the number of tweet entities belonging to the category.
4. Considering the graph of semantic Wikipedia categories, the tweet categories are smoothed to include the parent categories. Parents are given scores in inverse proportion to their out-degree for each child category. Common parent gets lesser contribution from the child's score as compared to a rare parent.

The UM is created based on the tweet categories. If the category is already present in the model, the score is updated by the normalized sum of the initial and the tweet category score. Otherwise the category and its score is added to the model. As the new tweet category scores get added to the UM, the model evolves to better represent the newly processed tweet.

To find the topics of interest for the user, each category is mapped to a single interest cluster. Although many clustering techniques over the Wikipedia network have been proposed [7], for efficient computations so as to enable streaming scenarios, we formed clusters based on the similar ancestors for a given category. The score of the k-th interest cluster, ic_u^k, for user u, is the sum of the weights of the categories in the cluster k.

Twitter users exhibit different interest behaviors: some users are highly specific while others tweet about almost every category. This can be inferred from the fact that some users tweet about trending hashtags or popular news, while others tweet only about highly specific products or companies. Entity disambiguation depends highly on the behavior of the users. While making use of interest models might be useful in the latter case, it might not be that effective in the former case. To handle this issue, we

[1] http://en.wikipedia.org/wiki/Cluster-weighted_modeling

[2] The parameter δ depends on the use case and performance of the underlying CM system. While high values ensure the large learning rates, low values ensure the performance of the UM system.

introduce the concept of relatedness between the User Model (UM) and the Disambiguation Model (DM).

Similarity between the UM and the DM is defined as the cosine similarity between the tweet categories vector obtained when DM is used ($Score_D(C_i)$, Eq. 6) vs. that when only the user model is used ($Score_U(C_i)$, Eq. 5) for disambiguation.

$$sim(UM, DM) = cos(Score_D(C_i), Score_U(C_i)) \tag{1}$$

Similarly, similarity between the CM and the DM is defined as the cosine similarity between the tweet categories vector obtained when DM is used ($Score_D(C_i)$, Eq. 6) vs. that when only CM is used ($Score_C(C_i)$) for disambiguation.

$$sim(CM, DM) = cos(Score_D(C_i), Score_C(C_i)) \tag{2}$$

At time t, let $R^{(t)}$ denote the relatedness which is defined as the weight-normalized similarity of UM and DM with that of CM and DM. Weights are assigned in inverse proportion to the model's contribution while disambiguation.

$$R^{(t)} = \frac{(1 - \alpha^{(t)}) \times sim(UM, DM)}{(1 - \alpha^{(t)}) \times sim(UM, DM) + \alpha^{(t)} \times sim(CM, DM)} \tag{3}$$

The parameter $\alpha^{(t)}$ measures the consistency of the user's behavior relative to the user's learnt model at time t. If the user's interest changes frequently, UM will not be able to disambiguate the entities properly, keeping the $\alpha^{(t)}$ low. The higher the value of $\alpha^{(t)}$, the more precise is the user model. The weight α is used to tradeoff between the contributions of the user model and the context model for disambiguation Eq. 6).

We update $\alpha^{(t)}$ after each tweet based on the relatedness value of the newly disambiguated tweet. So as to obtain a stable estimate of α, we compute α as the average of the last m relatedness values (Eq. 4). We used $m = 20$ for our experiments.

$$\alpha^{(t+1)} = \frac{1}{m} \sum_{k=0}^{m-1} R^{(t-m)} \tag{4}$$

To deal with the changing user's interests, we set $\alpha^{(t+1)}$ to $0.9\alpha^{(t)}$ after each day. This lowers the dependency of the DM on the UM with time. To avoid incorrect UM learning, α is restricted to maximum of 0.7. This resists the UM from making decisions without taking the context into consideration.

The UM is committed to the database after each transaction and is used whenever new tweet from the same user arrives. This helps us track huge number of users and build the streaming disambiguation system for Twitter streams.

Disambiguation : For each category C_i^j in a tweet, the final score given by the UM is

$$Score_U(C_i^j) = \sum_{k=0}^{n} sim(C_i^j, IC_u^k) \times Score(IC_u^k) \tag{5}$$

where $Score(IC_u^i)$ is the normalized score of the i-th interest cluster for user u, $sim(C_i^j, IC_u^i)$ is number of common elements in IC_u^i and $N_3(C_i^j)$ and n is the number of interest clusters of the user u. The $Score_U(C_i)$ is then normalized across all possible candidates for the i-th entity mention.

2.3 The Entity Disambiguation Modeling System (DM)

DM disambiguates the entities based on the textual context as well as the user's interests. The DM system combines both the context based model's score and the user based model's score using the parameter $\alpha^{(t)}$, that relates the stability of the user with respect to the previous tweeted topics. The final score predicted by the DM is

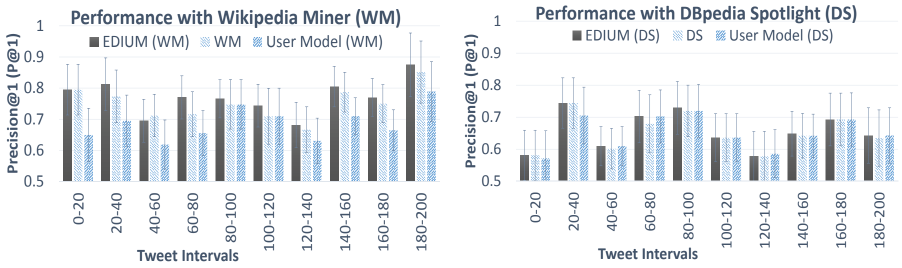
$$Score_D(C_i^j) = \alpha^{(t)} \times Score_U(C_i^j) + (1 - \alpha^{(t)}) \times Score_C(C_i^j) \tag{6}$$

The model selects the entity that maximizes the $Score_D(C_i^j)$ for the given entity mention i.

$$Entity_i = \arg\max_{C_i^j} Score_D(C_i^j) \tag{7}$$

3 Results and Discussions

We evaluated the performance of *EDIUM* on a dataset annotated manually by three individuals. The dataset consists of 200 tweets each from randomly selected 20 different Twitter users[3]. α is initialized to 0.001 for each user because UM has no prior information about the user. We performed experiments with $m = 20$ and $\delta = 0.9$. Parameter tuning was performed based on 200 tweets each from 5 different Twitter users. As the UM processes tweets from the user's tweet stream, it improves over time. For such a streaming scenario, Precision at 1 (P@1) score for entity disambiguation is calculated at interval of 20 tweets for each user. The system is evaluated with both DBpedia Spotlight (DS) and Wikipedia Miner (WM) as the context modeling systems. Fig. 1 reports the performance of the system over time when the proposed model is used vs. when just the CM is used or just the UM (built using previous tweets with the proposed model) is used for disambiguation. We observed that *EDIUM* started to outperform the CM after ~60 tweets (of each user) are processed by the system. The maximum performance is achieved when the proposed model is used with the Wikipedia Miner as the CM system.



(a) Performance with Wikipedia Miner (b) Performance with DBpedia Spotlight

Fig. 1. Precision@1 Score of *EDIUM* under Different Configurations

In general, we observed that *EDIUM* performs better with Wikipedia Miner than with DBpedia Spotlight. This is because the system is dependent on the underlying CM

[3] We would like to thank Wei Shen [2] for providing the dataset.

for learning the user interests initially. Precise CM leads to faster and more accurate UM. The higher the accuracy of the UM, the better the disambiguation.

Conversely, if the underlying CM has low accuracy for entity disambiguation, the UM usually takes much longer to learn the user's true interests. In that case, UM does not contribute in disambiguation giving insignificant improvement in Precision. However, it is observed that UM alone, after sufficient training, can also disambiguate the entities mentioned in the user's tweets and achieve comparable precision.

4 Conclusion and Future Work

In this paper, we have modeled entity disambiguation based on the user's past interest information. We proposed a way to model the user's interests using the entity linking techniques and then using it later to improve the disambiguation in entity linking systems. The gain in precision is proportional to the accuracy of the underlying entity linking system.

More analysis is required on the user modeling aspect of the system. Experiments on larger datasets are required to test the significance and performance of the system on different categories of the users over longer time duration. Currently user's past tweets are used for building the user model and the model's quality depends significantly on the underlying context model. In the future, we plan to include network and demographic information of the users to improve user modeling and hence the entity disambiguation system.

References

1. Murnane, E. L., Haslhofer, B., Lagoze, C.: RESLVE: Leveraging User Interest to Improve Entity Disambiguation on Short Text. In: Proc. of the 22^{nd} Intl. Conf. on World Wide Web (WWW), Republic and Canton of Geneva, Switzerland (2013) 81–82
2. Shen, W., Wang, J., Luo, P., Wang, M.: Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling. In: Proc. of the 19^{th} ACM Conf. on Knowledge Discovery and Data Mining (KDD), New York, NY, USA, ACM (2013) 68–76
3. Yerva, S. R., Catasta, M., Demartini, G., Aberer, K.: Entity Disambiguation in Tweets Leveraging User Social Profiles. In: Proc. of the 2013 Intl. Conf. on Information Reuse and Integration (IRI), 2013, IEEE (2013) 120–128
4. Mendes, P. N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. In: Proc. of the 7^{th} Intl. Conf. on Semantic Systems, New York, NY, USA, ACM (2011) 1–8
5. Milne, D., Witten, I. H.: An Open-source Toolkit for Mining Wikipedia. Artificial Intelligence **194** (2013) 222–239
6. Meij, E., Weerkamp, W., de Rijke, M.: Adding Semantics to Microblog Posts. In: Proc. of the 5^{th} ACM Intl. Conf. on Web Search and Data Mining (WSDM), New York, NY, USA, ACM (2012) 563–572
7. Qureshi, M. A., O'Riordan, C., Pasi, G.: Short-text Domain Specific Key Terms/Phrases Extraction Using an N-gram Model with Wikipedia. In: Proc. of the 21^{st} ACM Conf. on Information and Knowledge Management (CIKM), New York, NY, USA, ACM (2012) 2515–2518
8. Michelson, M., Macskassy, S. A.: Discovering Users' Topics of Interest on Twitter: A First Look. In: Proc. of the 4^{th} Workshop on Analytics for Noisy Unstructured Text Data, ACM (2010) 73–80