GURU PRASAD G, et al, International Journal of Research Sciences and Advanced Engineering [IJRSAE]™
Volume 2, Issue 11, PP: 147 - 150, JUL-SEP' 2015.

LEARNING TO LINK ENTITIES WITH KNOWLEDGE BASE

G GURU PRASAD 1*

1. Asst .Prof, Dept. of CSE, AM Reddy Memorial College of Engineering & Technology, Petlurivaripalem.

Abstract

This paper address the problem of entity linking. Specifically, given an entity mentioned in unstructured texts, the task is to link this entity with an entry stored in the existing knowledge base. This is an important task for information extraction. It can serve as a convenient gateway to encyclopedic information, and can greatly improve the web users' experience. Previous learning based solutions mainly focus on classification framework. However, it's more suitable to consider it as a ranking problem. In this paper, we propose a learning to rank algorithm for entity linking. It effectively utilizes the relationship information among the candidates when ranking. The experiment results on the TAC 20091 dataset demonstrate the effectiveness of our proposed framework. The proposed method achieves 18.5% improvement in terms of accuracy over the classification models for those entities which have corresponding entries in the Knowledge Base. The overall performance of the system is also better than that of the state-of-the-art methods.

1 Introduction

The entity linking task is to map a named-entity mentioned in a text to a corresponding entry stored in the existing Knowledge Base. The Knowledge Base can be considered as an encyclopedia for entities. It contains definitional, descriptive or relevant information for each entity. We can acquire the knowledge of entities by looking up the Knowledge Base. Wikipedia is an online encyclopedia, and now it becomes one of the largest repositories of encyclopedic knowledge. In this paper, we use Wikipedia as our Knowledge Base. Entity linking can be used to automatically augment text with links, which serve as a convenient gateway to encyclopedic information, and can greatly improve user experience. For example, Figure 1 shows news from BBC.com. When a user is interested in "Thierry Henry", he can acquire more detailed information by linking "Thierry Henry" to the corresponding entry in the Knowledge Base.

Figure 1: Entity linking example

Entity linking is also useful for some information extraction (IE) applications. We can make use of information stored in the Knowledge Base to assist the IE problems. For example, to answer the question "When was the famous basketball player Jordan born?", if the Knowledge Base contains the en-tity of basketball player Michael Jordan and his information (such as infobox2 in Wikipedia), the correct answer "February 17, 1963" can be easily retrieved. Entity linking encounters the problem of entity ambiguity. One entity may refer to several entries in the Knowledge Base. For example, the entity "Michael

Jordan" can be linked to the basketball player or the professor in UC Berkeley. Previous solutions find that classification based methods are effective for this task (Milne and Witten, 2008). These methods consider each candidate entity independently, and estimate a probability that the candidate entry corresponds to the target entity. The candidate with the highest probability was chosen as the target entity. In this way, it's more like a ranking problem rather than a classification problem. Learning to rank methods take into account the relations between candidates, which is better than considering them independently. Learning to rank methods are popular in document information retrieval, but there are few studies on information extraction. In this paper, we investigate the application of learning to rank methods to the entity linking task. And we compare several machine learning methods for this task. We investigate the pairwise learning to rank method, Ranking Perceptron (Shen and Joshi, 2005), and the listwise method, ListNet (Cao et al., 2007). Two classification methods, SVM and Perceptron, are developed as our baselines. In comparison, learning to rank methods show significant improvements over classification methods, and ListNet achieves the best result. The best overall performance is also achieved with our proposed framework. This paper is organized as follows. In the next section we will briefly review the related work. We present our framework for entity linking in section 3. We then describe in section 4 learning to rank methods and features for entity linking. A top1 candidate validation module will be explained in section 5. Experiment results will be discussed in section 6. Finally, we conclude the paper and discusses the future work in section 7.

2 Related Work

There are a number of studies on named entity disambiguation, which is quite relevant to entity linking. Bagga and Baldwin (1998) used a Bag of Words (BOW) model to resolve ambiguities among people. Mann and Yarowsky (2003) improved the performance of personal names disambiguation by adding biographic features. Fleischman (2004) trained a Maximum Entropy model with Web Features, Overlap Features, and some other features to judge whether two names refer to the same individual. Pedersen (2005) developed features to represent the context of an ambiguous name with the statistically significant bigrams. These methods determined to which entity a specific name refer by measuring the similarity between the context of the specific name and the context of the entities. They measured similarity with a BOW model. Since the BOW model describes the context as a term vector, the similarity is based on cooccurrences. Although a term can be one word or one phrase, it can't capture various semantic relations. For example, "Michael Jordan now is the boss of Charlotte Bobcats" and "Michael Jordan retired from NBA". The BOW model can't describe the relationship between Charlotte Bobcats and NBA. Malin and Airoldi (2005) proposed an alternative similarity metric based on the probability of walking from one ambiguous name to another in a random walk within the social network constructed from all documents. Minkov (2006) considered extended similarity metrics for documents and other objects embedded in graphs, facilitated via a lazy graph walk, and used it to disambiguate names in email documents. Bekkerman and McCallum (2005) disambiguated web appearances of people based on the link structure of Web pages. These methods tried to add background knowledge via social networks. Social networks can capture the relatedness between terms, so the problem of a BOW model can be solved to some extent. Xianpei and Jun (2009) proposed to use Wikipedia as the background knowledge for disambiguation. By leveraging Wikipedia's semantic knowledge like social relatedness between named entities and associative relatedness between concepts, they can measure the similarity between entities more accurately. Cucerzan (2007) and Bunescu (2006) used Wikipedia's category information in the disambiguation process. Using different background knowledge, researcher may find

different efficient features for disambiguation. Hence researchers have proposed so many effi- cient features for disambiguation. It is important to integrate these features to improve the system performance. Some researchers combine features by manual rules or weights. However, it is not convenient to directly use these rules or weights in another data set. Some researchers also try to use machine learning methods to combine the features. Milne and Witten (2008) used typical classifiers such as Naive Bayes, C4.5 and SVM to combine features. They trained a two-class classifier to judge whether a candidate is a correct target. And then when they try to do disambiguation for one query, each candidate will be classified into the two classes: correct target or incorrect target. Finally the candidate answer with the highest probability will be selected as the target if there are more than one candidates classi- fied as answers. They achieve great performance in this way with three efficient features. The classifier based methods can be easily used even the feature set changed. However, as we proposed in Introduction, it's not the best way for such work. We'll detail the learning to rank methods in the next section.

3 Entity Linking

We now describe the details of building such a system and summarize other systems built for this task. We define entity linking as matching a textual entity mention, possibly identified by a named entity recognizer, to a KB entry, such as a Wikipedia page that is a canonical entry for that entity. An entity linking query is a request to link a textual entity mention in a given document to an entry in a KB. The system can either return a matching entry or NIL to indicate there is no matching entry. In this work we focus on linking organizations, geo-political entities and persons to a Wikipedia derived KB. While the problem is applicable for any language, in this paper we restriction our attention to matching English names to an English knowledge base. 3.1 Key Issues There are 3 challenges to entity linking: Name Variations. An entity often has multiple mention forms, including abbreviations (Boston Symphony Orchestra vs. BSO), shortened forms (Osama Bin Laden vs. Bin Laden), alternate spellings (Osama vs. Ussamah vs. Oussama), and aliases (Osama Bin Laden vs. Sheikh Al-Mujahid). Entity linking must find an entry despite changes in the mention string. Entity Ambiguity. A single mention, like Springfield, can match multiple KB entries, as many entity names, like people and organizations, tend to be polysemous. Absence. Processing large text collections virtually guarantees that many entities will not appear in the KB (NIL), even for large KBs. The combination of these challenges makes entity linking especially challenging. Consider an example of "William Clinton." Most readers will immediately think of the 42nd US president. However, as of this writing, the only two William Clintons in Wikipedia are "William de Clinton" the 1st Earl of Huntingdon, and "William Henry Clinton" the British general. The page for the 42nd US president is actually "Bill Clinton".

4 Candidate Selection for Name Variants

The first system component addresses the challenge of name variants. As the KB contains a large number of entries (818,000 entities, of which 35% are PER, ORG or GPE), we require an efficient selection of the relevant candidates for a query. Previous approaches used Wikipedia markup for filtering – only using the top-k page categories [7] – which is limited to Wikipedia and does not work for general KBs. We first consider a KB independent approach to selection that also allows for tuning candidate set size. This involves a linear pass over KB entry names (Wikipedia page titles): a naive implementation took two minutes per query. Entity Linking as Ranking We consider a supervised machine learning approach to entity linking. Given a query represented by a D dimensional vector x, where $x \in R D$, and we aim to select a single KB entry y, where $y \in Y$, a set of possible KB entries for this query produced by the selection system above, which ensures that Y is small. The ith query is given by the pair {xi , yi},

where we assume at most one correct KB entry. Using these training examples, we can learn a system that produces the correct y for each query. To evaluate each candidate KB entry in Y we create feature functions of the form f(x, y), dependent on both the example x (document and entity mention) and the KB entry y. The features address name variants and entity disambiguation. We categorize the features as atomic features and combination features. Atomic features are derived directly from the named entity in question and its context while combination features are logical expressions of atomic features in conjunctive normal form (CNF). One natural approach to learning would be classification, in which each possible y ∈ Y is classified as being either correct or incorrect. However, such an approach enforces strong constraints: we not only require the correct KB entry to be classified positively, but all other answers to be classified negatively. Additionally, we can expect very unbalanced training, in which the vast majority of possible answers are incorrect. Furthermore, it is unclear how to select a correct answer at test time when multiple KB entries can be classified as correct. Instead, we select a single correct candidate for a query using a supervised machine learning ranker. A ranker will create an ordering over a set of answers Y given a query. Typically, the resulting order over all items is important, such as ranking results for web search queries. In our setting, we assume only a single correct answer and therefore impose a looser requirement, that the correct answer be ranked highest. This formulation addresses several of the challenges of binary classification. We require only that relative scores be ordered correctly, not that each entry be given a label of correct/incorrect. Training is balanced as we have a single ranking example for each query. And finally, we simply select the highest ranked entry as correct, no matter its score.

Conclusion

We presented a state of the art system to disambiguate entity mentions in text and link them to a knowledge base. Unlike previous approaches, our approach readily ports to KBs other than Wikipedia. We described several important challenges in the entity linking task including handling variations in entity names, ambiguity in entity mentions, and missing entities in the KB, and we showed how to each of these can be addressed. We described a comprehensive feature set to accomplish this task in a supervised setting. Importantly, our method discriminately learns when not to link with high accuracy.

References

1. Artiles, J., Sekine, S., Gonzalo, J.: Web people search: results of the first evaluation and the plan for the second. In: WWW (2008)

2. Asahara, M., Matsumoto, Y.: Japanese named entity extraction with redundant morphological analysis. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. pp. 8–15. NAACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003), http://dx.doi.org/10.3115/1073445. 1073447

3. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: Conference on Computational Linguistics (COLING) (1998)

4. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: Association for Computational Linguistics (2008)

5. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD Management of Data (2008)