Utilizing the Categories Network in Wikipedia for Linking Named Entities

Abdullah Bawakid
Faculty of Computing and Information Technology
University of Jeddah, Jeddah, Saudi Arabia
Email: abawakid [AT] uj.edu.sa

Abstract—**This paper addresses the problem of linking named entities in text documents to knowledge base entries. We describe the implementation of a novel unsupervised system we developed for this task and its main stages which include reforming the query, generating and selecting the candidate entities and clustering NIL entities which have no match in the reference knowledge base. The approach we describe in this paper relies on specific elements of Wikipedia, namely its categories network and articles titles. It also utilizes a term-categories vector that defines the relationship strength between terms and the different categories in Wikipedia. The experiments we performed on the Text Analysis Conference (TAC) English Entity Linking challenge reflect the competitiveness and effectiveness of our system.**

***Index Terms*—Named Entities Linking, Wikipedia, Knowledge base, Wikipedia categories.**

I. Introduction

With the continuous expansion of the web and the increasing amount of available electronic text data in the past two decades, the need to have efficient and practical methods to automatically extract useful information in a structured form has continued to grow. Work in the field of data mining spanned many areas which attempted to address this problem such as automatic text summarization, classification and clustering, and dealing with named entities. For named entities, there has been studies focusing on recognizing named entities as in [1] while some other studies focused on linking named entities [2]. Both of these sub problems are important and have many potential applications that are used by many web surfers. Take for example the application of web search. When a user searches for a specific named entity, it would be useful if the search engine provided facts and details about this named entity in the search results page instead of merely links to web pages that may include mentions of these entities. The social media is also another similar application.

In this paper, the focus is on the problem of linking named entities mentions in text to their corresponding entities in a previously prepared catalog of entities [3]. This problem is commonly known as named entities linking. We describe a system in this paper that addresses this problem and give details about its implementations. The system we developed is unsupervised and requires no training data. The evaluation we performed suggest a competitive performance for the developed system.

In the next section we give some background about what was achieved in the literature and the most related work to ours. In section 3 we give details about the design and implementation of our system. Section 4 describes the evaluation that was performed and presents the achieved results. Finally the conclusion is presented in section 5.

II. Background and Related Work

The problem of named entities disambiguation was addressed in the literature from different perspectives. Some work was achieved in this task without assuming prior knowledge about the queried entities. They addressed the problem through clustering equivalent or highly related entities as in [4], [5] and [6]. More recent work utilized knowledge repositories that contain information about different entities. They addressed the problem by attempting to match the queried named entities to the best reference entity in the available knowledge repository. The work done in [7] is among the earliest that took this direction. The authors developed a supervised learning based approach that uses a custom linear similarity function taking into account the context terms surrounding the ambiguous named entity. They also set out a threshold for the custom similarity function they developed. If the similarity score for a named entity is found to be less than the pre-defined threshold, it would not be assigned by their system.

In another work by [8] , they also utilized a reference knowledge base to match the queried entities to their corresponding ones in the reference. However, their work differed from most of the others in that they relied not on the surrounding terms, but on the context named entities that exist in the test document. The approach they implemented was also supervised and used the Vector Space IR model effectively evaluating the source documents against the reference vectors.

Previous attempts have also tried to address the disambiguation problem from another perspective by focusing only on specific entity types. For example, in [9] the authors focused on disambiguating only location names for the Finnish Language. They used two ontologies for Finnish locations names, namely SUO and SAPO, for identifying location names within text. The work in [10] also focused on disambiguating locations mentions within text. They argued in their findings that best results can be obtained for identifying entities is through the usage of gazetteer matching together with simple heuristics such as considering the population count and importance for the candidate locations.

The work in [11] considered the usage of a web-scale system for disambiguating named entities. The algorithm they suggested was meant to handle the problem of imbalanced distribution of data for the different references of named entities. This is especially evident in the web as most named entities tend to have a small set of very popular references that most mentions for the named entity refer to. Hence, a graph-based clustering algorithm was suggested which works by linking named entities mentions together if their similarity was found to exceed a previously set threshold.

In [12], Wikipedia was used as the basis for an algorithm that disambiguates named entities. In their work, Wikipedia was leveraged by building a large-scale semantic network for computing the similarity between the different occurrences of named entities. They compared their system against traditional bag-of-words methods and reported improvements in performance reaching up to 10.7%. In a similar direction, the work in [13], [14] and [15] also devised algorithms leveraging Wikipedia for computing the similarity between named entities for the purpose of entities disambiguation.

The work in [16] utilized several learning to rank methods including SVM regression [17] and ListNet [18]. They used twenty features for candidate entities representation. The used features can be classified into mainly three categories, namely surface features, context features and other special features. Their results indicate that the ListNet approach provided the best performance. In [19] and [20], different queries were handled with different learning to rank methods. They divided the queries into different groups based on their difficulties and applied a specific learning to rank approach to each group separately.

In this paper, we also use Wikipedia as our underlying ontology that aids our algorithm in the named entities linking task. However, our work differs from what was covered in the literature from different perspectives. We only use specific elements of Wikipedia, namely the categories network in addition to the articles titles. The inner content of the articles as well as the links are not employed. The implemented algorithm attempts to match queries to the reference named entities that exist in the reference knowledge base. In case there is no match, the system clusters the remaining NIL entities into different clusters based on how similar they are found to be. The system we developed is simple and requires no training data and is unsupervised. The performance results we obtained when we evaluated our system were encouraging and they indicate the competitiveness of the algorithm we utilized.

III. System Description

The system we developed is composed of different modules implementing three major stages. The first is reforming and expanding the query with the aid of custom rules we apply. The second is preparing and selecting the top candidate named entities for the reformulated query. The third is clustering the remaining NIL entities. In the next subsections we describe how each of these stages was implemented.

A. A. Reforming the Query

The purpose of this stage is to find possible expansions for the given query. This can be very useful especially for acronyms as they tend to be less ambiguous when they are expanded to their longer form with the aid of the surrounding context in the document they are contained in. Take as an example the acronym “Nato” in the following two sentences:

- Nato is an organization that is concerned with the operation of movie theatres in the USA.
- The North Atlantic Treaty includes an article requiring Nato members to aid any of its members that come under attack.

In the first sentence, the phrase “movie theatres” helps with expanding Nato into the “National Association of Theatre Owners”. The second sentence has the phrase “North Atlantic Treaty” which aids with expanding Nato to be become “North Atlantic Treaty Organization”. Thus, we consider both “National Association of Theatre Owners” and “North Atlantic Treaty” as possible expansions for the given query. The implementation of this process involves the usage of the Wikipedia Titles which were previously extracted to discover the alternative and expanded forms for acronyms.

The query reformation stage is also useful in cases when an entity name consists of more than one term, but it is referred to occasionally in the document with only one term. Take for example the following sentence for the query term “Bush”:

- George Bush is a businessman and served as a president for the USA.

In the above sentence, we see that the phrase “George Bush” contains the query “Bush”. Therefore, it is possible that the query is part of the longer phrase that contains the query in whole. Hence, the query “Bush” is expanded into different longer forms including “George W. Bush” and “George H. W. Bush”. The implementation for this was applied by scanning the document terms to see if the query is contained in whole by another phrase in the document. If a containing phrase is found, we see if the expanded form is a possible Wikipedia title. If it is, then we consider the expanded form of the query in the following stage.

B. B. Generating and Selecting Top Candidates

The goal of this stage is to make a list of the candidate named entities for the query and select the most suitable candidate from the reference knowledge base if it exists. If there is no matching named entity, NIL would be selected. The implementation of this stage involves the usage of a term-categories vector we previously built with the aid of Wikipedia. This vector serves the purpose of showing the relevance strength for each term to the different categories within Wikipedia. The construction of this vector involves the usage of the articles titles in Wikipedia in addition to the categories hierarchies attached to each of its articles. This was translated in our work with the following formula:

$$w_t = \frac{1}{|a_t|} \times log \frac{|C|}{|C_t|} \qquad (1)$$

In the above formula we have w_t referring to the weight of word t, $|a_t|$ representing the number of the articles titles that has the word t, $|C|$ reflecting the total number of the Wikipedia categories, and $|C_t|$ for the categories that has the word *t*.

The context terms for each query are examined against the built term-categories vector. The aggregation of the matched categories among the different terms in the context text leads to forming several top representative categories for each potential named entity for the given query. The best representative named entity would be selected in our system as the one having the highest ranked aggregated category score. In case there are no potential candidates that exist in the knowledge base for the query, we would label it with NIL. We perform the mentioned aggregation by applying the following formula:

$$w_a = \mathrm{sum}_{t\rightarrow d}(wf_t \times w_t) \times \frac{1}{|L_a|} \times \frac{|a_d|}{|a|} \qquad (3)$$

In formula 2 shown above, the weight of the candidate named entity represented in Wikipedia with the article *a* is referred to as w_a, the frequency of the word *t* is referred to as wf_t, the number of words present in the article title *a* for the candidate named entity is accounted for with wf_t, and $|a_d|$reflects the number of words that exist in both the candidate named entity and the context text *d*. Also in the above equation we have the element $|L_a|$ which accounts for the number of the articles that have the same title as the article *a*. This addition is especially important when stemming is applied as it gives more weight to the article titles which are linked the least amount of articles.

C. A. Clustering NIL entities

After completing the previous stage, we should have two sets of named entities: one that has query entities which are linked to their corresponding named entities in the knowledge base, and another set which has query entities called NIL with no match in the knowledge base. The goal of this stage is to cluster the NIL entities into segments based on their relatedness.

We perform this stage by examining the corresponding context for each of the NIL queries. We apply formula 2 to all of the context terms in d. This way, we should have a score wa for every article title a in Wikipedia representing how relevant it is to the d. Afterwards, we give a representative score to all of the previously retrieved categories from Wikipedia. The score should reflect how representative the category is to the context terms in d. We apply this aggregation through summing the weights of the elements in wa for all the terms in d. We then would have a corresponding score for each category representing how relevant it is to the d. The higher the obtained score, the more representative the category is.

Afterwards, we select the top P categories for the context document d and create clusters through the usage of the categories and their corresponding scores. In the beginning, the number of clusters would be equal to the number of NIL queries. However, this number would be reduced as we combine clusters together after computing the cosine distance between each cluster and the remaining clusters in the NIL set. If the computed distance is found to be under the previously set threshold, the two clusters would be merged.

IV. Evaluation and Results

We used the TAC2013 KBP dataset provided for the English entity linking to evaluate our system. The total number of queries to be evaluated in this dataset is 2190. We report the results of the evaluation performed with this dataset in Table 1. Also, we show in the same table the median results obtained for the submitted runs by other participants in that year’s competition in addition to the highest obtained score for different query types. It can be noted that our system obtained overall good results across the different document types and also for the geo-political (GPE) and Organizations (ORG) entity categories.

The achieved results illustrated in Table 1 show that our system scores for the different entities types are better than the obtained median for the other systems that participated in this task in TAC2013. Many of the runs that were evaluated for the English entity linking task were supervised and learning-based. The achieved results also highlight the competitiveness of the system we designed and implemented especially when factoring in the fact that it is unsupervised and requires no prior training data.

Table 1. B^3+ F1 scores obtained over the TAC2013-KBP English Entity Linking dataset

Query Focus	Highest	Median	Our System
All	74.6	57.4	58.3
In KB	72.2	55.4	58.4
Not in KB	77.7	56.6	61.3
NW (news docs)	82.9	64.5	64.9
WEB (web docs)	67.8	52.5	58.4
DF (forums docs)	66.2	48.8	48.1
PER	77.8	62.7	53.9
ORG	73.7	54.2	62.3
GPE	74.6	55.2	56.8

V. Conclusion

In this paper we described a novel system for linking query named entities to their corresponding entities in a knowledge base. In contrast to other systems in the literature, our system utilizes only limited aspects of Wikipedia which are its categories network and the articles titles. The inner content of the articles and its inner and inter links are not utilized in our system. The system also employs a term-categories vector that was constructed with the aid of Wikipedia. This vector defines how strongly a term is related to Wikipedia categories.

In the developed system, there are three main stages involved in the process of linking named entities. First is reforming and expanding the query to include its longer form. Second is generating candidate entities and selecting the most suitable one if it exists. Third is clustering NIL entities. We described in details the implementation of these stages. We also ran an evaluation experiment to test the performance of our system. Our findings from this experiment were encouraging, especially when considering that the developed system is unsupervised and requires no prior training or training data.

REFERENCES

[1] X. Liu, M. Zhou, F. Wei, Z. Fu, and X. Zhou, “Joint Inference of Named Entity Recognition and Normalization for Tweets,” in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1, Stroudsburg, PA, USA, 2012, pp. 526–535.

[2] W. Shen, J. Wang, P. Luo, and M. Wang, “Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling,” in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2013, pp. 68–76.

[3] L. Ratinov, D. Roth, D. Downey, and M. Anderson, “Local and Global Algorithms for Disambiguation to Wikipedia,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, Stroudsburg, PA, USA, 2011, pp. 1375–1384.

[4] A. Bagga and B. Baldwin, “Entity-based Cross-document Coreferencing Using the Vector Space Model,” in Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1, Stroudsburg, PA, USA, 1998, pp. 79–85.

[5] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, “Robust Disambiguation of Named Entities in Text,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2011, pp. 782–792.

[6] M. H. Nadimi and M. Mosakhani, “A more Accurate Clustering Method by using Co-author Social Networks for Author Name Disambiguation,” J. Comput. Secur., vol. 1, no. 4, Feb. 2015.

[7] R. Bunescu, “Using Encyclopedic Knowledge for Named Entity Disambiguation,” in In EACL, 2006, pp. 9–16.

[8] S. Cucerzan, “Large-Scale Named Entity Disambiguation Based on Wikipedia Data.,” EMNLP 2007 Empir. Methods Nat. Lang. Process. June 28-30 2007 Prague Czech Repub., pp. 708–716, 2007.

[9] T. Kauppinen, R. Henriksson, R. Sinkkilä, R. Lindroos, J. Väätäinen, and E. Hyvönen, “Ontology-based Disambiguation of Spatiotemporal Locations,” in IRSW, 2008.

[10] J. L. Leidner, “Toponym resolution: A comparison and taxonomy of heuristics and methods,” PhD Thesis, University of Edinburgh, 2007.

[11] L. Sarmento, A. Kehlenbeck, E. Oliveira, and L. Ungar, “An Approach to Web-Scale Named-Entity Disambiguation,” in Proceedings of the 6th International Conference on Machine Learning and Data Mining in Pattern Recognition, Berlin, Heidelberg, 2009, pp. 689–703.

[12] X. Han and J. Zhao, “Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge,” in Proceedings of the 18th ACM Conference on Information and Knowledge Management, New York, NY, USA, 2009, pp. 215–224.

[13] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, “Entity Disambiguation for Knowledge Base Population,” in Proceedings of the 23rd International Conference on Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 277–285.

[14] X. Han and J. Zhao, “Structural Semantic Relatedness: A Knowledge-based Method to Named Entity Disambiguation,” in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 50–59.

[15] M. Xu, Z. Wang, R. Bie, J. Li, C. Zheng, W. Ke, and M. Zhou, “Discovering Missing Semantic Relations between Entities in Wikipedia,” in The Semantic Web – ISWC 2013, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, Eds. Springer Berlin Heidelberg, 2013, pp. 673–686.

[16] Z. Zheng, F. Li, M. Huang, and X. Zhu, “Learning to Link Entities with Knowledge Base,” in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 483–491.

[17] D. Basak, S. Pal, and D. C. Patranabis, “Support vector regression,” Neural Inf. Process.-Lett. Rev., vol. 11, no. 10, pp. 203–224, 2007.

[18] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, “Learning to Rank: From Pairwise Approach to Listwise Approach,” in Proceedings of the 24th International Conference on Machine Learning, New York, NY, USA, 2007, pp. 129–136.

[19] Z. A. Zhu, W. Chen, T. Wan, C. Zhu, G. Wang, and Z. Chen, “To Divide and Conquer Search Ranking by Learning Query Difficulty,” in Proceedings of the 18th ACM Conference on Information and Knowledge Management, New York, NY, USA, 2009, pp. 1883–1886.

[20] I. Anastácio, B. Martins, P. Calado, and others, “Supervised learning for linking named entities to knowledge base entries,” Proc. TAC, 2011.