Disambiguating Entities Referred by Web Endpoints using Tree Ensembles

Gitansh Khirbat **Jianzhong Qi** **Rui Zhang**
Department of Computing and Information Systems
The University of Melbourne
Australia
gkhirbat@student.unimelb.edu.au
{jianzhong.qi, rui.zhang}@unimelb.edu.au

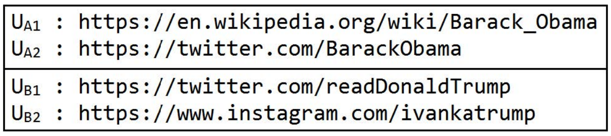
Abstract

This paper describes system details and results of team "EOF" from the University of Melbourne in the shared task of ALTA 2016, which addresses the use of cross document coreference resolution to determine whether two URLs refer to the same underlying entity. In our submission, we develop a two stage system which first identifies the underlying entity for a given URL using entity-level features by ranking the entity mentions present in the crawled text with the help of logistic regression. This is followed by disambiguating entities present in the given pair of URLs using a tree ensemble model to classify if both URLs refer to the same underlying entity. Our system achieved a final F1-score of 86.02% on the private leaderboard[1], which is the best score among all the participating systems.

1 Introduction

The exponential expansion of the World Wide Web has resulted in a large data repository, the majority of which is in the form of unstructured natural language text containing ambiguous name entities. A name entity mention may relate to multiple known entities. For example, the entity mention "New York" may refer to the city of New York or the movie New York which was released in 2009.

Entity linking (EL) is the process of resolving disambiguity between textual entity mentions and the correct entity node in the *knowledge base* (KB). EL systems usually rely on semantic resources like Wikipedia as endpoints for disambiguation (Shen et al., 2015), however, Chisholm et al. (2016) provide a relaxed definition of a KB as any *uniform resource locator* (URL) which reliably disambiguates linked mentions on the web (Chisholm et al., 2016a). This relaxed definition has motivated the shared task of ALTA 2016 (Chisholm et al., 2016b). The task organizers provided manually selected URL pairs from a heterogenous collection of websites including popular social networking websites like LinkedIn, Twitter, ResearchGate; knowledge bases like Wikipedia, IMDB and news websites like NDTV and Economic Times. The participants are asked to classify whether a given pair of URLs refer to the same underlying entity. For example, in Figure 1, URLs in the pair $< U_{A1}, U_{A2} >$ refer to the same entity "Barack Obama" whereas URLs in the pair $< U_{B1}, U_{B2} >$ refer to two different entities "Donald Trump" and "Ivanka Trump".



```
UA1 : https://en.wikipedia.org/wiki/Barack_Obama
UA2 : https://twitter.com/BarackObama
UB1 : https://twitter.com/readDonaldTrump
UB2 : https://www.instagram.com/ivankatrump
```

Figure 1: Example of URL pairs

Considerable research has been done in the field of EL using existing KB like DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008) and KnowItAll (Etzioni et al., 2004). Wikipedia has proven to be a great resource in solving EL tasks (Cucerzan, 2007); (Milne and Witten, 2008) where dictionary-based techniques, contextual features and entity references have been used to train classifiers. Chisholm et al. (2016) study link behaviour and propose a KB discovery method using URL path features by inferring endpoints via logistic regression.

We adopt a two stage approach to solve this problem. First, our system determines the possible underlying entities for a given URL using entity features obtained from the crawled text with the

[1] https://inclass.kaggle.com/c/alta-2016-challenge/leaderboard

Gitansh Khirbat, Jianzhong Qi and Rui Zhang. 2016. Disambiguating Entities Referred by Web Endpoints using Tree Ensembles. In *Proceedings of Australasian Language Technology Association Workshop*, pages 176−180.

help of logistic regression. Next, entities are disambiguated between the given URL pair to classify if both URLs refer to the same underlying entity. Contextual features in and around the entities are exploited and a tree ensemble model is trained for this task.

The rest of the paper is organized as follows. Section 2 describes the methodology in detail. Section 3 describes the experiments and results. Section 4 discusses the error analysis of the obtained results and Section 5 concludes the paper.

2 Methodology

The goal of ALTA 2016 shared task is to determine if a given pair of URLs refer to the same underlying entity. This is essentially a problem of cross-document coreference resolution. We tackle this task as an EL or *named entity disambiguation* (NED) problem. As compared to the traditional NED problem, where entity mention in the text is disambiguated to the entities present in a KB, the difference in this task lies in disambiguating the entities identified from two given URLs without an existing KB.

We treat this task as a supervised classification problem which involves two sequential subproblems, i.e., entity endpoint determination and entity disambiguation. The complete solution pipeline is show in Figure 2. First, the given URLs are crawled using *Scrapy* (Myers and McGuffee, 2015) to obtain textual content from the webpage. The next steps are described below.

2.1 Entity Endpoint Determination

The first stage of our system is to identify the underlying entity for a given URL. It involves three components as described below.

2.1.1 Preprocessing

The preprocessing module consists of tokenization of a given URL and the page title of the webpage corresponding to that URL. We define regex patterns which split a given URL on forward slash characters and hyphens. Research has shown that the path tokens are good indicators of entity mentions. We leverage the observation made by Chisholm et al. (2016a) that the URLs which contain terms like “profile”, “wiki”, “name”, “people” provide a positive evidence to refer to entity pages, whereas URLs containing terms like “news”, “topic” or date patterns like “YYYY/MM/DD” provide a negative evidence.

2.1.2 Named Entity Recognition

The next step is to make use of a *named entity recognition* (NER) system to identify all the entities present in the crawled text. We make use of Stanford’s NER system (Finkel et al., 2005) which uses a model trained on MUC6, MUC7 and ACE 2002 datasets to classify words into three categories namely *Location*, *Person* and *Organization*. The details about this NER system is beyond the scope of this paper and can be obtained from Finkel et al. (2005).

2.1.3 Entity Ranking

Entity ranking is the key step in Stage 1. It trains a logistic regression model using the features obtained in Sections 2.1.1 and 2.1.2 to assign a score for each entity identified in the crawled text. We consider four main features:

1. Comparison of entity mention with the text obtained from URL - Hamming distance is measured for a partial and exact match.
2. Comparison of entity mention with the text obtained from webpage title of the given URL - Hamming distance is measured for a partial and exact match.
3. Frequency of occurrence of entity mention - We observe that in most cases, the most frequent entity is the most probably endpoint.
4. Position of entity mention in the crawled text - We observe that in most cases, the most probable endpoint is an entity mention which is located within the first five tokens in the crawled text.

Using these features, we train a logistic regression model which gives us the probability of an entity being a possible webpage endpoint. This probability score is used to shortlist top-3 entity mentions as the most likely endpoints for a given URL. We observe that an entity endpoint is usually characterized by some related entities. This motivates us to retain the top-3 entities which prove to be useful in the next stage.

2.2 Entity Disambiguation

The second stage of our system solves the problem of determining whether a given pair of URLs refer to the same underlying entity. It makes use of the output of Stage 1 and involves two components as described below.

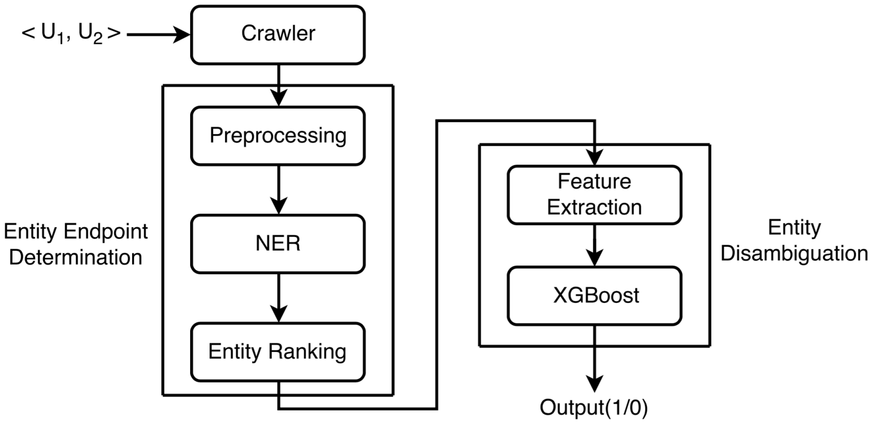


Figure 2: System pipeline

2.2.1 Feature Extraction

This module makes use of contextual features in and around the identified entities. A concept vector is created to represent the semantic content of the crawled text from the URL. This concept vector contains TF-IDF of URL path, page title and top-3 entity mentions obtained from Stage 1 and adds features of bag of words (Guo et al., 2013); (Ratinov et al., 2011) and anchor texts (Kulkarni et al., 2009) as described below.

- Bag of words - TF-IDF summary of the entire crawled text is generated and top-20 words after removal of stopwords are chosen as the representative bag of words.
- Anchor texts - The URLs referred in all the anchor texts are preprocessed according to Section 2.1.1 to obtain the URL endpoint. A vector containing all such endpoints and anchor texts is used to define a TF-IDF vector for the given URL pair.

2.2.2 XGBoost

The features defined in Section 2.2.1 are used to train a supervised tree ensemble classifier called extreme gradient boosting (XGBoost) (Chen and Guestrin, 2016). The intuition behind XGBoost is that since it is not easy to train all the trees at once, an additive strategy is employed to fix what has been learnt which adds one new tree at a time. XGBoost tackles regularization very carefully, which improves the overall score. Detailed working of XGBoost is beyond the scope of this paper and we refer the readers to Chen et al. (2016) for details.

3 Experiments and Results

The ALTA shared task is to classify whether a given pair of URLs refer to the same underlying entity. We first describe the given dataset briefly, followed by the experimental setup and results.

3.1 Dataset

The shared task organizers provide a corpus of URLs from a heterogenous collection of websites including popular social networking websites, knowledge bases and news websites. The training data consists of these URLs in the form of a pair along with their annotations, i.e., 0 if the URLs in a pair refer to different entities or 1 if they refer to the same entity. In addition to this, information about the webpage title and a small snippet is provided for both URLs. The training and test data consist of 200 pairs of URLs each. Data details are given by Chisholm et al. (2016b).

3.2 Experimental Setup and Results

In the Stage 1 sub-problem of entity endpoint determination, we leverage the output of NER to manually annotate the given 200 URL pairs of training data with the top-3 possible entity endpoints, which become the gold standard annotations for this sub-problem. We split this data equally into training and development datasets. We train a logistic regression model on this training data to learn the regression parameters. Using the learnt parameters, we run the model on development data and obtain a F1-score of 89% in classifying if an identified entity mention is one of the top-3 manually annotated entity endpoints for

Table 1: Results on public and private leaderboards

Features	**Precision**	**Recall**	**Public F1**	**Private F1**
{URL, Title}	68.63	87.5	76.92	80.85
+{Bag of Words}	80.39	85.42	82.82	83.49
+{Entity Features}	78.43	97.56	86.96	81.82
+{Anchor Texts}	86.27	95.65	90.72	**86.02**

the given URL. This gives us a positive confidence to proceed with combining the training and development datasets (i.e. the given original full training dataset consisting of 200 URL pairs) on which we train the logistic regression model, thus obtaining the final regression parameter values. This regression model is used to calculate the probability score for all the entity mentions in the crawled text obtained from the URL pairs in the given test dataset.

For the Stage 2 sub-problem of entity disambiguation, we split the given training data into training and development datasets to perform 5-fold cross validation using XGBoost tree ensemble method. First, we made use of the TF-IDF feature vector obtained from the given URL and its page title. In the second attempt, we added the bag of words TF-IDF feature vector as described in Section 2.2.1. Next, we added the feature vector containing TF-IDF of the top-3 entity mentions for both URLs. Finally, we added the anchor text feature vector.

The trained model is used for predictions corresponding to the public leaderboard which contains 50% of the total data. Finally, at the end of the competition, the predictions are measured against the remaining 50% of data which corresponds to the private leaderboard. The results obtained by using the aforementioned features is shown in Table 1. Standard precision, recall and F1-score metrics are used to report the prediction results.

4 Discussion

Our system performs well on both public and private leaderboards. Table 1 shows that a collective use of contextual features in and around the entities leads to an increase in the F1-score. In our system, we make use of TF-IDF of top-20 words and a bag of words approach to train the system. As compared to using just the URL and page title features, the bag of words led to an increment of 5.69% F1-score on the public leaderboard. Next, we identify top-3 entity mentions as the most probable endpoints for a given URL. This gives us a high confidence in disambiguation as most of the URLs are characterized by their top-3 entity mentions. An incorporation of this entity feature has led to an increment of 4.14% F1-score on the public leaderboard. Additionally, it has increased the system recall by a significant 12.14%. Finally, anchor texts prove to be informative features and provide another 3.74% improvement on F1-score. Our system does well in classifying most of the URL pairs as referring to the same underlying entity. However, it does not perform well in certain cases:

- **Lack of identified entities** - There are cases in which the crawled URL text contains just one entity which is usually the name of a person or organization. With no further information about that entity mention, our system fails to leverage the strength of contextual features and is unable to disambiguate the entities, e.g., the URL `www.imdb.com/name/nm5513294` refers to a person named “Johnny Dwyer“. There is no more information about that person on this URL. Its corresponding URL in the given pair is a LinkedIn profile and refers to a person named “Johnny Dwyer” who is an author based in New York. The gold annotations indicate that our system scores a false negative on such URLs.

- **Website search results** - Some URLs refer to search results within a website, which provides a listing of all articles containing an entity mention. While we tackle this problem by avoiding the URLs for news websites in a way so as to prune them for terms like “news” and “topic” as described in Section 2.1.1, there are few cases which were missed, e.g., the URL `deadline.com/tag/secrets-lies` refers to all the articles with a tag of secrets-lies. Our system gives a false positive for the disambiguation of this URL with the Twitter URL of the TV show "Secrets and Lies".

- **Dynamic URLs** - There are some dynamic URLs in the given dataset. A dynamic URL changes with time, i.e., either the contents of that URL change over time or the URL becomes void after some time. Since such URLs do not contain any information, our system is not able to disambiguate them to their valid static URL counterparts.

5 Conclusion

Disambiguating entities referred by web endpoints is an important and challenging problem which gives us insights to an important concept of knowledge base discovery and creation. In this paper, we described our system, which ranked the best with an F1-score of 86.02% in the official private leaderboard of the ALTA 2016 shared task. Our solution was based on a supervised classification method using gradient boosted trees which exploited contextual entity-level features.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, pages 722–735.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 101–110.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Andrew Chisholm, Hugo Australia, Will Radford, and Ben Hachey. 2016a. Discovering entity knowledge bases on the web. *Proceedings of AKBC*, pages 7–11.

Andrew Chisholm, Ben Hachey, and Diego Molla. 2016b. Overview of the 2016 alta shared task: Cross-kb coreference. In *Proceedings of the Australasian Language Technology Workshop 2016*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716, June.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web*, pages 100–110.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

Stephen Guo, Ming-Wei Chang, and Emre Kıcıman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of NAACL-HLT*, pages 1020–1030.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 457–466.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 509–518.

Daniel Myers and James W. McGuffee. 2015. Choosing scrapy. *J. Comput. Sci. Coll.*, 31(1):83–89, October.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. 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*, pages 1375–1384.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, Feb.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706.