ALTA Shared-task 2016: Filter and Match Approach to Pair-wise Web URI Linking

S. Shivashankar
Department of CIS,
University of Melbourne
shivashankar@student.unimelb.edu.au

Yitong Li
Department of CIS,
University of Melbourne
yitongl4@student.unimelb.edu.au

Afshin Rahimi
Department of CIS,
University of Melbourne
arahimi@student.unimelb.edu.au

Abstract

This paper describes the method and results of our approach, submitted as team 'NLPCruise' at ALTA shared task 2016. The goal of the shared task is to predict whether two given web Uniform Resource Identifiers (URIs) correspond to the same entity or not. Retrieving the URI content in addition to the dataset provided, we built a two stage filter and match technique that utilises search engine scores, semantic similarity and machine translation features. Our model achieved an $F1$ score of 0.85714 on the public test-set and ranked second finally on the private leaderboard.

1 Introduction

In general, establishing a mapping from entities in a knowledge base to URI end-points is a useful task both to collate information about entities and to disambiguate them. Typically, semantic sources such as Wikipedia, DBPedia are used as end-points. Although they provide a rich context for entities, they do not achieve sufficient recall over different domains and entities. On the other hand, domain-specific sources such as DBLP, IMDb or MusicBrainz cover only a single target domain (e.g. movies, music) well but aren't useful in other domains. To benefit from both general purpose and domain-specific knowledge bases, one could use WebKB (URIs) as end-points. In-order to do that one must infer its existence on the web. For every entity endpoint we discover, we may recover thousands of entity mentions via inlinks. While the effectiveness of inlink-driven entity disambiguation is known for a single KB setting, this can be extended to leverage inlinks across a collection of automatically discovered web KBs (Chisholm et al., 2016). Thus in this task, we classify whether a pair of URIs correspond to the same entity or not.

Similar tasks were addressed in other shared tasks/challenges, such as Web People Search task (WePS [1]), defined as the problem of organizing web search results for a given person name. At a more generic level, TREC relevance feedback track[2] had multiple tasks related to relevance classification for a set of documents given a query, which we can treat as similar problems to ours when formulated as pair-wise binary document classification.

The goal of ALTA shared task 2016 is to determine whether two URLs refer to the same underlying entity or not. For example, `http://www.nytimes.com/topic/person/barack-obama` and `https://twitter.com/BarackObama` refer to the same person, but `https://twitter.com/realDonaldTrump` and `https://www.instagram.com/ivankatrump` refer to different entities.

2 Data preparation

The original data includes URLs of entities which link to the HTML pages, titles of the HTML pages, and snippets or brief descriptions of pages. Titles and snippets were obtained using BING search API, where the input query was the named entity. The top two URLs from BING search are the URL pairs given for classification. Instances containing non-representative web-pages were removed, more details can be found here [3]. In addition to the given data, we automatically fetched the whole HTML page content using *xdotool*[4]. We removed HTML tags and Java scripts to clean the downloaded HTML documents. Now we have two lev-

[1] http://nlp.uned.es/weps/weps-3

[2] http://trec.nist.gov/data/relevance.feedback.html

[3] https://inclass.kaggle.com/c/alta-2016-challenge/forums/t/23480/how-was-the-data-obtained

[4] `http://www.semicomplete.com/projects/xdotool/`

S. Shivashankar, Yitong Li and Afshin Rahimi. 2016. Filter and Match Approach to Pair-wise Web URI Linking. In *Proceedings of Australasian Language Technology Association Workshop*, pages 182−186.

els of data: title & snippets at the sentence-level and cleaned HTML text at document-level.

3 Proposed Approach

Our proposed approach is shown in Figure 1. We employ filtering at the first step which includes pattern matching heuristics for both named entity and URL pairs. Pairs of URLs that are not disproved to be the same entities in the filtering step are considered for further analysis. The next step involves three different classification models: (a) based on Bing search results obtained by querying named entity from URL_a and domain name of URL_b and vice versa (b) building a semantic similarity based classifier using short-text (title and/or snippet) (c) using complete HTML content, we compute distributed representation based similarity scores, MT scores, Inverse Document Frequency (IDF) based scores, simple word-count and document length based measures and use them as features in a RandomForest binary classifier.

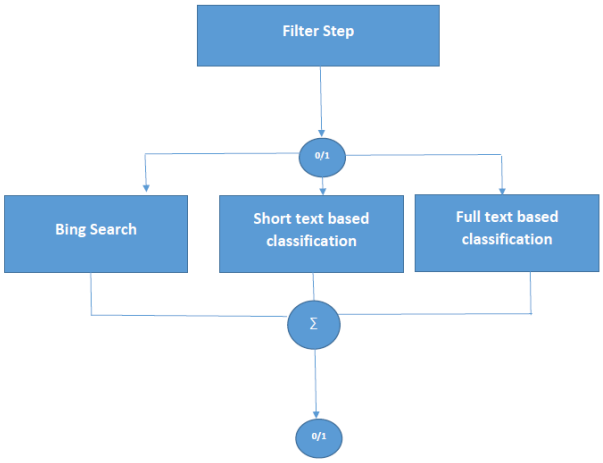

Figure 1: Proposed Approach

3.1 Filtering Step: Named Entity and URL matching

We leverage the title text for getting the named entity by considering words before delimiters such as $'|'$, '-' and extracting the capitalized words from the beginning until any special character. If there are no $'|'$ or '-' then we use Stanford NER tagger to annotate PER, ORG and LOC entities. Then if the names from both the URLs match by a minimum length of 2 and a match score of $\frac{n-1}{n}$, the pair is considered as a 'pass' for further analysis.

For URL match, we consider only pairs from the same domain. We compare path length of URL (number of sub-directories), and match names which can be either of the last two strings (non numeric) after $'/'$. If a URL pair doesn't pass through either of the criteria: named entity or URL matching (if both length and name match fails), then it is disapproved for further processing. Over the test set, we observed that 20% of the records were filtered out (classified as not correspond to the same entities) in this step.

3.2 Bing Search based Classification

3.2.1 Query Construction

We construct queries for Bing Search[5] by using named entities from URL_a and domain name of URL_b. We repeat the same with named entities from URL_b and domain name of URL_a. For instance, for URL pairs `www.imperial.ac.uk/people/f.allen` (URL_a) and `https://www.linkedin.com/in/franklin-allen-0557906` (URL_b), we create queries as "Franklin Allen LinkedIn", "Franklin Allen Brevan Howard Centre at Imperial College London LinkedIn" and "Franklin Allen Finance and Economics LinkedIn". Here, the name "Franklin Allen" is obtained from title & snippet of URL_a. Similarly other context phrases such as *Howard Centre at Imperial College London* and *Finance and Economics* are extracted from URL_a. We combine name and context words from URL_a with domain (in this case LinkedIn) of URL_b for constructing queries. We repeat the same with name and context from URL_b and domain name of URL_a. Named entities are fetched as given in 3.1 - using heuristics (for name) and Stanford NER output (for PER, ORG, LOC). If there are no named entities for querying, we use capitalized chunks of words. For example, "Shark Tank", "America's Most Wanted" and "America Fights Back" are obtained as capitalized chunks from title & snippet of `www.imdb.com/name/nm5507573`, since there were no context words such as ORG, LOC obtained using Stanford NER. Note, for capitalized chunks of text we break the chunks if there are any special characters such as ",","-":","!", and so on. Shorter context words are preferred and any common words are removed from longer phrases. Also, for URLs from Twitter, we fetch location information using Twitter API[6].

[5]https://datamarket.azure.com/dataset/bing/search
[6]http://twitter4j.org/en/

3.2.2 Match Score Computation

After querying on Bing Search engine with name and context words from URL_a and domain name from URL_b, we check if URL_b is present among top 10 search results (we refer to this as a 'hit'). We repeat similar querying with name and context words from URL_b and domain name from URL_a. We compute $P(match|URL\ Pair)$ as the ratio between number of hits and number of queries ($\frac{number\ of\ hits}{number\ of\ queries}$). We refer to above probability score as P, and $Q = P(mismatch|URLPair) = 1 - P$. If $P = Q$, we consider length of queries to weigh the hits. In the following formulation $length(q)$ is the length of query q (number of words in a query), $HITS(q)$ is a Boolean value that indicates if that query was a hit or not,

$$P(match|URL\ Pair) = \frac{\Sigma_q length(q) \times HITS(q)}{\Sigma_q length(q)}$$

If there is a tie still, then it is broken by using the prior for a given pair of domains, for instance $P(match|IMDb, LinkedIn)$.

3.3 Short-text similarity

We set a threshold for classification at 50%, assuming the data is balanced, using semantic similarity and MT based similarity scores (after geometric mean). The predictions are combined by complete consensus for class 0 (mismatch) and atleast one vote for class 1 (match). The intuition is that we expect the filtering step would have removed a good number of mismatches in the previous step.

3.3.1 Semantic Similarity

We use combined title & snippet short text to compute the semantic similarity of pairs using Dandelion API [7]. It is claimed to work well for short text. Here the words are mapped to a Wikipedia like knowledge base and similarity is computed using the mapped vectors.

3.3.2 Machine Translation based Similarity

We compute Machine Translation (MT) evaluation metrics between two short texts. If two URLs refer to the same entity, intuitively, the scores should indicate one text being a paraphrase of another. We score the similarity for both snippets using MT evaluation metrics, including BLEU (BiLingual Evaluation Understudy) (Papineni et al., 2002), METEOR (Metric for Evaluation of Translation with Explicit ORdering) (Banerjee and Lavie, 2005), TER (Translation Error Rate) (Snover et al., 2006), at token-level. In general, MT evaluation metrics are designed to assess whether the output of a MT system is semantically equivalent to a set of reference translations. MT scores are combined using geometric mean ($combined\ score = \sqrt[3]{BLEU * METEOR * TER}$).

[7] https://dandelion.eu/semantic-text/text-similarity-demo/

3.4 Document-level similarity

3.4.1 Machine Translation based Similarity

We also compute MT scores between two documents. Similar to short text setup, we calculate BLEU, METEOR and TER metrics and use them together with distributed representations similarity scores (3.4.2 and 3.4.3) and vector space models (3.4.4).

3.4.2 Distributed Similarity (Word2Vec) based Scores

We calculate the document representation by averaging the pre-trained word embeddings, which are generated by Word2Vec. And then, we compute the cosine similarity between document representations for both training and test pairs as the feature.

3.4.3 Job Descriptions Similarity

Intuition behind using job similarity is that we expect it could uncover profession based similarity that can be effective for web KB instances such as LinkedIn, Avvo, IMDb, etc. To find the job similarity of two pieces of text from a pair, we first find the similarity of each piece of text with each of the 1000 job descriptions existing in Occupational Information Network (ONET) (Peterson et al., 2001) resulting in a 1000d vector for each text. Then we use the similarity of the 1000d vectors of the two pieces of text together and use it as a feature for training the classifier. To find the similarity of a piece of text with one of the job descriptions we use the similarity of the average word vectors using Cosine measure.

3.4.4 Similarity Scores with Inverse Document Frequency (IDF)

The cleaned HTML text is still noisy with many web-page items and unimportant context. Aim to reduce the influence of unrelated tokens, we use

basic idf scores to quantify the importance of tokens (Wu et al., 2008). The inverse document frequency, defined as

$$idf(t, D) = log \frac{N}{|\{d \in D : t \in d\}|}$$

with $N = |D|$ and training corpus D. The idf is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. Considering the web-page tokens could be repeated more times than entity information, we omit the term frequency score in our method.

First, we build a dictionary mapped with idf scores based on English Europarl corpus (Koehn, 2005). Then, empirically we build two kinds of idf features of the documents:

$$f_1 = \Sigma_{t \in (D_A \cap D_B)} idf(t)$$
$$f_2 = |\{d : d \in (D_A \cap D_B), idf(d) \geq idf_m\}|$$

Where, D_A and D_B are documents of entity A and B, idf_m is the mean of all the idf scores in the dictionary. Furthermore, we normalize f_1 and f_2 by the average document length $l = \sqrt{|D_A| \cdot |D_B|}$. Last, we use both normalized and un-normalized features (4 totally) in our method.

3.5 RandomForest Classifier

We build a RandomForest classifier using 16 document level similarity features computed using MT based techniques - 3 types of scores (BLEU, METEOR, TER) for translating both ways (text 'A' to 'B and vice versa), that gives 6 scores totally; distributed word similarity (word2vec) & job similarity based scores - 2 totally; IDF based scores - 4 totally; word count and document length for both URLs using complete HTML text - 4 totally. Since the document level text can be noisy, we consider only high confident (> 0.7) predictions of this model in the next stage for overall prediction.

4 Overall Prediction

The overall prediction is done by combining three classifiers: Bing search, short-text based classification and document level classification. The heuristic used to combine the predictions is given as follows:

- For a given URL pair, if domain specific IDF scores for short-text are high, i.e., it contains common words in all the Title & snippets, for example LinkedIn, then short-text based classification would be unreliable. So we set a threshold empirically using training set, to decide if predictions based on short-text similarity can be used or not.

- Similarly, only confident predictions using RandomForest classifier on document level data are considered reliable.

- With this we consider a URL pair to be 'match' (or 1) if any one of the classifiers predict 'match, and 'not-match' (or 0) if all the classifiers predict 'not-match'. The intuition is that, since we consider a variety of information: short-text, document level information and collaborative information through search engine, we label it as a positive instance, if any of the information/views classifies it as positive, and negative otherwise.

5 Discussion

Combining multiple approaches (lexicon & corpus based) to compute semantic relatedness is an important research topic (Lee et al., 2016). We have employed an approach where the similarity between two pages are obtained by a combination of semantic, machine translation scores (para-phrase) and corpus driven measures such as word2vec. On the test data, we observed the performance of individual models as follows: short-text based semantic similarity (threshold for classification is set at 0.37) gave an F-measure of 0.62, short-text MT features gave a score of 0.53, RandomForest classifier with document level features (distributed similarity, IDF features and MT similarity) gave an F-measure of 0.63. Apart from that, similarity measures based only on distributed models such as averaged word2vec score, job descriptions similarity were not discriminative. Though they are potential directions, it might require more data to re-train the embeddings for this problem. Finally, combined with filtering and Bing Search, we obtained an F-measure of 0.85714 on the public test-set. We believe that this can be more effective if the proposed models are combined together (on a larger dataset) using a stronger ensemble method such as Boosting.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72.

Andrew Chisholm, Will Radford, and Ben Hachey. 2016. Discovering entity knowledge bases on the web. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 7–11.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Yang-Yin Lee, Hao Ke, Hen-Hsen Huang, and Hsin-Hsi Chen. 2016. Combining word embedding and lexical database for semantic relatedness measurement. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 73–74.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Norman G Peterson, Michael D Mumford, Walter C Borman, P Richard Jeanneret, Edwin A Fleishman, Kerry Y Levin, Michael A Campion, Melinda S Mayfield, Frederick P Morgeson, Kenneth Pearlman, et al. 2001. Understanding work using the occupational information network (o* net): Implications for practice and research. *Personnel Psychology*, 54(2):451–492.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):13.