CIRG_IRGDISCO at RepLab2013 Filtering Task: Use of Wikipedia's Graph Structure for Entity Name Disambiguation in Tweets

Muhammad Atif Qureshi[1,2], Arjumand Younus[1,2], Daniel Abril[3], Colm O'Riordan[1], and Gabriella Pasi[2]

[1] Computational Intelligence Research Group, National University of Ireland Galway, Ireland

[2] Information Retrieval Lab,Informatics, Systems and Communication, University of Milan Bicocca, Milan, Italy

[3] Institut d'Investigacio en Intel-ligencia Artificial, Consejo Superior de Investigaciones Cientificas, Campus UAB, Bellateraa, Spain

muhammad.qureshi@nuigalway.ie,arjumand.younus@nuigalway.ie, dabril@iiia.csic.es, colm.oriordan@nuigalway.ie, pasi@disco.unimib.it

Abstract. Social media repositories serve as a significant source of evidence when extracting information related to the reputation of a particular entity (e.g., a particular politician, singer or company). Reputation management experts manually mine the social media repositories (in particular Twitter) for monitoring the reputation of a particular entity. Recently, the online reputation management evaluation campaign known as RepLab at CLEF has turned attention to devising computational methods for facilitating reputation management experts. A quite significant research challenge related to the above issue is to disambiguate tweets with respect to entity names. In fact, finding if a particular tweet is relevant or irrelevant to a particular entity is an important task not satisfactorily solved yet; to address this issue in this paper we use "context phrases" in a tweet and Wikipedia disambiguated articles for a particular entity in an SVM classifier that utilizes features extracted from the Wikipedia graph structure i.e., links into Wikipedia articles and links from Wikipedia articles. Additionally we also use features derived from term-specificity and term-collocation features derived from the Wikipedia article of an entity under investigation. The experimental evaluations do not show a significant improvement over the baseline and other systems outperform our approach; however, manual inspection of feature sets and training data demonstrates the proposed Wikipedia graph-based features may show a promising outcome when used in combination with sophisticated learning algorithms.

1 Introduction

Companies are increasingly making use of social media for broadening their reach and enhancing their marketing. At the same time social media users excessively

voice out their opinions about various entities (e.g. musicians, movies, companies) [5]. This has given birth to a new area within the marketing domain known as "online reputation management" whereby automated methods for monitoring reputation of entities are essential requiring novel computational algorithms to facilitate the work of reputation management experts [1, 3]. This paper describes our experience in devising a completely automated algorithm for dealing with the "entity name disambiguation" challenge in the context of RepLab2013 filtering task [2] where we are given a set of entities and for each entity a set of tweets, which contain some tweets relevant to the entity and some irrelevant ones.

Our approach consists in making use of the knowledge encoded within the Wikipedia graph structure for entity name disambiguation in tweets. We utilize the Wikipedia disambiguation pages for an entity to determine the amount of disambiguation within a particular tweet while at the same time proposing a technique on top of Wikipedia graph structure to determine context in a tweet[4] Although the experimental results do not show a striking performance over the baseline, we argue that the use of Wikipedia graph structure for entity name disambiguation in tweets is a promising direction to pursue.

2 Related Work

There has been an increasing interest in research on applying natural language processing techniques to tweets over the past few years. However, in spite of the immense significance of extracting commercially useful information from tweets, the amount of research dedicated to company name disambiguation in tweets is very limited. The only two serious efforts which have been undertaken to stimulate this research task are represented by the WePS online reputation management evaluation campaign at CLEF 2010 [1], and by the RepLab online reputation management evaluation campaign at CLEF 2012 [3].

The best two teams in the WePS online reputation management evaluation campaign were LSIR-EPFL [7] and ITC-UT [8]. The LSIR-EPFL system builds profiles for each company relying on external resources such as WordNet or the company homepage in addition to a manual list of keywords for the company and the most frequent unrelated senses for the company name. The profiles are then used for extraction of tweet-specific features for use in an SVM classifier. The ITC-UT system is based on a two-step algorithm. In the first step, the algorithm categorizes queries by predicting the class of each company ("organization-like names" or "general-word like names") using a Naive Bayes classifier with six binary features (for example, is the query an acronym?, is the query an entry of a dictionary? etc.). They use thresholds manually set by looking at the training data results for this categorization. The second step consists in categorizing the tweets using a set of heuristics. Despite showing promising results, the two systems LSIR-EPFL and ITC-UT indicate heavy reliance on manual selection of both terms and thresholds for the company name disambiguation task.

[4] This is a huge research challenge within itself given the huge noise and less amount of text in tweets.

During the RepLab2012 online reputation management evaluation campaign, the best performing team relied on hand-coded rules [3] for the filtering task. Here we have defined a completely automatic algorithm for this task that relies on Wikipedia graph structure as an external knowledge resource of evidence. The method is unique in that it does not require any sort of manual keywords or hand-coded rules.

3 Methodology

3.1 Background

The underlying filtering algorithm makes use of the encyclopedic structure in Wikipedia; more specifically the knowledge encoded in Wikipedia's graph structure is utilized for the classification of tweets as relevant or irrelevant with respect to a particular entity. Wikipedia is organized into categories in a taxonomy-like[5] structure (see Figure 2). Each Wikipedia category can have an arbitrary number of subcategories as well as being mentioned inside an arbitrary number of supercategories (e.g., category C_4 in Figure 1 is a subcategory of C_2 and C_3, and a supercategory of C_5, C_6 and C_7.) Furthermore, in Wikipedia each article can belong to an arbitrary number of categories, where each category is a kind of semantic tag for that article [10]. As an example, in Figure 2, article A_1 belongs to categories C_1 and C_{10}, article A_2 belongs to categories C_3 and C_4, while article A_3 belongs to categories C_4 and C_7. It can be seen that the articles and Wikipedia Category Graph are interlinked and our algorithm makes use of these interlinks for the task of entity name disambiguation within tweets.

3.2 Wikipedia-Based Feature Set

Our proposed approach involves a two-step method for entity name disambiguation. In the first step we determine the context phrases within a tweet using an approach similar to Meij et al. [6]. In the second step we use the link structure of Wikipedia to extract a rich feature set which enables us to perform the disambiguation task.

Context phrase extraction is performed by the generation of possible n-grams within phrase chunks of a tweet[6]. Similar to the technique in [6][7] we then reduce candidate phrases extracted from a tweet to those that have a match in Wikipedia article titles. The reduced set of phrases extracted from a tweet are referred to as *ContextPhrases*.

[5] We say taxonomy-like because it is not strictly hierarchical due to the presence of cycles in the Wikipedia category graph.

[6] We do not perform n-gram generation for the complete tweet but instead treat a tweet as a composition of phrase chunks with boundaries such as commas, semicolons, sentence terminators etc. along with other tweet-specific markers such as @, RT etc.

[7] We differ in that we do not apply supervised machine learning for reduction of candidate phrases.

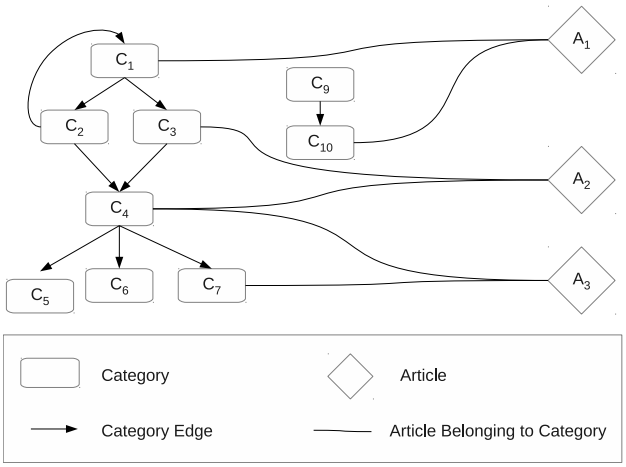

Fig. 1. Wikipedia Category Graph Structure along with Wikipedia Articles

As mentioned in Section 3.1 a significant aspect of our proposed approach is the Wikipedia graph structure; more specifically links between categories and

Feature	Description
$Intersection_{duplication}$	No. of intersections between *inlinks*, *outlinks* and *inlinks+outlinks* sets of e and p without removing duplicated articles
$NormalizedIntersection_{duplication}$	No. of intersections between *inlinks*, *outlinks* and *inlinks+outlinks* sets of e and p without removing duplicated articles and normalized by total number of articles in the sets
$Intersection_{noduplication}$	No. of intersections between *inlinks*, *outlinks* and *inlinks+outlinks* sets of e and p after removing duplicated articles
$NormalizedIntersection_{noduplication}$	No. of intersections between *inlinks*, *outlinks* and *inlinks+outlinks* sets of e and p without removing duplicated articles and normalized by total number of articles in the sets
$Ratio_{inlink:outlink}$	Ratio between articles in *inlinks* to articles in *outlinks*

Table 1. Rich feature set for entity name disambiguation in tweets on top of Wikiped Article Link Structure

articles and within articles are used as the fundamental building block for extraction of Wikipedia graph-specific features. At the first level, we use the parent Wikipedia article for the entity under investigation[8] and extract its parent categories PC_{entity} from which we manually chose categories related to the entity under investigation. Sub-categories are then extracted from PC_{entity} up to a hop count of two; finally all articles belonging to these sub-categories are marked as being related to the entity under investigation and we refer to the set of these articles as $ARelated_{entity}$.

The final step consists of constructing an information table of Wikipedia-based features as follows:

- We extract the disambiguation pages for the entity under investigation and the context phrases extracted in the first step. For each of these we then find the sets of Wikipedia articles in *inlinks*, *outlinks*, and *inlinks+outlinks*. More specifically for each disambiguated Wikipedia article for the entity say e_d and each context phrase p in set *ContextPhrases*, we extract the following sets of Wikipedia articles
  - Wikipedia articles linking to e_d and p referred to as *inlinks*
  - Wikipedia articles linking from e_d and p referred to as *outlinks*
  - Wikipedia articles linking to and from e_d and p referred to as *inlinks+outlinks*
- Using information in sets *inlinks*, *outlinks* and *inlinks+outlinks* the features shown in Table 1 are constructed.
- Corresponding to each feature in Table 1 is a boolean feature that reflects a mapping between the numerical feature to articles in $ARelated_{entity}$. This mapping is constructed after taking average scores across all context phrases in a tweet and chosing the disambiguated Wikipedia article with highest score; if the mapping is to an entity in $ARelated_{entity}$ we chose the value of this feature as "1" and "0" otherwise.

3.3 Additional Features

We also use five additional features and these were obtained from our system [9] used for the last RepLab online reputation management evaluation campaign in 2012 [3]. The technique described in [9] is a two-pass approach where the first pass uses term specificity scores of concept terms (i.e., terms in infoboxes corresponding to the Wikipedia article of the entity, proper nouns *"NNP"* appearing in the Wikipedia article of the entity[9]), and the second pass utilizes a score propagation mechanism where terms co-located with concept terms are assigned a new score for re-computation of a score for each tweet. Further, the following additional scores were also used in our submission for RepLab2012 where our team was the second best amongst the participating teams:

[8] The parent Wikipedia article for each entity is given as part of the dataset for this task.

[9] These are obtained after applying Stanford POS tagger to the Wikipedia article of the entity

Table 2. Results of Filtering Task of RepLab 2013

Team	Reliability	Sensitivity	F(R,S)
popstar_2	0.729	0.451	0.489
popstar_3	0.764	0.440	0.480
popstar_7	0.759	0.428	0.470
popstar_8	0.589	0.444	0.448
SZTE_NLP_7	0.599	0.444	0.439
SZTE_NLP_10	0.547	0.428	0.407
SZTE_NLP_5	0.508	0.429	0.3911
SZTE_NLP_1	0.491	0.429	0.3910
SZTE_NLP_8	0.507	0.428	0.3893
SZTE_NLP_4	0.480	0.429	0.3889
SZTE_NLP_6	0.517	0.428	0.3886
SZTE_NLP_3	0.496	0.425	0.3882
SZTE_NLP_2	0.492	0.426	0.387
lia_1	0.658	0.357	0.381
SZTE_NLP_9	0.519	0.416	0.380
UAMCLYR_4	0.555	0.401	0.379
UAMCLYR_1	0.631	0.400	0.375
lia_6	0.619	0.331	0.341
UNED_ORM_2	0.425	0.384	0.338
BASELINE	0.490	0.320	0.326
UAMCLYR_3	0.697	0.303	0.322
Daedalus_1	0.353	0.448	0.321
Daedalus_3	0.349	0.443	0.318
lia_10	0.680	0.291	0.312
lia_9	0.680	0.282	0.302
UNED_ORM_2	0.473	0.327	0.3018
Daedalus_4	0.302	0.474	0.297
Daedalus_2	0.299	0.479	0.2963
lia_8	0.687	0.266	0.2962
UAMCLYR_2	0.573	0.313	0.292
lia_5	0.489	0.310	0.289
lia_4	0.489	0.310	0.289
UAMCLYR_5	0.569	0.307	0.286
popstar_5	0.521	0.267	0.282
CIRG_IRDISCO_4	0.341	0.329	0.2724
lia_2	0.423	0.331	0.2720

– POS tag of the company name occurring within the tweets
– URL occurring within the tweets
– Hashtag occurring within the tweets

Note that the score of the first pass, score of the second pass, POS tag of company name in the tweet, URL occurring in the tweet and hashtag occuring in the tweet are used as features in our system for RepLab2013.

3.4 Machine Learning and Experimental Runs

Using the feature sets described in Section 3.2 and 3.3, we train a support vector machine over the training data and then use it to predict labels for the test data. We perform six machine learning runs as follows:

1. For the first run, we use all features i.e. both Wikipedia graph-based features and additional score-based features of Section 3.2 and 3.3 whilst training SVM per entity
2. For the second run, we use only Wikipedia graph-based features of Section 3.2 whilst training SVM per entity
3. For the third run, we use only the score-based features of Section 3.3 whilst training SVM per entity
4. For the fourth run, we use all features i.e. both Wikipedia graph-based features and additional score-based features of Section 3.2 and 3.3 whilst training SVM per categories i.e. combining all tweets related to a a particular category into one training and one test set
5. For the fifth run, we use only Wikipedia graph-based features of Section 3.2 whilst training SVM per categories i.e. combining all tweets related to a a particular category into one training and one test set
6. For the sixth run, we use only the score-based features of Section 3.3 whilst training SVM per categories i.e. combining all tweets related to a a particular category into one training and one test set

4 Experimental Results

4.1 Dataset

We performed our experiments by using the data set provided by the organizers of RepLab 2013 [2]. In this data set 61 entities were provided, and for each entity at least 2200 tweets were collected: the first 700 constituted the training set, and the rest served as the test set. Furthermore, the entities are grouped into categories based on their type and the four types distributed as part of RepLab2013 are as follows: 1) automatives, 2) banking, 3) universities, and 4) music.

4.2 Evaluation Measures

The measures used to the evaluation purposes are Reliability and Sensitivity, which are described in detail in [4]. In the case of filtering, the measures of Reliability and Sensitivity are equivalent to the product of precision scores over positive and negative classes (reliability) and the product of recall scores (sensitivity). The property that makes them particularly suitable for the filtering problem is that they are strict with respect to standard measures, i.e., a high value according to Reliability and Sensitivity implies a high value in all standard measures.

4.3 Results

Table 2 presents a snapshot of the official results for the filtering task of RepLab 2013, where CIRG_IRDISCO is the name of our team. As can be seen from Table 2, out of a total of 11 participating teams in RepLab2013 filtering task 6 teams outperform our best run along with the baseline system. We believe this to be a consequence of a considerably high amount of skewness in the training set of tweets. Most of the tweets contained a high percentage of related tweets which affects the performance of learning algorithms such as support vector machines.

5 Future Work

Despite the unfavorable outcome of the RepLab2013 filtering task for our runs, we see significant value in the graph-based features mined from Wikipedia article inlinks and outlinks. Manual inspection of feature set shows an obvious difference of inlink and outlink intersections for the related and non-related tweets. As future work we aim to investigate the value of Wikipedia graph-based features when used in combination with sophisticated learning algorithms.

References

1. E. Amigó, J. Artiles, J. Gonzalo, D. Spina, B. Liu, and A. Corujo. WePS3 Evaluation Campaign: Overview of the On-line Reputation Management Task. In *2nd Web People Search Evaluation Workshop (WePS 2010), CLEF 2010 Conference, Padova Italy*, 2010.
2. E. Amigo, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martin, E. Meij, M. de Rijke, and D. Spina. Overview of replab 2013: Evaluating online reputation monitoring systems. In *Fourth International Conference of the CLEF initiative, CLEF 2013, Valencia, Spain. Proceedings*, Springer LNCS, 2013.
3. E. Amigó, A. Corujo, J. Gonzalo, E. Meij, and M. d. Rijke. Overview of replab 2012: Evaluating online reputation management systems. In *CLEF 2012 Labs and Workshop Notebook Papers*, 2012.
4. E. Amigó, J. Gonzalo, and F. Verdejo. A General Evaluation Measure for Document Organization Tasks. In *Proceedings SIGIR 2013*, July.

5. C. Dellarocas, N. F. Awad, and X. M. Zhang. Exploring the value of online reviews to organizations: Implications for revenue forecasting and planning. In *MANAGEMENT SCIENCE*, pages 1407–1424, 2003.
6. E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 563–572, New York, NY, USA, 2012. ACM.
7. S. R. Yerva, Z. Miklós, and K. Aberer. What have fruits to do with technology?: the case of orange, blackberry and apple. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 48:1–48:10, New York, NY, USA, 2011. ACM.
8. M. Yoshida, S. Matsushima, S. Ono, and I. Sato. Itc-ut: Tweet categorization by query categorization for on-line reputation management.
9. A. Younus, C. O'Riordan, and G. Pasi. Cirgdisco at replab2012 filtering task: A two-pass approach for company name disambiguation in tweets. In *CLEF 2012 Evaluation Labs and Workshop*, 2012.
10. T. Zesch and I. Gurevych. Analysis of the Wikipedia Category Graph for NLP Applications. In *Proceedings of the TextGraphs-2 Workshop (NAACL-HLT)*, 2007.