A Generic Open World Named Entity Disambiguation Approach for Tweets

Mena B. Habib[1] and Maurice van Keulen[1]
[1] *Faculty of EEMCS, University of Twente, Enschede, The Netherlands*
{*m.b.habib, m.vankeulen*}*@ewi.utwente.nl*

Keywords: Named Entity Disambiguation, Social Media, Twitter.

Abstract: Social media is a rich source of information. To make use of this information, it is sometimes required to extract and disambiguate named entities. In this paper, we focus on named entity disambiguation (NED) in twitter messages. NED in tweets is challenging in two ways. First, the limited length of Tweet makes it hard to have enough context while many disambiguation techniques depend on it. The second is that many named entities in tweets do not exist in a knowledge base (KB). We share ideas from information retrieval (IR) and NED to propose solutions for both challenges. For the first problem we make use of the gregarious nature of tweets to get enough context needed for disambiguation. For the second problem we look for an alternative home page if there is no Wikipedia page represents the entity. Given a mention, we obtain a list of Wikipedia candidates from YAGO KB in addition to top ranked pages from Google search engine. We use Support Vector Machine (SVM) to rank the candidate pages to find the best representative entities. Experiments conducted on two data sets show better disambiguation results compared with the baselines and a competitor.

1 INTRODUCTION

1.1 Overview

The rapid growth in IT in the last two decades has led to a growth in the amount of information available on the World Wide Web. A new style for exchanging and sharing information is short text. Examples for this style of text are tweets, social networks statuses, SMSs, and chat messages. In this paper, we use twitter messages as an example of short informal context.

Twitter is an important source for continuously and instantly updated information. The average number of tweets exceeds 140 million tweet per day sent by over 200 million users around the world. These numbers are growing exponentially [1]. This huge number of tweets contains a large amount of unstructured information about users, locations, events, etc.

Information Extraction (IE) is the research field that enables the use of such a vast amount of unstructured distributed information in a structured way. IE systems analyze human language text in order to extract information about different types of events, entities, or relationships. Named entity disambiguation (NED) is the task of exploring which correct person, place, event, etc. is referred to by a mention. Wikipedia articles are widely used as entities' references. For example, the mention '*Victoria*' may refer to one of many entities like '`http://en.wikipedia.org/wiki/Victoria_(Australia)`' or '`http://en.wikipedia.org/wiki/Queen_Victoria`'. According to Yago KB (Suchanek et al., 2007) the mention '*Victoria*' may refer to one of 188 entities in Wikipedia.

1.2 Challenges

NED in Tweets is challenging. Here we summarize the challenges of that problem:

- The limited length (140 characters) of Tweets forces the senders to provide dense information. Users resort to acronyms to reserve space. Informal language is another way to express more information in less space. All of these problems makes the disambiguation more complex. For example, case 1 in table 1 shows two abbreviations ('*Qld*' and '*Vic*'). It is hard to infer their entities without extra information.
- The limited coverage of KB is another challenge facing NED. According to (Lin et al., 2012), 5

[1] `http://www.marketinggum.com/twitter-statistics-2011-updated-stats/`

Table 1: Some challenging cases for NED in Tweets (mentions are written in bold).

Case #	Tweet Content
1	**Qld** flood victims donate to **Vic** bushfire appeal
2	**Laelith Demonia** has just defeated **liwanu Hird**. Career wins is 575, career losses is 966.
3	Adding **Win7Beta**, **Win2008**, and **Vista** x64 and x86 images to munin. #wds
4	"Even Writers Can Help..An Appeal For **Australian** Bushfire Victims" http://cli.gs/Zs8zL2

million out of 15 million mentions on the web could not be linked to Wikipedia. This means that relying only on KB for NED leads to around 33% loss in disambiguated entities. This percentage becomes higher on twitter because of its social nature where people talk more about non famous entities. For example, case 2 in table 1 contains two mentions for two users on '*My Second Life*' social network. One would never find their entities in a KB but their profile pages ('`https://my.secondlife.com/laelith.demonia`' and '`https://my.secondlife.com/liwanu.hird`') can be easily found by any search engine.

- Named entity (NE) representation in KB implies another NED challenge. Yago KB uses Wikipedia anchor text as possible mention representation for named entities. However, there might be more representations that do not appear in Wikipedia anchor text. Either because of misspelling or because of a new abbreviation of the entity. For example, in case 3 in table 1, the mentions '*Win7Beta*' and '*Win2008*' are not representing any entity in YAGO KB although they refer to the entities '`http://en.wikipedia.org/wiki/Windows_7`' and '`http://en.wikipedia.org/wiki/Windows_Server_2008`' respectively.

- The process of NED involves degrees of uncertainty. For example, case 4 in table 1, it is hard to asses whether '*Australian*' should refer to '`http://en.wikipedia.org/wiki/Australia`' or '`http://en.wikipedia.org/wiki/Australian_people`'. Both might be correct. This is why we believe that it is better to provide a list of ranked candidates instead of selecting only one candidate for each mention.

- A final challenge is the update of the KBs. For example, the page of '*Barack Obama*' on Wikipedia was created on 18 March 2004. Before that date '*Barack Obama* was a member of the Illinois Senate and you could find his profile page on '`http://www.ilga.gov/senate/Senator.asp?MemberID=747`'. It is very common on social networks that users talk about some non famous entity who might become later a public figure.

1.3 Our Approach

According to a literature survey (see section 2), almost all researchers use KBs entities as references for NED. Some of those researchers assign *null* to mentions with no possible reference entity and others assign an entity to a mention once it is in the dictionary containing all candidates for surface strings even if the correct one is not in the entity repository. Furthermore, researchers who studied NED in Tweets are mostly entity oriented (i.e. given an entity like '*Apple Inc*', it is required to classify the mention '*Apple*' if it is a correct representative for that entity or not).

In our opinion, for the NED task in Tweets, it is necessary to have a generic system that doesn't rely only on the closed world of KBs in the disambiguation process. We also believe that the NED task involves degrees of uncertainty. In this paper, we propose a generic open world NED approach that shares ideas from NED and IR.

Given a tweet mention, we get a set of possible entity candidates' home pages by querying YAGO KB and Google search engine. We query Google to get possible candidate entities' home pages. We enrich the candidate list by querying YAGO KB to get Wikipedia articles' candidates.

For each candidate we extract a set of context and URL features. Context features (like language model and tweet-document overlapped terms) measure the context similarity between mention and entity candidates. URL features (like path length and mention-URL string similarity) measure how likely the candidate URL could be a representative to the entity home page. In addition we use the prior probability of the entity from YAGO KB. An SVM is trained on the aforementioned features and used to rank all candidate pages.

Wikipedia pages and home pages are different in their characteristics. Wikipedia pages tend to be long, while home pages tend to have short content. Sometimes it has no content at all but a title and a flash introduction. For this reason we train the SVM to distinguish between three types of entity pages, a Wikipedia page (Wiki entity), a Non-Wikipedia home page (Non-Wiki entity), and a non relevant page.

Furthermore, we suggested an approach to enrich the context of the mention by adding frequent terms

from other targeted tweets. Targeted tweets are a set of tweets talking about same event. This approach improves the recognition of NonWiki entities.

We conduct experiments on two different datasets of tweets having different characteristics. Our approach achieves better disambiguation results on both sets compared with the baselines and a competitor.

1.4 Contributions

The paper makes the following contributions:

- We propose a generic approach for NED in Tweets for any named entity (not entity oriented).
- Mentions are disambiguated by assigning them to either a Wikipedia article or a home page.
- Instead of just selecting one entity for each mention we provide a ranked list of possible entities.
- We improve NED quality in Tweets by making use of the gregarious nature of targeted tweets to get enough context needed for disambiguation.

1.5 Paper structure

The rest of the paper is organized as follows. Section 2 presents related work on NED in both formal text and Tweets. Section 3 presents our generic approach for NED in Tweets. In section 4, we describe the experimental setup, present its results, and discuss some observations and their consequences. Finally, conclusions and future work are presented in section 5.

2 RELATED WORK

NED in web documents is a topic that is well covered in literature. Several approaches use Wikipedia or a KB derived form Wikipedia (like DBPedia and YAGO) as entity store to look up the suitable entity for a mention. (Cucerzan, 2007) proposes a large-scale system for disambiguating named entities based on information extracted from Wikipedia. The system employs a vast amount of contextual and category information for better disambiguation results. (Kulkarni et al., 2009) introduce the importance of entity-entity coherence measure in disambiguation. Similarly, (Hoffart et al., 2011) combine three measures: the prior probability of an entity being mentioned, the similarity between the contexts of a mention and a candidate entity, as well as the coherence among candidate entities for all mentions together. AIDA [2] (Yosef et al., 2011) is a system built on (Hoffart et al., 2011)'s approach. We used AIDA as a competitor in our paper.

Ad-hoc (entity oriented) NED represents another direction in NED research. Given a set of predefined entities and candidate mentions, it determines which ones are true mentions of the given entities. An example of such approach is the work done by (Wang et al., 2012).

NED in Tweets has attracted researchers recently. Most of these researches investigate the problem of entity oriented disambiguation. Within this theme, (Spina et al., 2011), (Yerva et al., 2012) and (Delgado et al., 2012) focus on the task of filtering Twitter posts containing a given company name, depending of whether the post is actually related with the company or not. They develop a set of features (co-occurrence, Web-based features, Collection-based features) to find keywords for positive and negative cases. Similarly, (Christoforaki et al., 2011) propose a topic centric entity extraction system where interesting entities pertaining to a topic are mined and extracted from short messages and returned as search results on the topic.

A supervised approach for real time NED in tweets is proposed by (Davis et al., 2012). They focused on the problem of continually monitoring the Twitter stream and predicting whether an incoming message containing mentions indeed refers to a predefined entity or not. The authors propose a three-stage pipeline technique. In the first stage, filtering rules (colocations, users, hash tags) are used to identify clearly positive examples of messages truly mentioning the real world entities. These messages are given as input to an Expectation-Maximization method on the second stage, which produces training information to be used during the last stage. Finally, on the last stage they use the training set produced by the previous stage to classify unlabeled messages in real time. Another real time analysis tool proposed by (Steiner et al., 2013). The authors provide a browser extension which is based on a combination of several third party NLP APIs in order to add more semantics and annotations to Twitter and Facebook micro-posts.

Similar to our problem, the problem of entity home page finding was part of TREC web and entity tracks. The task is to extract target entity and find its home page given an input entity, the type of the target entity and the relationship between the input and the target entity. One of the proposed approaches for this task was (Westerveld et al., 2002). The authors combine content information with other sources as diverse as inlinks, URLs and anchors to find entry page. Another approach for entity home page recognition was

[2] https://d5gate.ag5.mpi-sb.mpg.de/webaida/

introduced by (Li et al., 2009). It selects the features of link or web page content, and constructs entity homepage classifiers by using three kinds of machine learning algorithms of Logistic, SVM, AdaBoost to discover the optimal entity homepage.

Although the TREC problem looks similar to ours, the tweets' short informal nature makes it more tricky to find entity reference page. Moreover, distinguishing Wikipedia pages for Wiki entities from home pages for Non-Wiki entites adds another challenge to our problem.

3 OUR GENERIC OPEN WORLD APPROACH

We can conclude from the previous section that almost all NED approaches in tweets are entity oriented. In contrast, we present a generic open world approach for NED for any named entity based on the mention context and with support from targeted tweets if available.

First of all let us formalize the problem. Given a mention m_i that belongs to tweet t, the goal is to find a ranked list of entities' home pages e_{ij} that m_i represents. We make use of the context of the mention $\{w\} = \{m_i, w_1, w_2, ..w_n\}$ to find the best entity candidate. $\{w\}$ is the set of words in the Tweet after removing the stop words. A set of features is extracted from each e_{ij} measuring how relative is it to m_i and its context. An SVM is trained over training set of manually annotated mentions and used for ranking of entity pages for unseen mentions.

Figure 1 illustrates the whole process of NED in Tweets. The system is composed of the three modules; the matcher, the feature extractor, and the SVM ranker.

3.1 Matcher

This module contains two submodules: Google API, and YAGO KB. Google API is a service provided by Google to enable developers from using Google products from their applications. YAGO KB is built on Wikipedia. It contains more than 447 million facts for 9.8 million entities. A fact is a tuple representing a relation between two entities. YAGO has about 100 relation types, such as `hasWonPrize`, `isKnownFor`, and `isLocatedIn` . Furthermore, it contains relation types connecting mentions to entities such as `hasPreferredName`, `means`, and `isCalled`. The `means` relation represents the relation between the entity and all possible mention representations in wikipedia. For example, the mentions {"*Chris Ronaldo*", "*Christiano*", "*Golden Boy*", "*Cristiano Ronaldo dos Santos Aveiro*"} and many more are all related to the entity "`http://en.wikipedia.org/wiki/Cristiano_Ronaldo`" through the `means` relation.

This module takes the mention m_i and looks for its appropriate web pages using Google API. A list of top 18 web pages retrieved by Google is crawled. To enlarge the search space, we query YAGO KB for possible entities for that mention. Instead of taking all candidate entities related to that mention, we just take the set of candidates with top prior probabilities. Prior probability represents the popularity for mapping a name to an entity. YAGO calculates those prior by counting, for each mention that constitutes an anchor text in Wikipedia, how often it refers to a particular entity. We sort the entities in descending order according to their prior probability. We select the top entities satisfying the following condition:

$$\frac{Prior(e_{ij})}{Maximum(Prior(e_{ij}))} > 0.2 \qquad (1)$$

In this way we consider a set of most probable entities regardless of their count instead of just considering fixed number of top entities.

For all the YAGO selected entities we add their Wikipedia articles to the set of Google retrieved web pages to form our search space for the best candidates for the input mention.

After crawling the candidate pages we apply a wrapper to extract its title, description, keywords and textual content. For this task we used HtmlUnit library[3].

3.2 Feature Extractor

This module is responsible for extracting a set of contextual and URL features that give the SVM indicators on how likely the candidate entity page could be a representative to the mention. The mention tweet is tokenized with a special tweet tokenizer (Gimpel et al., 2011). Similarly, other target tweets (revolving the same event as the mention tweet) are tokenized and top frequent k words are added to the mention context. Only proper nouns and nouns are considered according to the part of speech tags (POS) generated by a special tweet POS tagger (Gimpel et al., 2011). Target tweets can be obtained by considering tweets with the same hashtag. In this paper, we just use the target tweets as provided in one of the two datasets we used in the experiments.

On the candidate pages side, for each candidate page we extract the following set of features:

[3] http://htmlunit.sourceforge.net/

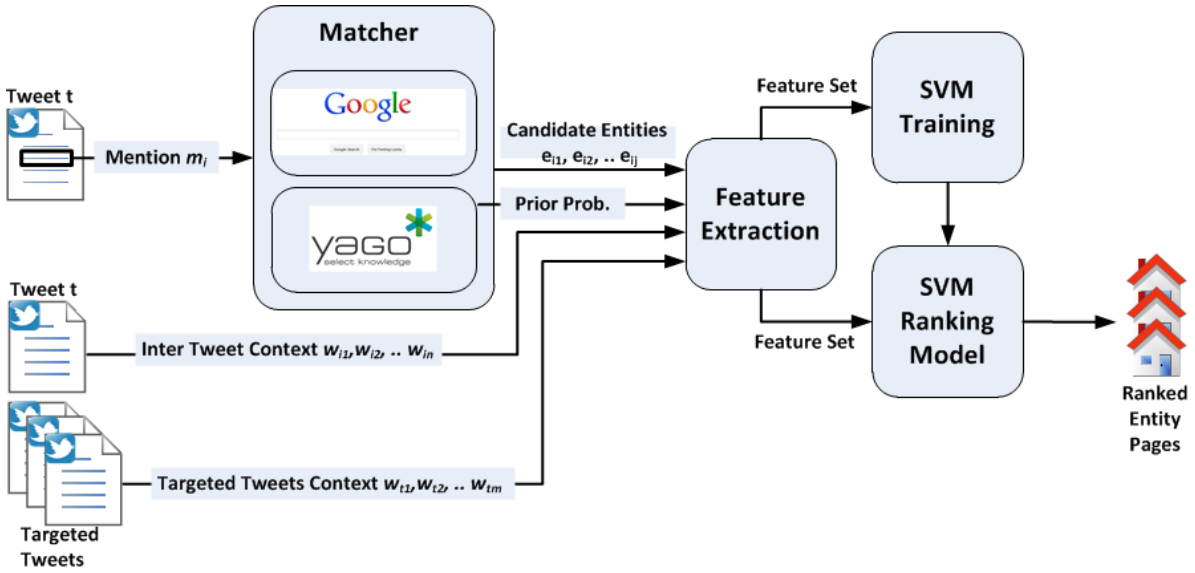


Figure 1: System Architecture.

Table 2: URL features.

Feature Name	Feature Description
URL Length	The length of URL.
Mention-URL Similarity	String similarity between the mention and the URL domain name (for non Wikipedia pages) or the Wikipedia entity name (for Wikipedia pages) based on Dice Coefficient Strategy (Dice, 1945).
Is Mention Contained	Whether or not the mention is contained in the whole URL.
Google Page Rank	The page order as retrieved by Google. Wikipedia pages added from YAGO are assigned a rank after all Google retrieved pages.
Title Keywords	Whether or not page title contains keywords like ('Official', or 'Home page').
#Slashes	Path length of the page (i.e number of slashes in the URL).

- **Language Model (LM):** We used a smoothed unigram LM (Zhai and Lafferty, 2001). We treat the mention along with its tweet keywords as a query and the entity pages as documents. The probability of a document being relevant to the query is calculated as follows:

$$logP(q|d) = \sum_{w \in q,d} log \frac{P_s(w|d)}{\alpha_d P(w|c)} + \sum_{w \in q} logP(w|c) + nlog\alpha_d \quad (2)$$

where $q = \{m_i, w_{i1}, ..w_{in}\}$, d is the e_{ij} candidate page, c is the collection of all the candidate pages for m_i, n is the query length and α_d is document length normalization factor, $P(w|c)$ is the collection LM and $P_s(w|d)$ is the Dirichlet conjugate prior (MacKay and Peto, 1994). These probabilities can be calculated as follows:

$$P(w|c) = \frac{tf(w,c)}{c_s} \quad (3)$$

$$P_s(w|d) = \frac{tf(w,d) + \mu P(w|c)}{|D| + \mu} \quad (4)$$

where tf is the term frequency of a word w in a document d or in the entire collection c, c_s is raw collection size (total number of tokens in the collection) and μ is a smoothing parameter that is calculated as the average document length in the collection c.

We calculated a separate LM for each of the entity pages parts (the title, description, keywords, and content).

- **Tweet-Page Overlap:** The difference in length between Wikipedia pages and non Wikipedia pages in addition to the document length nor-

malization in the LM led to favor short documents (non Wikipedia pages) over long documents (Wikipedia pages). This is why we looked for another feature that does not favor documents based on its length. The feature Tweet-Page Overlap is inspired by Jaccard distance with disregarding lengths. This feature represents the count of the overlapped words between the query q and the document d. It can be calculated as follows:

$$Overlap(q,d) = |q \cap d|$$

Again 4 versions of this feature are calculated for pages title, description, keywords, and content.

- **Entity Prior Probability:** It is a value provided by YAGO KB as described in section 3.1. Only Wikipedia pages have Prior Probabilities. Non Wikipedia pages are just assigned zero for this feature.

In addition to the context features we also extract a set of URL features shown in table 2.

3.3 SVM Ranker

After extracting the aforementioned set of features, an SVM classifier (Chang and Lin, 2011) with RBF kernel function is trained to rank candidate entities of a mention. The SVM is trained on three types of entity classes; Wikipedia home page, non Wikipedia home page, and non relevant page. The reason behind this is that the characteristics of Wikipedia home pages and non Wikipedia home pages are different, and we don't want the classifier to get confused. In this way, the classifier would use the best set of features for each of the relevant classes. Wikipedia home pages have rich contents and thus context features would be better for calculating how relevant is the Wikipedia page to the mention context. While non Wikipedia home pages tend to be short and sometimes with almost no content. In this case URL features might be more useful to find the relevant entity page of a mention.

Moreover, we automatically look into the Wikipedia page infobox for a home page URL for the entity. If found, we remove that home page from the candidate list. For example, for the mention '*Barcelona*', if we find among the candidate pages the Wikipedia page '`http://en.wikipedia.org/wiki/FC_Barcelona`' and we find in the infobox of this page that the official site for '*Barcelona*' is '`http://www.fcbarcelona.com/`', we remove the latter page if found among the candidate pages. The idea behind this action is that our training data is annotated by assigning only one entity page for each mention with the priority for Wikipedia pages. We don't want to confuse the classifier by assigning a non relevant class to a home page for one mention and assigning a relevant class for home page of another mention that doesn't have a Wikipedia entity.

The SVM is trained to provide three probabilities for the three mentioned classes. Due to the imbalance in the training data between the first two classes and the third (only one page is assigned to the mention and the rest is treated as non relevant page), the probabilities of majority class (non relevant) are dominating. Dealing with the task as a ranking task instead of hard classification enables us to overcome this problem.

For testing and evaluating, we rank the mentions candidate pages according to the highest probabilities of the two relevant classes. Evaluation is done by looking at the quality of finding the correct entity page of the mention at top k rank.

3.4 Targeted Tweets

Due to the limitation of tweet context which sometimes affect the disambiguation process, we introduce an improvement by making use of the gregarious nature of tweets. Given a targeted set of tweets (tweets about the same topic), we find the most frequent nouns and add those terms to the context of each tweet in the targeted set. This approach improves the recognition of NonWiki entities as will be shown in the next section.

4 EXPERIMENTAL RESULTS

4.1 Datasets

To validate our approach, we use two twitter datasets[4]. The two datasets are mainly designed for named entity recognition (NER) task. Thus to build our ground truth we only annotated each NE with one appropriate entity page. We gave higher priority to Wikipedia pages. If Wikipedia has no page for the entity we link it to a home page or profile page. The first dataset (Brian Collection) is the one used in (Locke and Martin, 2009). The dataset is composed of four subsets of tweets; one public timeline subset and three subsets of targeted tweets revolving around economic recession, Australian Bushfires and and gas explosion in Bozeman, MT. The other dataset (Mena Collection) is the one used in (Habib and van Keulen, 2012) which is relatively small in size of tweets but rich in number of NEs. It is composed mainly from

[4]Our datasets are available at `https://github.com/badiehm/TwitterNEED`

Table 3: Candidate Pages for the mention "Houston".

http://www.houstontx.gov/
http://en.wikipedia.org/wiki/Houston
http://www.visithoustontexas.com/
http://www.chron.com/
http://www.tripadvisor.com/Tourism-g56003-Houston_Texas-Vacations.html
http://www.forbes.com/places/tx/houston/
http://www.nba.com/rockets/
http://www.uh.edu/
http://www.houstontexans.com/
http://www.houston.org/
http://www.citypass.com/houston
http://www.portofhouston.com/
http://www.hillstone.com/
http://wikitravel.org/en/Houston
http://houston.craigslist.org/
http://houston.astros.mlb.com/

tweeted news about players, celebrities, politics, etc. Statistics about the two data sets are shown in table 4. The two collections are good representative examples for two types of tweets: the formal news titles tweets (Mena Collection) and the users targeted tweets that discuss some events (Brian Collection).

4.2 Experimental Setup

Our evaluation measure is the accuracy of finding the correct entity page of a mention at rank k. We consider only top 5 ranks. The reason behind focusing on recall instead of precision is that we can't consider other retrieved pages as a non-relevant (false positives). In some cases, there may exist more than one relevant page among the candidate pages for a given mention. So that, as we link each mention to only one entity page, it is not fair to consider other pages as a non relevant pages. For example, table 3 shows some candidate pages for the mention '*Houston*'. Although we link this mention to the Wikipedia page `http://en.wikipedia.org/wiki/Houston`, we could not consider other pages (such as `http://www.houstontx.gov/` and http://wikitravel.org/en/Houston) that appear in the top k ranks as non-relevant pages.

All our experiments are done through a 4-fold cross validation approach for training and testing the SVM.

4.3 Baselines and Upper bounds

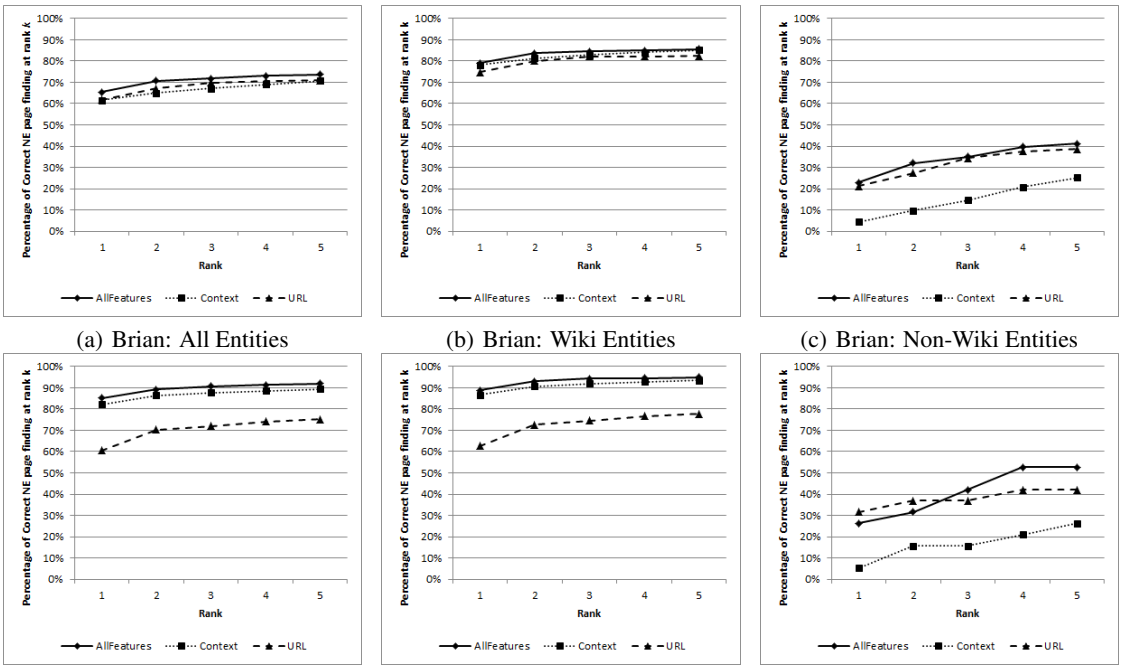
Table 5 shows our baselines and upper bounds in terms of the percentage of correctly finding the entity page of a mention. Three baselines are defined. The first is *Prior*, which represents the disambiguation results if we just pick the YAGO entity with the highest prior for a given mention. The second is the *AIDA* disambiguation system. We used the system's RMI to disambiguate mentions. The third is *Google 1st rank* which represents the results if we picked the Google 1st ranked page result for the input mention. It might be surprising that *AIDA* gives worse results than one of its components which is *Prior*. The reason behind this is that *AIDA* matching of mentions is case sensitive and thus could not find entities for lower case

Table 4: Datasets Statistics.

	Brian Col.	**Mena Col.**
#Tweets	1603	162
#Mentions	1585	510
#Wiki Entities	1233(78%)	483(94%)
#Non-Wiki Entities	274(17%)	19(4%)
#Mentions with no Entity	78(5%)	8(2%)
#Avg Google rank for correct entity	9	5

Table 5: Baselines and Upper bounds.

	Brian Col.	**Mena Col.**
Prior	846(53%)	394(77%)
AIDA	766(48%)	389(76%)
Google 1st rank	269(17%)	197(39%)
YAGO coverage	990(62%)	449(88%)
Google coverage for:		
All entities	1218(77%)	476(93%)
Wiki entities	1077(87%)	462(96%)
Non-Wiki entities	141(51%)	14(74%)



(a) Brian: All Entities (b) Brian: Wiki Entities (c) Brian: Non-Wiki Entities

(d) Mena: All Entities (e) Mena: Wiki Entities (f) Mena: Non-Wiki Entities

Figure 2: Disambiguation results at rank k using different feature sets.

mentions. It was not possible to turn all mentions to initials upper case because some mentions should be in all upper case to get matched (like 'USA'). For *Prior*, we do the match case insensitively. *AIDA* and *Prior* are upper bounded by the YAGO coverage for mentions entity. Coverage means how much mention-entity pairs of our ground truth exist in the KB. Note that more mentions might have a Wikipedia entity but it is not covered in YAGO because it doesn't have the proper surface mention (like 'Win7Beta').

On the other hand, we have an upper bound we can not exceed. The set of candidates retrieved by Google and enriched through KB does not cover our ground truth completely. Hence, we could not exceed that upper bound.

4.4 Feature Evaluation

To evaluate the importance of each of the two feature sets used, we conduct an experiment to measure the effect of each feature set on the disambiguation results. Figure 2 shows the disambiguation results on our datasets using each of the introduced feature sets. It also shows the effect of each feature sets on both types of entities, Wiki and Non-Wiki.

Figures 2(b) and 2(e) show that context features are more effective than URL features in finding Wiki entities. On the other side, figures 2(c) and 2(f) show the superiority of URL features over context features in finding Non-Wiki entities.

Although Wikipedia URLs are normally quite informative, the context features have more data to be investigated and used in the selection and ranking of candidate pages than the URL features. Furthermore, some Wiki URLs are not informative for the given mention. For example, the mention '*Qld*' refers to the Wikipedia entity '`http://en.wikipedia.org/wiki/Queensland`' which is not informative regarding the input mention. This is why context features are more effective than URL features in finding Wiki entities.

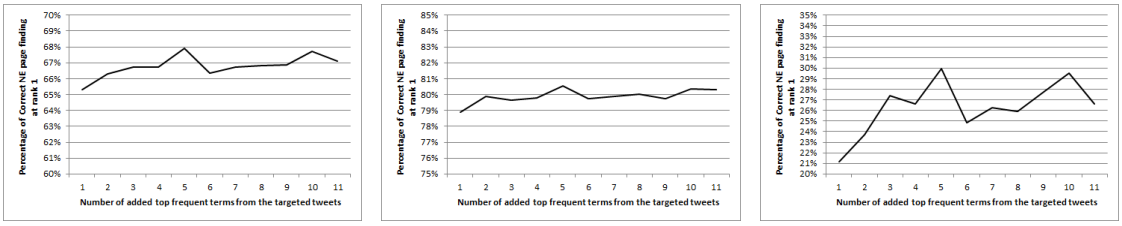
On the other hand, context features are less effective than URL features in finding Non-Wiki entities because many home pages nowadays are either developed in flash or have at least some flash and graphics contents and hence contains less textual content to be used.

All sub figures of figure 2 show that usage of both sets of features yields better entity disambiguation results. The only exception is the first two ranks in figure 2(f). However, it is not an indicator for the failure of our claim as the number of Non-Wiki entities in Mena collection is very small (19 entities).

Compared to table 5, our approach shows improvements on the disambiguation quality for all entities by about 12% on Brian Collection and by about 8% on Mena Collection over the best baseline (prior) at rank $k = 1$. At rank $k = 5$, the improvements over

Table 6: Top 10 frequent terms in Brian col. targeted tweets.

Bozeman Explosion	Australian Bushfires	Economic Recession , MT	Public Timeline
bozeman, montana, bozexplod, mt, twitter, gov, boodles, schweitzer, nw, twitterers	bushfire, sitepoint, appeal, australia, victoria, aussie, coles, brumby, friday, vic	intel, reuters, u.s., fargo, job, san, denver, tuesday, wells, grad	twitter, la, youtube, god, black, mac, tx, iphone, itunes, queen



(a) Brian: All Entities (b) Brian: Wiki Entities (c) Brian: Non-Wiki Entities

Figure 3: Disambiguation results over different top k frequent terms added from targeted tweets.

the best baseline are 21% and 15% respectively.

4.5 Targeted Tweets Improvement

Due to the limitation of tweet context which sometimes affect the disambiguation process, we introduce an improvement by making use of the gregarious nature of tweets. Given a targeted set of tweets (tweets about the same topic), we find the most frequent nouns and add those terms to the context of each tweet in the targeted set. An experiment is performed on Brian collection to study the effect of the frequent terms on the disambiguation results. Table 6 shows top 10 frequent terms in each of the targeted sets. Figure 3 shows the disambiguation results at rank 1 over different top k frequent terms added from targeted tweets. The overall trend is that disambiguation results of all entities are improved by 2% on average by adding frequent terms to tweet context (see figure 3(a)). Non-Wiki entities in figure 3(c) make better use of the frequent terms and achieve improvement of about 4%-5% on average. While Wiki entities in figure 3(b) achieve an improvement of about 1% only. The reason behind this is that Non-Wiki entities' pages are much shorter in contents so that an extra term in the tweet context helps more in finding the correct entity page.

5 CONCLUSIONS AND FUTURE WORK

Named entity disambiguation is an important step to make better use of the unstructured information in tweets. NED in tweets is challenging because of the limited size of tweets and the non existence of many mentioned entities in KBs. In this paper, we introduce a generic open world approach for NED in tweets. The proposed approach is generic as it is not entity oriented. It is also open world because it is not limited by the coverage of a KB. We make use of a KB as well as Google search engine to find candidate set of entities' pages for each mention. Two sets of features (context and URL) are presented for better finding of Wiki and Non-Wiki entity pages. An SVM is used to rank entities' pages instead of assigning only one entity page for each mention. We are inspired by the fact that NED involves degree of uncertainty. We also introduce a method to enrich a mention's context by adding top frequent terms from targeted tweets to the context of the mention.

Results show that context features are more helpful in finding entities with Wikipedia pages, while URL features are more helpful in finding entities with non Wikipedia pages. Adding top frequent terms improves the NED results of Non-Wiki entities by about 4-5%.

For future work, we want to enhance our system to be able also to discover entities with *null* reference. Furthermore, we want to increase the upper bound of candidate pages coverage by re-querying Google search engine for mentions with no suitable candidate pages.

REFERENCES

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Christoforaki, M., Erunse, I., and Yu, C. (2011). Searching social updates for topic-centric entities. In *Proc. of the First International Workshop on Searching and Integrating New Web Data Sources - Very Large Data Search (VLDS)*, pages 34–39.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716.

Davis, A., Veloso, A., da Silva, A. S., Meira, Jr., W., and Laender, A. H. F. (2012). Named entity disambiguation in streaming data. In *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 815–824.

Delgado, A. D., Mart'ınez, R., Pérez Garc'ıa-Plaza, A., and Fresno, V. (2012). Unsupervised Real-Time company name disambiguation in twitter. In *Workshop on Real-Time Analysis and Mining of Social Streams (RAMSS)*, pages 25–28.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 42–47.

Habib, M. B. and van Keulen, M. (2012). Unsupervised improvement of named entity extraction in short informal context using disambiguation clues. In *Proc. of the Workshop on Semantic Web and Information Extraction (SWAIE 2012)*, pages 1–10.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792.

Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S. (2009). Collective annotation of wikipedia entities in web text. In *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 457–466.

Li, L., Yu, Z., Zou, J., Su, L., Xian, Y., and Mao, C. (2009). Research on the method of entity homepage recognition. *Journal of Computational Information Systems (JCIS)*, 5(4):1617–1624.

Lin, T., Mausam, and Etzioni, O. (2012). Entity linking at web scale. In *Proc. of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 84–88.

Locke, B. and Martin, J. (2009). Named entity recognition: Adapting to microblogging. Senior Thesis, University of Colorado.

MacKay, D. J. and Peto, L. C. B. (1994). A hierarchical dirichlet language model. *Natural Language Engineering*, 1:1–19.

Spina, D., Amigó, E., and Gonzalo, J. (2011). Filter keywords and majority class strategies for company name disambiguation in twitter. In *Proc. of the Second international conference on Multilingual and multimodal information access evaluation*, CLEF'11, pages 50–61.

Steiner, T., Verborgh, R., Gabarró Vallés, J., and Van de Walle, R. (2013). Adding meaning to social network microposts via multiple named entity disambiguation apis and tracking their data provenance. *International Journal of Computer Information Systems and Industrial Management*, 5:69–78.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proc. of the 16th international conference on World Wide Web*, WWW '07, pages 697–706.

Wang, C., Chakrabarti, K., Cheng, T., and Chaudhuri, S. (2012). Targeted disambiguation of ad-hoc, homogeneous sets of named entities. In *Proc. of the 21st international conference on World Wide Web*, WWW '12, pages 719–728.

Westerveld, T., Kraaij, W., and Hiemstra, D. (2002). Retrieving web pages using content, links, urls and anchors. In *Tenth Text REtrieval Conference, TREC 2001*, volume SP 500, pages 663–672.

Yerva, S. R., Miklós, Z., and Aberer, K. (2012). Entity-based classification of twitter messages. *IJCSA*, 9(1):88–115.

Yosef, M., Hoffart, J., Bordino, I., Spaniol, M., and Weikum, G. (2011). Aida: An online tool for accurate disambiguation of named entities in text and tables. volume 4, pages 1450–1453.

Zhai, C. and Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 334–342.