Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK
doi:10.3906/elk-1809-167

Exploring the power of supervised learning methods for company name disambiguation in microblog posts

Esma Nafiye POLAT[1,2,*], **Ali ÇAKMAK**[2], **Rabia Nuray TURAN**[3]
[1]R&D Center, Kuveyt Türk Participation Bank, Kocaeli, Turkey
[2]Department of Computer Science, Faculty of Engineering, İstanbul Şehir University, İstanbul, Turkey
[3]Interos Inc., Redwood City, California, USA

Received: 23.09.2018 • **Accepted/Published Online:** 16.10.2019 • **Final Version:** ..201

Abstract: Twitter is an online social networking website where people can post short messages on any subject, and these messages become visible to other users. Users intentionally express their opinions about companies or products via microblogging texts. Analyzing such messages might help explore what customers think about company products, or what the broad feelings of customers are. Identifying tweets referring to products and companies is becoming an important tool recently. However, company names are often vague. Hence, the first step is to locate the messages that are relevant to a company. In this paper, we present a number of supervised learning techniques to decide whether a given tweet is about a company, e.g., whether a message containing the term 'amazon'is related to the company Amazon Inc. or not. Solving this task is challenging in comparison to the classical classification process. The main difficulty with this problem is that tweets and company names include limited information. To make this task tractable, external resources are used to get richer data about a company. More specifically, we generate several profiles for each organization, which contain richer information. Then we perform feature extraction to obtain both numerical and categorical features and we do feature selection to identify the most relevant attributes with our task. Finally, we train several supervised classifiers. Our constructed classifiers and carefully selected features provide high accuracy on the WePS-3 dataset. Our results show considerable improvement of accuracy by 11% over baseline approaches.

Key words: Text processing, name disambiguation, entity resolution, supervised classification, microblogs

1. Introduction

Online reputation management, social media monitoring, and opinion mining from social media are some research areas that focus on users' views in social media on different types of entities. These tasks are challenging as company and product names are often ambiguous. Twitter has experienced fast-paced growth with 326 million monthly active users in 2019 posting more than 500 million tweets and performing more than 2.1 billion search queries each day. In their tweets, users often discuss their good and bad experiences with companies and products so that the other people in their networks may become aware of the brand and its shortcomings or strengths. Hence, the identification of those tweets that refer to particular products and companies is an important tool to manage brand awareness on social media. For instance, Apple Inc. has a name shared by a fruit. The same word may have various other senses based on the context, e.g., as in the story of Adam, Eve, and the serpent. If the content analysis tools that companies use are not able to disambiguate between the content related to the company and the content related to the namesakes, the data they are tracking will be

[*] Correspondence: nafiye.polat@kuveytturk.com.tr

erroneous and irrelevant. In this study, we focus on finding relevant tweets about a company in the context of the WePS-3 dataset, where we are given a set of companies and, for each company, a set of tweets that may or may not be related to the company. The tweets contain the company name as a keyword.

1.1. Approach

The information that tweets contain is usually very brief. This forces us to utilize external resources for an association. More specifically, we produce 10 profiles for each organization, which contain more information. Six of 10 profiles have essentially sets of keywords that are related to the company in some way. On the other hand, the remaining profiles intentionally contain unrelated keywords. In order to construct the profiles, we first exploit Wikipedia and company review pages to gather information about the brand and its namesakes. We construct the profile vectors out of the top N keywords that have the highest weight values (e.g., tf-idf scores). We evaluate the effect of N by considering different values including 100, 250, and 500. Next, each tweet's similarity to these profiles is computed by using cosine similarity. Similarity values together with several extracted categorical features from each tweet are used to create feature vectors per tweet, which are then used in the training and testing of a classification model. Then we build a number of classification models with feature selection. We later employ those models to predict the related tweets for previously unseen companies.

Furthermore, we employ a threshold algorithm [9], simple approach algorithm [22], and Wikipedia two-profile approach [5] as alternative methods. In the threshold algorithm, a similarity threshold between a tweet and a company profile is learned. Then a tweet whose similarity to a company profile is less than the learned threshold is classified as irrelevant. Otherwise, the tweet is classified as relevant to the company. The simple approach algorithm determines the relevancy of a tweet to a company by simply checking if the tweet contains at least one of the company's profile vector terms. Finally, the Wikipedia two-profile approach marks a tweet as relevant to a company if the tweet is more similar to the company's Wikipedia page than the Wikipedia disambiguation page that contains the other meanings of a term in the company name (e.g., Apple).

Lastly, we employ an entity ranking algorithm [1] that uses two language models, namely entity mention language and review language. We obtain the best performance with the company latent semantic indexing profile. We show that our results have considerable improvement of accuracy ranging from 9% to 12% over our baseline algorithm.

1.2. Contributions

1. The primary contribution of this paper is the comparative experimental evaluation of various supervised learning methods with different feature sets for the company name disambiguation problem.

2. Moreover, in addition to what has been commonly used in the literature by similar works, we also employ additional profiles such as the latent semantic indexing profile and noun phrase profile to enrich company information, differently from other approaches.

3. Finally, distinct from other tweet–company classification methods, we adapt the entity ranking algorithm [1] for the tweet classification problem. This algorithm is more efficient than the standard tf-idf approach as illustrated by our experimental results.

2. Related work

2.1. Company disambiguation

The online reputation management task of WePS-3 aims to identify those tweets that are relevant to a given organization. Researchers evaluate their methods using standard training and test datasets. Yerva et al. [2] obtained the best score in the WePS-3 competition. They used a rich variety of company profiles including Homepage, Metadata Profile, Category Profile, GoogleSet Profile, User Feedback Positive Profile, and Negative Feedback Profile. After extracting entity profile features, tweet-specific features, and some heuristic features, the SVM classifier was trained on these features. In the WePS-3 competition, the second most accurate system was developed by the ITC-UT team of Yoshida et al. with accuracy of 0.75. Their intuition was that company names in the dataset are usually either organization-like names (McDonald's) or general word-like names (Pioneer). In a preliminary study [4], we employed a limited set of features and company profiles for company name disambiguation. More specifically, we constructed only two profiles, namely the Company Wikipedia Page Profile and Company Wikipedia Disambiguation Page Profile. The research of [6] utilized the named entities and external data. For certain organization names, they used Wikipedia, DBPedia, and the company homepage. Their system's performance was 0.63. Kalmar [7] concentrated on a bootstrapping method. To classify the tweets, some external data were gathered and the study obtained 0.46 accuracy. Distinct from these studies, we use different approaches like term frequency and a latent semantic indexing weighting scheme, which are used in order to extract company-related keywords. Moreover, we pick company review pages and parse them to obtain company-related keywords. In summary, in this work, we extensively enrich our profile set and propose new classification mechanisms to locate company-related tweets.

Differently from the WePS-3 evaluation campaign, some researchers used the WePS-3 dataset with the purpose of solving company name disambiguation in later years. Yerva et al. [8] constructed user profiles by putting together content from two different social networks, i.e. Twitter and StackOverflow, using the WePS-3 dataset. They demonstrated that the content published on users' social networks may boost the accuracy of the entity disambiguation task considerably. In a related study, Yerva et al. [10] defined a relatedness factor, which is the percentage of tweets that belong to a given company. In another related study [11], Yerva et al. looked into the reasons why some companies underperformed with their previous approaches. By comparing the current profiles to the generated perfect profiles they observed that errors could occur in three different ways: (i) missing words errors, (ii) errors in word weights, and (iii) wrongly placed error words. In [12], the main approach is based on the idea of representing the information of a company in the form of a unique profile. They tested their application with the WePS-3 Online Reputation Management Corpus and obtained 0.69 accuracy. In [13], company tweets in WePS-3 are clustered as true or false according to the term expansion methodology. In order to solve company name ambiguity, they employed a clustering technique different from our classification task.

2.2. Entity disambiguation

The entity disambiguation problem also emerges with regard to other entities. In [14], an in-depth study on existing semantic question answering (SQA) systems was performed to understand the scope of the SQA processes, and the existing disambiguation solutions for SQA systems were summarized. In a related study [15], the authors identified the major problem areas in recognizing named entities in microblog posts and made suggestions on which problems should be further studied to improve the recent techniques. In [16], the authors used a reranking model, which adds collective features that are not considered in other entity linking tasks. Their reranking approach beats the state-of-the-art entity linking strategies. The authors in [17] employed phrases extracted from tweets and Wikipedia articles as features in a random forest classifier, which outperformed their baseline approach to identifying company-related tweets. Additionally, there are crucial studies on the author ambiguity problem [18], web search for individuals [18], and personal name ambiguity problems in emails [19]. The problem that we study in this paper differs from personal ambiguity in that our training and test corpora do not allow to train company-specific models. That is, with the provided training and testing datasets in WePS-3, a single classification model may be built, and the same model is used to make predictions for all the test companies.

2.3. Entity matching

In [20], the SHINE approach was proposed. This approach is based on linking the named entities in Web text with heterogeneous information networks that consist of multitype interconnected objects. The work in [21] concentrated on the likelihood of an attribute reappearing over time. Depending on attribute's past value, an entity might alter its attribute value. It uses temporal information of entity records in the form of time stamps. In [1], the authors focused on review-object matching problems by utilizing just reviews' contexts. If a review mentions an object, each review word is drawn either from an object description or a generic review language that does not depend on the object. We adapt their entity matching approach to determine whether a given tweet is relevant or irrelevant to a company. In our adaptation, we use two language models: one is entity mention language and the other is the review language model. The authors of [22] studied the object matching problem in tweets. Their model depends on the tendency of a user to post an object-related tweet, the reputation of the object, and the geographic distance between a user and an object.

3. Methods

3.1. Problem statement

Given a tweet T and an organization name O, the tweet classification problem is to determine if T is related to O or not. Input tweet data include the following: the tweet id, the organization (entity) name, the tweet query (it is a query for retrieving tweets, corresponding with the organization identifier in the corpus such as delta, ford, best buy, etc.), the tweet content, and the author identifier. We are also provided with the organization name and homepage URL for each association, as well as the tweet annotation. Each of the tweet outputs is annotated with a 'True', 'False', or 'Unknown' label that represents tweet relevancy or irrelevancy to a given company. Tweet and organization name include limited data. Hence, we need to create several profiles for an organization, each of which is either related or unrelated to the company.

3.2. Company profile representation

Table 1 presents each organization as a collection of several profiles, which are summarized in the next section. We represent each profile as a set of weighted keywords.

3.3. Tweet representation

Each tweet is represented as a set of words, i.e. the occurrence of each word is used as a feature in the classifier. We represent the vector for a tweet t that includes n words as:

$$V_i = \text{set}((\text{word}_1, \text{tf.idf}_{word_{1,t}}), ..., (\text{word}_n, \text{tf.idf}_{word_{n,t}})).$$

Table 1. Constructed profile names for company ambiguity problem.

Profile no.	Profile name	Acronym
P_1	Company Wikipedia Page Profile	wikiProf
P_2	Company Wikipedia Disambiguation Page Profile	wikidisambigProf
P_3	Company Home Page Profile	homepageProf
P_4	Company Review Page Profile	reviewProf
P_5	Company Wikipedia Page Noun Phrase Profile	wikinounphraseProf
P_6	Company Wikipedia Disambiguation Page Noun Phrase Profile	wikinounphrasedisambigProf
P_7	Company Wikipedia Page Term Frequency Profile	wikitermfreqProf
P_8	Company Wikipedia Disambiguation Page Term Frequency Profile	wikitermfreqdisambigProf
P_9	Company Wikipedia Page Latent Semantic Indexing Profile	wikilatentProf
P_{10}	Company Wikipedia Disambiguation Page Latent Semantic Indexing Profile	wikilatentdisambigProf

The tf.idf of a word is computed using the following formula:

$$tf.idf_{word_i,t} = \left(tf_{word_i,t} \cdot \log \frac{|D|}{|\{t' \in D | word_i \in t'\}|} \right), \tag{1}$$

where tf is the term frequency and idf is the inverse document frequency calculated on the following corpus, D. We consider each tweet in the training data of a company as a document. As an example for the Alcatel-Lucent company, we have 481 tweets in our training data. Therefore, our corpus D for Alcatel-Lucent contains 481 documents.

3.4. Learning company name disambiguation

We build a set of supervised classifiers on the extracted features. The first step is feature selection that maximizes the accuracy of classification. We train a classifier with selected features using the provided training data.

3.5. Feature extraction

First, we apply some well-known preprocessing techniques to text data, such as removal of stop words, Porter stemming, and verb lemmatization (to eliminate tense differences).

3.5.1. Numerical features

Company Wikipedia Page Profile (wikiProf): We consider that the most important words on a company's Wikipedia pages are the most relevant words to the company and may be utilized to identify the company. In the first step, we download the Wikipedia pages and the pages linked from these Wikipedia pages up to depth = 2. These pages are then parsed and the text in the pages is used to derive the Company Wikipedia Page Profile for the company. In order to generate this profile, we compute the weight of each term using the tf/idf term weighting scheme where tf is the term frequency and idf is the inverse document frequency calculated on the entire corpora. For example, for Alcatel-Lucent, when we download the Wikipedia pages and the pages linked from these Wikipedia pages up to depth = 2, we obtain 174 distinct Web pages. Each Web page is considered as a document. Therefore, our document corpus contains 174 documents for Alcatel-Lucent. In sum, company Wikipedia page profiles consist of <term, tf.idf-weight> pairs for the top N terms. We evaluate different values of N (i.e. 100, 250, 500) in our experiments. For example, for Apple Inc., the first five keywords with their associated weights are:

$$P_{wikiProf} = [('ipad', 0.53), ('mac', 0.48), ('iphone', 0.42),$$
$$('store', 0.36), ('steve', 0.32)].$$

Company Wikipedia Disambiguation Page Profile (wikidisambigProf): We generate another profile for each company using the Wikipedia page for the different meanings of the company name. To find the different meanings of the company name, we use company disambiguation pages. Afterwards, this profile is constructed in a way similar to the above Wikipedia page profile.

Company Review Page Profile (reviewProf): Tweets usually mention companies in an informal context. That is, they do not include complete and clearly identifiable company names. Thus, we consider that if we can capture the informal context/language on a more content-rich platform, we might be able to get better company profiles. To this end, we create review pages-based company profiles. More specifically, we collected reviews from several web sites including Yelp, Pissed Consumer, Amazon CNET, AirlineQuality.com, and TripAdvisor. The same preprocessing steps as for the above profiles are applied to the crawled data. Then the top K (100, 250, and 500) terms that are ranked according to their tf/idf are selected as the Company Review Page Profile. All review pages for a company constitute the document corpus for that company.

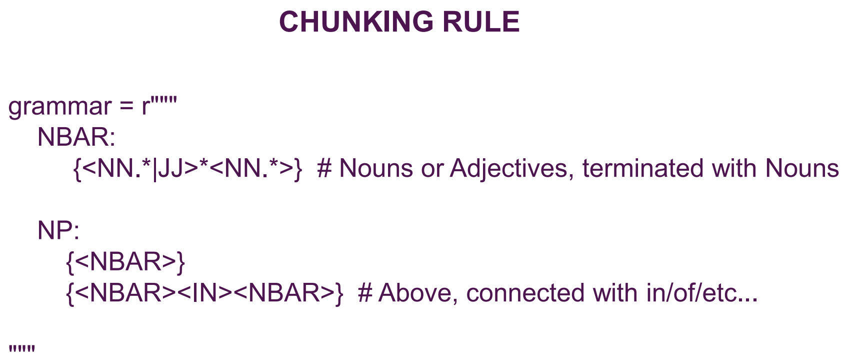
Company Wikipedia Page Noun Phrase Profile (wikinounphraseProf): The above company profiles contain vocabulary from all linguistic categories, such as noun, verb, adverb, etc. We believe that representing a company with its extracted significant nouns may provide a more concise representation of a company. This profile is constructed by considering only the nouns from company Wikipedia pages. Therefore, we eliminate verbs from profiles to bring nouns to the forefront. Grammatically, adjectives and adverbs have meaning when they are used with other words. Since adverbs do not make much contribution to representing company content noticeably, we discard adverbs from company profiles. Hence, we construct a company profile vector including only nouns and noun phrases.

First, we tag the given list of tokens using NLTK's postag module. Then we explore chunking, which segments and labels multitoken sequences. In order to do that, we define a rule that finds a chunk structure in a given text using NLTK's RegexpParser module. The rule is extracting a single noun, noun + noun, and adjective + noun in a given text. The rule that we use is represented in Figure 1.

Formally, the rule says that any number of nouns (NN) or adjectives (JJ) is followed by any number of nouns (NN). Using this grammar rule, a chunk parser is created and a chunk tree is produced. For instance, for the sentence ‘This is the best digital camera’, the chunk tree is constructed as in Figure 2.

For our problem, we feed the obtained tokens to the chunker as a parameter to obtain noun + noun and adjective + noun phrases from the text. Next, we split the obtained phrases into its words. Then we select the top 100, 250, and 500 important words as the company Wikipedia page noun phrase profile vector for Wikipedia company pages.

Company Wikipedia Disambiguation Page Noun Phrase Profile (wikinounphrasedisambigProf): In order to create a profile that is irrelevant to the company and includes noun and noun phrases, we apply the same strategy to Wikipedia disambiguation pages for each corresponding company.



CHUNKING RULE

```
grammar = r"""
   NBAR:
       {<NN.*|JJ>*<NN.*>}  # Nouns or Adjectives, terminated with Nouns

   NP:
      {<NBAR>}
      {<NBAR><IN><NBAR>}  # Above, connected with in/of/etc...

"""
```

Figure 1. Chunking rule for extracting noun + noun phrase.

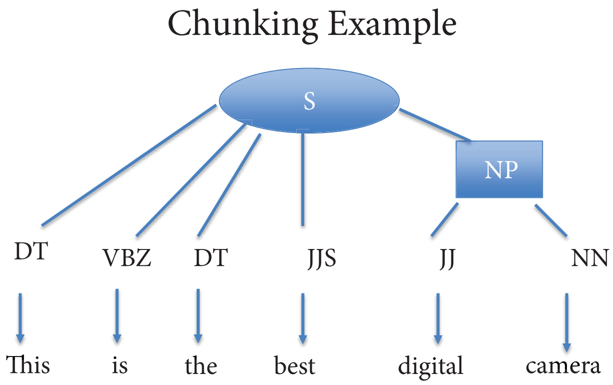


Figure 2. Chunking example for extracting noun + noun phrase.

Company Wikipedia Page Term Frequency Profile (wikitermfreqProf): Term frequency is mostly used in information retrieval and it shows how often a particular word appears in a document. We build this profile out of term frequencies extracted from company Wikipedia pages. That is, company Wikipedia page term frequency profiles consist of <term, tf> pairs for the top N terms. We evaluate different values of N (i.e. 100, 250, 500) in our experiments. Term frequency values are computed as follows, where D is the company Wikipedia page. Please note that we also normalize term frequency with the highest frequency:

$$termfrequency(w_i, D) = \left(\frac{freq(w_i, D)}{maxfreq(w_k, D); \text{for word} w_k \in D} \right).$$

For Apple Inc., the first five keywords based on the Company Wikipedia Page Term Frequency Profile are:

$$P_{wikitermfreqProf} = [('comput', 1.0), ('iphon', 0.7), ('disk', 0.5),$$
$$('system', 0.5), ('macintosh', 0.1)].$$

Company Wikipedia Disambiguation Page Term Frequency Profile (wikitermfreqdisambigProf): This profile is constructed similar to the above ones based on the terms that appear on company Wikipedia disambiguation pages.

Company Wikipedia Page Latent Semantic Indexing Profile (wikilatentProf): Latent semantic indexing (LSI) is an information retrieval method that detects semantically related terms that are latent in document collections. Based on LSI, words that are used in the same contexts tend to have similar meanings. In the first step, a term-document matrix A is generated to represent the occurrences of the m terms within a collection of n documents. In the term-document matrix A, each term is represented by a row, and each document is represented by a column. For a given term i and document j, the cell $a_{i,j}$ in the matrix shows the number of times that term i appears in document j. This matrix is usually very large and sparse. After a document matrix is generated, each cell count is modified using the tf/idf weighting formula given in Eq. (1). Here, we use the same document corpus D as the one used for the Wikipedia Company Page Profile.

As a result, rare words are weighted more heavily than common words. Next, LSI performs singular value decomposition on the matrix A to determine patterns between the document terms and document concepts. We employ LSI on company Wikipedia pages to extract semantically related keywords for each company and build another profile out of these keywords.

Company Wikipedia Disambiguation Page Latent Semantic Indexing Profile (wikilatentdisambigProf): A similar LSI profile is built on the company Wikipedia disambiguation corpus.

Number of alternative meanings: As another feature, we obtain the number of different links from the Wikipedia Disambiguation Page for each company to see how disambiguation intensity influences the related or unrelated tweet evaluation process.

3.5.2. Categorical features

Capital words: Words in capital case are highly likely to be significant words or named entities. We assume that when a tweet has a company name in capitalized form, most probably the tweet is relevant to the company.

Url: URL is also a strong indicator. If the tweet contains a link to the company website or Wiki page, then it is more probable that the tweet is relevant to the company of interest.

Bigram: We propose a rule that if a tweet has its full entity name (as including more than one word), such as “Dunkin’ Donuts”, then it is more likely to be labeled as related to the given company.

Prepositions: The basic English grammar rule is that the prepositions ‘at’, ‘for’, and ‘of’ commonly come in front of organization names. Therefore, we define such information as another feature that would help us determine whether a tweet refers to a given organization or not.

3.5.3. Feature representation

For a given pair of tweet T_x and company C_y, the feature vector is constructed as follows:

$$F(T_x, C_y) = [M_i, H_j]^T, \tag{2}$$

where i ranges from 1 to 10 and j ranges from 1 to 4. M_1 through M_9 are similarity values (computed with cosine similarity) between the vector representation of tweet T_x and the nine different profiles of company C_y as listed in Section 3.5.1. M_{10} is the number of the alternative meanings feature. Lastly, the H_j features are the categorical features as explained in Section 3.5.2.

3.6. Learning tweet classification with supervised classifiers

For the tweet classification problem, we first employ some of the commonly used supervised learning algorithms as listed below:

- Logistic regression with liblinear solver,
- MLPClassifier (activation=relu, initial learning rate=0.28, max iteration=2000,)
- RandomForestClassifier (number of estimators=100, max depth=2),
- Gradient boosting classifier (number of estimators=13, initial learning rate=1.0, max depth=1),
- Naive Bayes algorithm (likelihood of the features is assumed to be Gaussian).

Next, we further employ alternative classification methods as summarized below.

Simple approach algorithm (SAA): In this approach, a company profile vector is generated using the company Wikipedia web page. Next, the top 100 key words that have the highest tf/idf values are obtained. Here, the document corpus D is the same as the one used for the Wikipedia company page profile. Then, if a tweet includes one of the company profile vector terms, the tweet is considered to be relevant to the given company. Otherwise, it is considered as irrelevant. In [22], a similar approach was used in order to select restaurant-related tweets from a tweet corpus. First, the authors picked the top occurring words in the reviews as the keyword set. Then they eliminated a tweet that did not contain any of those keywords from the tweet corpus. Using this approach, they generated a restaurant-related tweet dataset that they used for the entity matching problem.

Entity ranking algorithm (ERA): We employ the language model proposed in [1]. The model incorporates the entity information and a general review language model. For our problem, the goal of this mixture model is to determine whether a given tweet is relevant or irrelevant to a company. Informally, when a tweet t is written about a company, each word in t is drawn either from entity (company) information or generic review language (reviews obtained from several web sites including Yelp, Pissed Consumer, Amazon, CNET, AirlineQuality.com, and TripAdvisor). In our adaptation, we use two language models: one is entity mention language and the other is review language model. Our review language model includes all review data for training and testing companies. Also, entity mention language consists of two profiles: one with Wikipedia company keywords and another with Wikipedia disambiguation company keywords.

Threshold-based classification (TA): We experimentally determine a similarity threshold based on training data. More specifically, the threshold value is determined so as to achieve the highest accuracy on the training set. Then this threshold value is used for unseen testing data. When the computed similarity (e.g., cosine) of a test tweet is below the specified threshold, we consider that the tweet is unrelated to the corresponding company. Otherwise, the tweet is considered to be related. We use the threshold learning algorithm for the above presented profiles. We employ the cosine similarity technique for this purpose.

4. Experiments and evaluation

We use the dataset of the WePS-3 evaluation campaign, which includes two tasks: one focuses on the problem of personal name ambiguity and the other focuses on company name ambiguity. In this study, we deal with the second task. The WePS-3 data include 3 datasets: (i) a trial dataset comprising 24 companies (17 English and 6 Spanish organizations) with 100 tweets for each company, (ii) a training dataset comprising 52 companies with about 400 tweets for each company, and (iii) a test dataset comprising 47 companies with about 400 tweets for each company. We use the WePS training dataset (except those with unknown labels) to train our models, and we later test them on the WePS test dataset (except those with unknown labels). Below, we provide some statistics for training and test datasets:

Training dataset:

- The total number of tweets is 23,789 (positive: 9526, negative: 13,563, unknown: 700).
- The average number of words per tweet is 15.
- In total, 6166 hashtags, 6829 mentions, and 11,384 URLs appear.

Test dataset:

- The total number of tweets is 22,490 (positive: 7717, negative: 13,076, unknown: 1697).
- The average number of words per tweet is 16.
- In total, 6323 hashtags, 6144 mentions, and 10,752 URLs appear.

We explain the evaluation metrics that are commonly used in information retrieval. Precision is the fraction of retrieved instances that are relevant, while recall is the relevant instances that are retrieved. In other words, precision is the measure of the quality demonstrating that the algorithm returns relevant terms more than irrelevant terms; recall is the measure of the quantity demonstrating that the algorithm returns most of the relevant results. For example, for a text search on a set of documents, precision is the number of correct results divided by the number of all returned results, and recall is the number of correct results divided by the number of results that should have been returned.

In the classification task, some terms are used in order to compare the classification results under test with external judgments. The terms “positive” and “negative” show the classifier’s prediction, while “true” and “false” define whether the prediction corresponds to the judgment result. These terms are explained below:

True positive (TP): Tweets that are correctly labeled as belonging to the positive class.

True negative (TN): Tweets that are correctly labeled as belonging to the negative class.

False positive (FP): Tweets that are labeled as belonging to the positive class incorrectly.

False negative (FN): Tweets that are labeled as belonging to the negative class incorrectly.

We use the following metrics to study the performance of our classification process:

$$\begin{aligned}
\text{Accuracy (Acc)} &= (TN + TP)/(TN + TP + FN + FP),\\
\text{Precision}^{+}(P^{+}) &= TP/(TP + FP),\\
\text{Precision}^{-}(P^{-}) &= TN/(TN + FN),\\
\text{Recall}^{+}(R^{+}) &= TP/(TP + FN),\\
\text{Recall}^{-}(R^{-}) &= TN/(TN + FP),\\
F^{+}_{measure}(F^{+}) &= 2 \cdot Precision^{+} \cdot Recall^{+}/(Precision^{+} + Recall^{+}),\\
F^{-}_{measure}(F^{-}) &= 2 \cdot Precision^{-} \cdot Recall^{-}/(Precision^{-} + Recall^{-}).
\end{aligned}$$

At first, a baseline approach is employed using the weighted bag of keywords from Twitter. Binary (1/0) weighting is used for our baseline approach. We train a random-forest classifier (n_estimators=100, max_depth=2,random_state=0) on the training dataset. The classifier's accuracy on the test dataset is 54.4%. This is our baseline approach.

4.1. Feature selection

As shown by the baseline approach, our bag of words method is not well suited as a solution for the company ambiguity problem. The results indicate that the use of external features may be highly instrumental to improving the accuracy. As we discussed in the previous section, we use both numerical and categorical features for feature selection. For brevity, we use the following feature names: $wiki_{cosineSim}$, $disambig_{cosineSim}$, $review_{cosineSim}$, $nounphrase_{cosineSim}$, $disambignounphrase_{cosineSim}$, $termfreq_{cosineSim}$, $disambigtermfreq_{cosineSim}$, $latent_{cosineSim}$, $disambiglatent_{cosineSim}$, $numberofmeanings$, $capital$, $being_{prep}$, url, and bigram.

We apply several feature selection methods for our tweet classification task, namely removing features with low variance, univariate feature selection, recursive feature elimination, and tree-based feature selection. For this purpose, we employ Python's Scikit-Learn library. We obtained the best result with GradientBoostingClassifier using tree-based feature selection, which is a tree-based estimator to compute feature importances. Our selected features are: 1: $wiki_{cosineSim}$, 2: $disambig_{cosineSim}$, 3: $review_{cosineSim}$, 4: $latent_{cosineSim}$, 5: numberofmeanings, and 6: bigram, respectively. The scores for all the employed classification algorithms are presented in Table 2, where each classification algorithm is run with the best feature set that provides the maximum accuracy for that algorithm.

Table 2. Experimental results with both selected numerical and categorical features.

Exp. no.	Estimator	Accur.(%)	P^{+}(%)	R^{+}(%)	F^{+}(%)	P^{-}(%)	R^{-}(%)	F^{-}(%)
1	GradientBoostingClassifier	66.0	52.0	51.0	52.0	61.0	73.0	67.0
2	Multilayer perceptron	63.0	63.0	50.0	56.0	63.0	85.0	72.0
3	RandomForestClassifier	63.0	69.0	45.0	54.0	62.0	91.0	73.0
4	Support vector machine	62.0	69.0	39.0	50.0	60.0	92.0	73.0
5	Logistic regression	62.0	70.0	38.0	49.0	60.0	93.0	73.0
6	Naive Bayes algorithm	60.0	67.0	34.0	45.0	59.0	92.0	72.0

Of 23,789 training tweets, 9526 of them belong to the positive class and 13,563 of them belong to the negative class (700 of them are labeled as unknown). In the test data, the numbers of tweets that belong to positive and negative classes are 7717 and 13,076, respectively. As evident from the dataset statistics, we have more negatively labeled tweets than positively labeled ones. Our results show that the applied machine learning algorithms on unseen test data are more successful in finding irrelevant tweets. Therefore, we have more true negative tweets in comparison to true positive tweets. This increases our negative rates (P, R, and hence F). Moreover, as we have more negatively labeled tweets in the training and test datasets, we have fewer false negatives than false positives. For that reason, the recall for negative samples is higher than the precision for positive ones.

When we omit the feature selection step, the accuracy of GradientBoostingClassifier decreases to 64.9%. That is, feature selection slightly improves the accuracy by about 1%.

4.2. Threshold algorithm

We perform threshold experiments with several company profiles. The results are shown in Table 3.

Table 3. Accuracy for the threshold experiment.

Exp. no.	Profile (learned threshold value)	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
1	reviewProf (0.003)	61.3	60.7	42.5	50.0	66.0	80.3	72.5
2	wikinounphraseProf (0.001)	66.6	66.0	40.5	50.3	66.7	85.3	74.8
3	wikitermfreqProf (0.005)	68.7	67.5	48.0	56.0	69.0	83.4	75.6
4	wikiProf (0.001)	69.9	69.0	51.0	59.0	70.0	84.0	76.0
5	wikilatentProf (0.002)	71.0	73.6	43.3	54.6	68.6	88.8	77.4

The results show that, as opposed to our expectations, review pages (Exp. 1) do not highly overlap with tweet messages. In particular, both true negative and positive values are less than the other profiles. One of the primary reasons is that review pages mostly include terms that belong to daily language. This indicates that the learned cosine similarity threshold value between a tweet vector and a review profile vector is higher than the learned cosine similarity threshold value with wikinounphraseProf (Exp. 2) and wikiProf (Exp. 5) (0.003 vs. 0.001). Thus, our false negative scores increase considerably. The learned threshold value for each profile is given in Table 3 next to the profile name in parentheses.

When we use wikilatentProf for threshold classification, we obtain the best accuracy. LSI is proven to be good at finding semantically related significant keywords in a document corpus, and this is confirmed with higher accuracy results in our experiments, as well. With this profile, the true positives are higher than that of wikiProf (Exp. 5), so the accuracy value is higher, as well.

We further employ majority voting on these profiles with a threshold-based classifier. Majority voting chooses a particular class based on the largest number of votes that it receives from a set of classifiers, which are, in our case, threshold algorithm-based classifiers constructed for each of the profiles separately. The obtained accuracy with those profiles is 71.3%. When we compare this result in Table 4 to those in Table 3, it seems that all profile combinations make approximately 0.3% contribution to the overall classification accuracy.

Table 4. Majority voting on all profiles with threshold-based classifier.

Profile	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
allProfiles	71.3	73.8	47.0	57.4	70.5	88.4	78.4

As in the previous section, our results show that the recall value for the negative class is significantly higher than that of the positive class, while the precision of the negative class is slightly lower than that of the positive class. The primary reason for such a dramatic difference, especially in recall values, may be attributed to the distribution of tweets in negative and positive classes in the training data. That is, in the training data, the majority of the tweets (59%) belong to the negative class. Therefore, the built models tend to predict tweet labels as negative. Thus, we have higher recall values for the negative class.

In summary, we can make the following general comment about our threshold experiment: Our method tends to classify tweets as unrelated for all given profiles. Thus, we get better recall and F-values with negative examples in comparison to the positive examples.

4.3. Simple approach algorithm

The accuracy values for simple approach algorithm are shown in Table 5. Although this approach is very simple, the results are surprisingly comparable to those of other more complicated methods. With this approach, because of the increase in the true positive rate, the number of positive examples is higher than in the earlier experiments. Moreover, the number of true negative examples is as high as those presented in earlier experiments.

Table 5. Accuracy for simple approach algorithm using Wikipedia company profiles.

Prof.	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
wikiProf	69.8	66.7	55.0	60.3	71.3	80.3	76.0

4.4. Wikipedia two-profile approach (WTPA)

In this experiment, we test another approach for labeling tweets. More specifically, for a given tweet, if the cosine similarity between the company's Wikipedia profile vector and the tweet vector is greater than the cosine similarity between the Wikipedia disambiguation profile vector and the tweet vector, the tweet is marked as related to the given company. Otherwise, the tweet is marked as unrelated. The results are shown in Table 6. Although the approach is simpler, we obtain comparable accuracy values as previously observed for simple classifiers.

Table 6. Accuracy with Wikipedia two-profile approach.

Prof.	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
wikiProf+wikidisambigProf	70.6	70.3	85.6	77.2	71.3	49.8	58.7

4.5. Entity ranking algorithm

For this algorithm, we use different profile vectors with different keyword sets. In Table 7 through Table 9, the keyword sets are represented as tuples. The first element of the tuple denotes the number of relevant keywords, and the second element of the tuple represents the number of irrelevant keywords. The highest accuracy value is marked in bold font for each profile.

As seen in the tables, the best accuracy is obtained with the Company Wikipedia Page Latent Semantic Indexing Profile and Company Wikipedia Disambiguation Page Latent Semantic Indexing Profile including 250 related keywords and 250 unrelated keywords (Table 9 - Exp. 4). As we mentioned in the previous section, the latent semantic indexing algorithm is successful in finding highly related terms in a document set using the

Table 7. Entity ranking classification results for company Wikipedia page profile and company Wikipedia disambiguation page profile for different keyword sets.

Exp. no.	Keyword set	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
1	(100,100)	**67.9**	67.0	45.4	54.1	68.2	84.0	75.3
2	(250,100)	67.2	60.7	59.0	59.8	71.2	72.7	71.9
3	(500,100)	64.4	56.0	68.3	61.6	73.1	61.6	66.8
4	(250,250)	65.8	59.9	54.4	57.1	69.4	73.9	71.6
5	(500,500)	65.3	58.2	60.2	59.2	70.8	69.0	69.9

Table 8. Entity ranking classification results for company Wikipedia page term frequency profile and company Wikipedia page term frequency profile for different keyword sets.

Exp. no.	Keyword set	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
1	(100,100)	65.4	60.9	47.9	53.6	67.6	77.9	72.4
2	(250,100)	**66.3**	59.6	59.9	59.8	71.2	70.9	71.0
3	(500,100)	61.8	53.2	69.2	60.2	71.9	56.5	63.3
4	(250,250)	64.8	58.1	56.2	57.1	69.4	71.0	70.2
5	(500,500)	62.1	54.0	60.8	57.2	69.2	63.0	65.9

Table 9. Entity ranking classification results for company Wikipedia page latent semantic indexing profile and company Wikipedia disambiguation page latent semantic indexing profile for different keyword sets.

Exp. no.	Keyword set	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
1	(100,100)	67.7	63.2	54.1	58.3	70.2	77.4	73.6
2	(250,100)	66.9	60.1	61.6	60.9	72.0	70.7	71.4
3	(500,100)	66.0	58.2	65.4	61.6	72.8	66.4	69.5
4	(250,250)	**69.1**	64.5	58.1	61.1	71.9	77.1	74.4
5	(500,500)	68.3	62.1	61.3	61.7	72.5	73.2	72.9

singular value decomposition method. It provides the most correlated words considering the whole Wikipedia document corpus.

4.6. General evaluation

Table 10 compares the accuracy of WePS-3 participants to that of our work. In our study, the threshold algorithm provides the best accuracy (71.4%), which is less than only the LSIR-EPFL and ITC-UT systems. Moreover, several other proposed techniques, such as majority voting, the simple approach algorithm, and the Wikipedia two-profile algorithm, outperform other systems, except LSIR and ITC-UT.

In order to measure the significance of accuracy change in experiments, we perform statistical t-tests, where we consider accuracy change (in comparison to the baseline approach) as significant if the computed P-value is less than 0.05. The t-test results show that the improvement over baseline is significant for our threshold algorithm (P = 0.0024), simple approach algorithm (P = 0.023), and entity ranking algorithm (P = 0.0015).

Table 10. Comparison to other techniques.

System	Accur.(%)	P^+(%)	R^+(%)	F^+(%)	P^-(%)	R^-(%)	F^-(%)
LSIR-EPFL	**83.0**	71.0	74.0	**63.0**	**84.0**	52.0	56.0
ITC-UT	75.0	75.0	54.0	49.0	74.0	60.0	57.0
Our method	71.4	73.9	47.3	57.6	70.5	**88.7**	**78.5**
SINAI	63.0	**84.0**	37.0	29.0	68.0	71.0	53.0
UVA	56.0	47.0	41.0	36.0	60.0	64.0	55.0
KALMAR	46.0	48.0	**75.0**	47.0	65.0	25.0	28.0

5. Conclusion and future work

In this paper, we focus on the problem of company name disambiguation in Twitter messages. We utilize different kinds of external resources to get richer information about companies. We develop a highly accurate classification approach with the help of different entity profiles that are constructed using different data sources. Since the presented method is text-based and does not depend on any HTML structure, this classification system may directly be applied to other social networks or blogs. As part of our future work, we will consider the following items. First, sometimes a keyword might exist in both company-related profiles and unrelated profiles, leading to erroneous results. For example, when we analyze the top 100 keywords of Apple Inc., we observe that "game" and "video" keywords exist in both the company Wikipedia profile and company Wikipedia disambiguation profile. Therefore, as part of our future work, we will investigate how to improve our proposed methods to address the overlapping keyword set. Second, the TD/IDF weighting scheme is widely used but its performance is dominated by language models and embeddings if syntax semantics are considered in evaluations. As an example, Klenin and Botov [3] presented the results of the evaluation of various vector space models, namely TF-IDF, LSA, LDA, averaged Word2vec, and Paragraph2vec, within the context of educational course program documents. Paragraph2Vec, which involves using various neural networks to learn word and document vectors in low-dimensional space approaches, provides the best results for both clustering and classification tasks. Similarly, Word2Vec is a two-layer neural network that uses distributional semantics to learn the correlation between words and their contexts using two architectures, namely continuous bag-of-words and skip-gram. As future work, we plan to generate our profiles based on those algorithms.

References

[1] Dalvi N, Kumar R, Pang B, Tomkins A. Matching reviews to objects using a language model. In: EMNLP Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing; Singapore; 2009. pp. 609-618.

[2] Yerva SR, Zoltan M, Aberer K. It was easy, when apples and blackberries were only fruits. In: CLEF 2010 LABs and Workshops, Notebook Papers; Padua, Italy; 2010. p. 13.

[3] Klenin J, Botov D. Comparison of vector space representations of documents for the task of matching contents of educational course programmes. In: Supplementary Proceedings of the Sixth International Conference on Analysis of Images, Social Networks and Texts; Moscow, Russia; 2017. pp. 79-90.

[4] Polat N. Experiments on company name disambiguation with supervised classification techniques. In: 2013 International Conference on Electronics, Computer and Computation; Ankara, Turkey; 2013. pp. 139-142.

[5] Yano T, Kang M. Taking Advantage of Wikipedia in Natural Language Processing. Technical Report. Pittsburgh, PA, USA: Carnegie Mellon University Language Technologies Institute, 2016.

[6] García-Cumbreras MA, García-Vega M, Martínez-Santiago F, Perea-Ortega JM. Sinai at weps-3: Online reputation management. In: CLEF 2010 LABs and Workshops, Vol. 1176; Padua, Italy; 2010.

[7] Kalmar P. Bootstrapping websites for classification of organization names on twitter. In: 3rd Web People Search Evaluation Workshop; 2010.

[8] Yerva SR, Catasta M, Demartini G, Aberer K. Entity disambiguation in tweets leveraging user social profiles. In: Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration; San Francisco, CA, USA; 2013. pp. 120-128.

[9] Ahmad T, Ramsay A, Ahmed H. CENTEMENT at SemEval-2018 Task 1: Classification of tweets using multiple thresholds with self-correction and weighted conditional probabilities. In: Proceedings of the 12th International Workshop on Semantic Evaluation; New Orleans, LA, USA; 2018. pp. 200-204.

[10] Yerva SR, Miklós Z, Aberer K. What have fruits to do with technology?: The case of orange, blackberry and apple. In: WIMS '11 Proceedings of the International Conference on Web Intelligence, Mining and Semantics; New York, NY, USA; 2011. p. 48.

[11] Yerva SR, Miklós Z, Aberer K. Entity-based classification of twitter messages. International Journal on Computational Science and Applications 2012; 9: 88-115.

[12] Delgado DA, Martinez-Unanue R, Garcia-Plaza AP, Fernandez VF. Unsupervised real-time company name disambiguation in twitter. In: Sixth International AAAI Conference on Weblogs and Social Media; Dublin, Ireland; 2012. pp. 25-28.

[13] Perez-Tellez F, Pinto D, Cardiff J, Rosso P. On the difficulty of clustering company tweets. In: Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents; Toronto, Canada; 2010;. pp. 95-102.

[14] Hazrina S, Sharef NM, Ibrahim H, Murad MAA, Noah SAM. Review on the advancements of disambiguation in semantic question answering system. Information Processing and Management 2017; 53: 52-69.

[15] Derczynski L, Maynard D, Rizzo G, van Erp M, Gorrell G et al. Analysis of named entity recognition and linking for tweets. Information Processing and Management 2015; 51: 32-49.

[16] Zhao G, Wu J, Wang D, Li T. Entity disambiguation to Wikipedia using collective ranking. Information Processing and Management 2016; 52: 1247-1257.

[17] Qureshi MA, O'Riordan C, Pasi G. Exploiting Wikipedia for entity name disambiguation in tweets. In: International Conference on Applications of Natural Language to Data Bases/Information Systems; Montpellier, France; 2014. pp. 184-195.

[18] Kalashnikov DV, Mehrotra S, Chen Z. Exploiting relationships for domain-independent data cleaning. In: SIAM SDM; Newport Beach, CA, USA; 2005. pp. 262-273.

[19] Minkov E, Cohen WW, Ng AY. Contextual search and name disambiguation in email using graphs. In: SIGIR; Seattle, WA, USA; 2006. pp. 27-34.

[20] Shen W, Han J, Wang J. A probabilistic model for linking named entities in web text with heterogeneous information networks. In: SIGMOD '14 Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data; Snowbird, UT, USA; 2014. pp. 1199-1210.

[21] Chiang YH, Doan A, Naughton JF. Modeling entity evolution for temporal record matching. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data; Snowbird, UT, USA; 2014. pp. 1175-1186.

[22] Dalvi N, Kumar R, Pang B. Object matching in tweets with spatial models. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining; Seattle, WA, USA; 2012. pp. 43-52.