It was easy, when apples and blackberries were only fruits

Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer

EPFL IC LSIR
Lausanne, Switzerland
{surenderreddy.yerva, zoltan.miklos, karl.aberer}@epfl.ch

Abstract. Ambiguities in company names are omnipresent. This is not accidental, companies deliberately chose ambiguous brand names, as part of their marketing and branding strategy. This procedure leads to new challenges, when it comes to finding information about the company on the Web. This paper is concerned with the task of classifying Twitter messages, whether they are related to a given company: for example, we classify a set of twitter messages containing a keyword *apple*, whether a message is related to the company Apple Inc. Our technique is essentially an SVM classifier, which uses a simple representation of relevant and irrelevant information in the form of keywords, grouped in specific "profiles". We developed a simple technique to construct such classifiers for previously unseen companies, where no training set is available, by training the meta-features of the classifier with the help of a general test set. Our techniques show high accuracy figures over the WePS-3 dataset.

1 Introduction

Twitter [1] is a popular service where users can share short messages (a.k.a. tweets) on any subject. Twitter is currently one of the most popular sites of the Web, as of February 2010, Twitter users send 50 million messages per day [2]. As users are sharing information on what matters to them, analyzing twitter messages can reveal important social phenomena, indeed there are number of recent works, for example in [11], exploring such information. Clearly, twitter messages are also a rich source for companies, to study the opinions about their products. To perform sentiment analysis or obtain reputation-related information, one needs first to identify the messages which are related to a given company. This is a challenging task on its own as company or product names are often homonyms. This is not accidental, companies deliberately choose such names as part of their branding and marketing strategy. For example, the company Apple Inc. shares its name with the fruit apple, which again could have a number of figurative meanings depending on the context, for example, "knowledge" (Biblical story of Adam, Eve and the serpent) or New York (the Big Apple).

[1] http://twitter.com

[2] http://www.telegraph.co.uk/technology/twitter/7297541/Twitter-users-send-50-million-tweets-per-day.html

In this paper, we focus on how to relate tweets to a company, in the context of the WePS-3 challenge, where we are given a set of companies and for each company a set of tweets, which might or might not be related to the company (i.e. the tweets contain the company name, as a keyword). Constructing such a classifier is a challenging task, as tweet messages are very short (maximum 140 characters), thus they contain very little information, and additionally, tweet messages use a specific language, often with incorrect grammar and specific abbreviations, which are hard to interpret by a computer. To overcome this problem, we constructed profiles for each company, which contain more rich information. For each company, in fact, we constructed several profiles, some of them automatically, some of them manually. The profiles are essentially sets of keywords, which are related to the company in some way. We also created profiles, which explicitly contains unrelated keywords. Our technique is essentially an SVM classifier, which uses this simple representation of relevant and irrelevant information in the "profiles". We developed a simple technique to construct such classifiers for previously unseen companies, where no training set is available, by training the meta-features of the classifier with the help of a general test set, available in WePS-3. Our techniques show high accuracy figures over the WePS-3 dataset.

The rest of the paper is organized as follows. Section 2 gives a more precise problem definition. Section 3 presents our techniques, while Section 4 gives more details on the classification techniques we used. Section 5 gives details on the experimental evaluation of our methods. Section 6 summarizes related work and finally Section 7 concludes the paper.

2 Problem Statement

In this section we formulate the problem and our computational framework more formally. The task is concerned to classify a set of Twitter messages $\Gamma = \{T_1, \ldots, T_n\}$, whether they are related to a given company C. We assume that each message $T_i \in \Gamma$ contains the company name as a sub-string. We say that the message T_i is related to the company C, $related(T_i, C)$, if and only if the Twitter message refers to the company. It can be that a message refers both to the company and also to some other meaning of the company name (or to some other company with the same name), but whenever the message T_i refers to company C we try to classify as TRUE otherwise as FALSE. The task has some other inputs, such as the URL of the company $url(C)$, the language of the webpage, as well as the correct classification for a small number of messages (for some of the companies).

3 Information representation

The tweet messages and company names alone contain very little information to realize the classification task with good accuracy. To overcome this problem, we created profiles for the companies, several profiles for each company. These set of profiles can be seen as a model for the company. In this section, we discuss

how we represent tweet messages and companies and we also discuss how we obtained these profiles. In the the classification task we eventually compare a tweet against the profiles representing the company (see Section 4).

3.1 Tweet Representation

We represented a tweet as a bag of words (unigrams and bigrams). We do not access the tweet messages directly in our classification algorithm, but apply a preprocessing step first, which removes all the stop-words, emoticons, and twitter specific stop-words (such as, for example, RT,@username). We store a stemmed[3] version of keywords (unigrams and bigrams), i.e.

$$T_i = set\{wrd_j\}.$$

3.2 Company Representation

We represent each company as a collection of profiles, formally

$$E^k = \{P_1^k, P_2^k, \ldots, P_n^k\}.$$

Each profile is a set of weighted keywords i.e. $P_i^k = \{wrd_j : wt_j\}$, with $wt_j \geq 0$ for positive evidence and $wt_j < 0$ for negative evidence.

For the tweets classification task, we eventually compare the tweet with the entity (i.e. company) profile. For better classification results, the entity profile should have a good overlap with the tweets. Unfortunately, we do not know the tweet messages in advance, so we tried to create such profiles from alternative sources, independently of the tweet messages. The entity profile should not be too general, because it would result many false positives in the classification and also not too narrow, because then we could miss potential relevant tweets.

We generated most of our profiles automatically, i.e. if one would like to construct a classifier for a previously unseen company, one can automatically generate the profiles. Further, small, manually constructed profiles could further improve the accuracy of the classification, as we explain in Section 5.

In the following we give an overview of the profiles we used, and their construction.

Homepage Profile For each company name, the company homepage URL was provided in the WePS-3 data. To construct the homepage profile, we crawled all the relevant links up to a depth of level d(=2), starting from the given homepage URL. We extracted all the keywords present on the relevant pages, then we removed all the stopwords, finally we stored in the profile the stemmed version of these keywords. From this construction process one would expect that homepage-profile should capture all the important keywords related to the company. However, since the construction is

[3] Porter stemmer from python based natural language toolkit available at http://www.nltk.org

an automated process, it was not always possible to capture good quality representation of the company, for various reasons: the company Webpages use java-scripts, flash, some company pages contain irrelevant links, there are non-standard homepages etc.

Metadata Profile HTML standards provides few meta tags[4], which enables a webpage to list set of keywords that one could associate with the webpage. We collect all such meta keywords in this profile whenever they are present. If these meta-keywords are present in the HTML code, they have high quality, the meta-keywords are highly relevant for the company. On the negative side, only a fraction of webpages have this information available.

Category Profile The category, to which the company belongs, is a good source of relevant information of the company entity. The general terms associated with the category would be a rich representation of the entity. One usually fails to find this kind of keywords in the homepage profile. We make use of wordnet, a network of words, to find all the terms linked to the category keywords. This kind of profile helps us assign keywords like: software,install, update, virus, version, hardware, program, bugs etc to a software company.

GoogleSet/CommonKnowledge Profile GoogleSet is a good source of obtaining "common knowledge" about the company. We make use of GoogleSets[5] to get words closely related to the company name. This helps us identify companies similar to the company under consideration, we get to know the products, competitor names etc. This kind of information is very useful, especially for twitter streams, as many tweets compare companies with others. With this kind of profile, we could for example associate Mozilla, Firefox, Internet Explorer, Safari keywords to Opera Browser entity.

UserFeedback Positive Profile The user himself enters the keywords which he feels are relevant to the company, that we store in the manually constructed UserFeedback profile. In case of companies where sample ground truth is available, we can infer the keywords from the tweets (in the training set) belonging to the company.

UserFeedback Negative Profile The knowledge of the common entities with which the current company entity could be confused, would be a rich source of information, using which one could classify tweets efficiently. The common knowledge that "apple" keyword related to "Apple Inc" company could be interpreted possibly as the fruit, or the New York city etc. This particular profile helps us to collect all the keywords associated with other entities with similar keyword. An automated way of collecting this information would be very helpful, but it is difficult. For now we make use of few sources as an initial step to collect this information. The user himself provides us with this information. Second, the wiki disambiguation pages[6] contains this information, at least for some entities. Finally this information could be

[4] *http* : *//www.w*3*schools.com/html/html_meta.asp*

[5] http://labs.google.com/sets

[6] http://en.wikipedia.org/wiki/Apple_(disambiguation) page contains apple entities

gathered in a dynamic way i.e., using the keywords in all the tweets, that do not belong to the company. This information could also be obtained if we have training set for a particular company with tweets that do not belong to the company entity.

Table 1 shows how an "Apple Inc"[7] company entity is represented using different profiles.

Table 1. Apple Inc Company Profiles

Profile Type	**Keywords**
WebPage	iphone, ipod, mac, safari, ios, iphoto, iwork, leopard, forum, items, employees, itunes, credit, portable, secure, unix, auditing, forums, marketers, browse, dominicana, music, recommend, preview, type, tell, notif, phone, purchase, manuals, updates, fifa, 8GB, 16GB, 32GB,...
HTML Metatag	{empty}
Category	opera, code, brainchild, movie, telecom, cruncher, trade, cathode-ray, paper, freight, keyboard, dbm, merchandise, disk, language, microprocessor, move, web, monitor, diskett, show, figure, instrument, board, lade, digit, good, shipment, food, cpu, moving-picture, fluid, consign, contraband, electronic, volume, peripherals, crt, resolve, yield, server, micro, magazine, dreck, byproduct, spiritualist, telecommunications, manage, commodity, flick, vehicle, set, creation, procedure, consequence, second, design, result, mobile, home, processor, spin-off, wander, analog, transmission, cargo, expert, record, database, tube, payload, state, estimate, intersect, internet, print, factory, contrast, outcome, machine, deliver, effect, job, output, release, turnout, convert, river,...
GoogleSet	itunes, intel, belkin, 512mb, sony, hp, canon, powerpc, mac, apple, iphone, ati, microsoft, ibm,...
User Positive	ipad, imac, iphone, ipod, itouch, itv, iad, itunes, keynote, safari, leopard, tiger, iwork, android, droid, phone, app, appstore, mac, macintosh
User Negative	fruit, tree, eat, bite, juice, pineapple, strawberry, drink

4 Classification Task

In machine learning literature, the learning tasks could be broadly classified as supervised and unsupervised learning. The problem scenario for the WePS-3 task, classification of tweets with respect to a company entity can be seen as a problem where one needs a machine learning technique between supervised and unsupervised learning, since we have no training set for the actual classification task, but a test training set is provided for a separate set of companies. Here we briefly discuss the different classes of machine learning techniques, and outline our classification method.

[7] http://www.apple.com

Supervised Learning for Classification Task Supervised learning is a machine learning technique for deducing a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can predict a class label of the input object (called classification). The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output). To achieve this, the learner has to generalize from the presented data to unseen situations in a "reasonable" way. An example of supervised learning in our current setting is: given a training set of tweets for a particular company(XYZ company), with example of tweets belonging to and not belonging to the company, one learns a classifier for this particular company(XYZ company). Using this classifier the new unseen tweets related to this company(XYZ company) can be classified as belonging or not belonging to that company.

Unsupervised Learning In machine learning, unsupervised learning is a class of problems in which one seeks to determine how the data are organized. Many methods employed here are based on data mining methods used to preprocess data. It is distinguished from supervised learning in that the learner is given only unlabeled examples. In broad sense, the task of classifying tweets of an unknown company, without seeing any relevant examples can fall into this category.

Generic Learning For the current scenario (WePS-3 - challenge 2), we are provided with training sets corresponding to few companies (C^{TR}). Finally we have to classify test sets corresponding to new companies(C^{Test}), with $C^{TR} \bigcap C^{Test} = 0$. This particular scenario can be seen as in-between supervised and unsupervised learning. It is unsupervised as we are not given any labeled tweets corresponding to the test set. At the same time it is also related to supervised learning as we have access to few training sets, with labeled tweets corresponding to the companies. This kind of generic learning needs the classifier to identify the generic features from the general training set, based on which one can make accurate classification of tweets corresponding to the unseen companies. The classifiers based on the features of the tweet decides if it belongs to a company or not. In the following section 4.1, we discuss the features which our classifiers take as input. After the features are introduced, we propose different ways of developing a generic classifier in section 4.2

4.1 Features Extraction

We define a feature extraction function, which compares a tweet T_i to the company entity representation E_k and outputs a vector of features.

$$Fn(T_i, E_k) = \{ \overbrace{G_1, \ldots, G_m}^{meta-features}, \underbrace{F_1, \ldots, F_n}_{tweet-specific}, \overbrace{U_1, \ldots, U_z}^{heuristics} \}$$

Here the G_i are generic/meta features, which are entirely based on the quality of the entity profiles and do not depend on Tweet message T_i. One could use different ways of quantifying the quality of the profiles.

- Boolean: In this work we make use of boolean metrics to represent if a profile is empty or has sufficient keywords.
- Other possibility is that a human can inspect the profiles and assign a metric of x ∈ [0,1] based on the perceived quality. One could think of exploring an automated way of assigning this number.

The F_i features are tweet specific features, i.e. they quantify how close a tweet overlaps with the entity profiles. We use a comparison function to compare the tweet message T_i, which is a bag of words, with j^{th} profile P_j^k, which is also a bag of weighted keywords, to get the F_j^{th} feature. In this work we make use of a simple comparison function, which compares two bags of words looking for exact overlap of keywords, and for all such keywords the sum of their weights quantify how close the tweet message is to the entity profile. Formally with T_i = Set$\{w_1^t, w_2^t, \ldots, w_k^t\}$ and P_j^k = Set$\{w_1^p : wt_1, w_2^p : wt_2, \ldots, w_m^p : wt_m\}$, we compute the F_j feature using the simple comparison function as:

$$\begin{aligned} F_j = CmpFn(T_i, P_j^k) = \sum_q wt_q, \text{ where q such that} \\ w_q^p \in Set\{w_1^t, w_2^t, \ldots, w_k^t\} \bigcap Set\{w_1^p, w_2^p, \ldots, w_m^p\} \end{aligned} \tag{1}$$

The above comparison function is simple and easy to realize, but it may miss out some semantically equivalent words. One could make use of cosine similarity, or semantic similarity based comparison functions.

The U_i features encapsulate some user based rules, for example, presence of the company URL domain in the tweet URL list, is a big enough evidence to classify the tweet as belonging to the company.

4.2 Generic Classifier

The classifier is a function which takes the feature vector as input and classifies the tweet as $\{TRUE, FALSE\}$, with TRUE label if the tweet is related to the company and as FALSE otherwise. We are provided with training data corresponding to a set of companies (C^{TR}). Based on the training data we have the task of training a generic classifier, which should be used to classify the tweets corresponding to a new set of companies (C^{Test}). We present here two possible ways of designing this generic classifier.

Ensemble of Naive Bayes Classifiers: We adapt the Naive Bayes Classifier model for this task. For each company in the training set(C^{TR}), based on the company tweets we find the conditional distribution of values over features for two classes i.e. a class of tweets which are related to the company and another class of tweets which are not related to the company. With these conditional

probabilities, shown in equations(2,3) and by applying Bayes theorem, we can classify an unseen tweet whether it is related to the company or not.

Let us denote the probability distribution of features of the tweets that are related to a given company with

$$P(f_1, f_2, \ldots, f_n \mid C), \tag{2}$$

and the probability distribution of features of the tweets that are not related to the company with

$$P(f_1, f_2, \ldots, f_n \mid \overline{C}). \tag{3}$$

Then, for an unseen tweet t, using the features extraction function we compute the features values:$(f_1, f_2, \ldots, f_n)$. The posterior probabilities of whether the tweet is related to the company or not, are calculated as in equations (4, 5).

$$P(C \mid t) = \frac{P(C) * P(t \mid C)}{P(t)} = \frac{P(C) * P(f_1, f_2, \ldots, f_n \mid C)}{P(f_1, f_2, \ldots, f_n)} \tag{4}$$

$$P(\overline{C} \mid t) = \frac{P(\overline{C}) * P(t \mid \overline{C})}{P(t)} = \frac{P(\overline{C}) * P(f_1, f_2, \ldots, f_n \mid \overline{C})}{P(f_1, f_2, \ldots, f_n)} \tag{5}$$

Depending on whether $P(C \mid t)$ is greater than $P(\overline{C} \mid t)$ or not, the naive Bayes classifier decides whether the tweet t is related to the given company or not, respectively.

Corresponding to each company $c_i \in C^{TR}$, we train a naive Bayes classifier[12] [15], NBC_i, for which the input features are tweet specific features $F_1, \ldots, F_n$ and heuristics based features $U_1, \ldots, U_z$, as discussed in the section 4.1. Along with training a naive Bayes classifier, we also assign an accuracy measure for this classifier and keep a note of meta features $G_1, \ldots, G_m$ of this classifier.

The generic classifier makes use of ensemble function which either chooses the best classifier or combines the decision of classifiers from this set, to classify an unseen tweet corresponding to a new company i.e. $c_i \in C^{Test}$. The ensemble function would make use of the meta-features and accuracy measures to pick up the right classifier or the right combination of classifiers. We refer to [9] [21] for details about the design of such ensemble functions.

SVM Classifier: Alternatively one could train a single classifier based on all the features: meta-features, tweet-specific features and heuristics-features. This single classifier can be seen as using an ensemble function implicitly in either picking an apt classifier or aptly combing the classifier decisions. In the current work, we train an SVM Classifier [10],[16] as a generic classifier, which makes use of all features: meta-features, tweet-specific features and heuristics-based features, in its classification task.

5 Experiments and Evaluation

Our experimental setup was the following. We are given a general training set, which consists tweets related to about 50 companies (we denote this set as C^{TR}). For each company $c \in C^{TR}$ we are provided around 400 tweets with their corresponding ground truth, i.e. if the tweet is related to the company or not. For each company, we are provided with the following meta-information: URL, Language, Category. We have trained a generic *classifier* based on this training set. The test set for this task consisted tweets of around 50 new companies. We denote this set of companies as C^{Test}. There was no overlap with the training set, $C^{TR} \bigcap C^{Test} = 0$. For each company $c \in C^{Test}$ there are about 400 Tweets, which are to be classified. We classified them with our trained generic *classifier*, as explained in Section 4. The WePS-3 dataset is available at `http://nlp.uned.es/weps/weps-3/data`.

The task is of classifying the tweets into two classes: one class which represents the tweets related to the company (positive class) and second class represents tweets that are not related to the company (negative class). For evaluation of the task, the tweets can be grouped into four categories: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The true positives are the tweets that belong to positive class and in fact belong to the company and the other tweets which are wrongly put in this class are false positives. Similarly for the negative class we have true negatives which are correctly put into this class and the wrong ones of this class are false negatives.

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

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

$$Precsion^{(+)} = \frac{TP}{TP+FP};\ Recall^{+} = \frac{TP}{TP+FN};\ F-Measure^{+} = \frac{2*Precsion^{+}*Recall^{+}}{Precsion^{+}+Recall^{+}}$$

$$Precsion^{-} = \frac{TN}{TN+FN};\ \ Recall^{-} = \frac{TN}{TN+FP};\ F-Measure^{-} = \frac{2*Precsion^{-}*Recall^{-}}{Precsion^{-}+Recall^{-}}$$

In Table 2 we show the average values of the different performance metrics, along with the corresponding variances.

Table 2. Performance of Classifier which makes use of all profiles

Metric	(Mean)Value	Variance
Accuracy	0.83	0.02
Precision (positive class)	0.71	0.07
Recall (positive class)	0.74	0.13
F-Measure (positive class)	0.63	0.1
Precision (negative class)	0.84	0.07
Recall (negative class)	0.52	0.17
F-Measure (negative class)	0.56	0.15

The results show high accuracy figures for our classifier. The precision and recall values corresponding to positive class can be further increased by refining

the profiles corresponding to positive evidence, for example by using more sources to accumulate more relevant keywords and by using efficient quality metrics for rejecting irrelevant keywords. In spite of using very few sources for populating the negative profile of a company, we are still able to have high precision and decent recall values for the negative class. Similarly, by using more sources for negative evidences we can further improve these performance measures.

Next we study the impact of the different profiles, we have used in the entity representation, on the classification task. We study the importance of the negative-keywords-profile and the category-based profile on the performance of the classification process. We considered the following cases:

LSIR.EPFL_1 (ALL) We make use of all the profiles of a company for the classification process.

LSIR.EPFL_2 (No-Neg) We make use of all the profiles except the negative-evidence profile, of a company to classify unseen tweets.

LSIR.EPFL_3 (No-Cat) To study the impact of using the category-related profile in the classification process, we make an experiment which uses all the profiles of a company except the profile corresponding to category and common-sense-keywords profile.

LSIR.EPFL_4 (Only-HP) Company homepage URL is provided as a representation of the entity. We want to study how accurate the classifier performs when a profile is built only based on the keywords, extracted through crawling the homepage.

Table 3. Importance of Different Profiles

Metric	ALL	No-Neg	No-Cat	Only-HP
Accuracy	0.83	0.77	0.79	0.66
Precision (positive)	0.71	0.81	0.69	0.73
Recall (positive)	0.74	0.53	0.71	0.27
F measure (positive)	0.63	0.56	0.64	0.3
Precision (negative)	0.84	0.7	0.86	0.6
Recall (negative)	0.52	0.83	0.52	0.89
F measure (negative)	0.56	0.68	0.56	0.66

From the results shown in table 3, it is clear that the homepage URL does provide us some very relevant information for the classification task, however the accuracy is low. The accuracy can be improved if one uses also other sources of information, like negative evidence, category and common sense based keywords.

6 Related work

The classification of tweets has already been addressed in the literature, in different contexts. Some of the relevant works include [5][18][17][13].

In [5], the authors take up the task of classifying the tweets from twitter into predefined set of generic categories such as News, Events, Opinions, Deals and Private Messages. They propose to use a small set of domain-specific features extracted from the tweets and the user's profile. The features of each category are learned from the training set. This task which can be seen as a supervised learning scenario is different from our current task which is a generic learning task.

The authors in [18], build a news processing system based on Twitter. From the twitter stream they build a system that identifies the messages corresponding to late breaking news. Some of the issues they deal with are separating the noise from valid tweets, forming tweet clusters of interest, and identifying the relevant locations associated with the tweets. All these tasks are done in an online manner. They build a naive Bayes classifier for distinguishing relevant news tweets from irrelevant ones. They construct the classifier from a training set. They represent intermediate clusters as a feature vector, and they associate an incoming tweet with cluster if the distance metric to a cluster is less than a given threshold.

In [13] and [17], the authors make use of twitter for the task of sentiment analysis. They build a sentiment classifier, based on a tweet corpus. Their classifier is able to classify tweets as positive, negative, or neutral sentiments. The papers identify relevant features (presence of emoticons, n-grams), and train the classifier on an annotated training set. Their work is complementary to ours: the techniques proposed in our work could serve as an essential preprocessing step to these sentiment or opinion analysis, which identifies the relevant tweets for the sentiment analysis.

The paper [19] proposes a technique to retrieve photos of named entities with high precision, high recall and diversity. The innovation used is query expansion, and aggregate rankings of the query results. Query expansion is done by using the meta information available in the entity description. The query expansion technique is very relevant for our work, it could be used for better entity profile creation.

Many works based on entity identification and extraction, for example in [4], [8], [14], [21], usually make use of the rich context around the entity reference for deciding if the reference relates to the entity. However, in the current work, the tweets which contain the entity references usually have very little context, because of the size-restrictions of tweet messages. Our work addresses these issues, namely how to identify an entity in scenarios where there is very little context information.

Bishop [6] discusses various machine learning algorithms for supervised and unsupervised tasks. The task we are addressing in this paper is generic learning, which can be seen as in between supervised and unsupervised learning. Yang et al. [20] discuss generic learning algorithms for solving the problem of verification of unspecified person. The system learns generic distribution of faces, and intra-personal variations from the available training set, in order to infer the distribution of the unknown new subject, which is very related to the current task. We adapt techniques from [6] and [9] for the tweets classification task.

There are many ways to represent entities. In Okkam[7] project, which aimed to enable the Web of entities, an entity is represented as a set of attribute-value pairs, along with the meta information related to the evolution of entity, and relationships with other entities. In dbpedia[1] and linked data[2] the entities are usually represented using RDF models. These rich models are needed for allowing sophisticated querying and inferences. Since we use the entity representation for our classification algorithms, we resort to representing an entity simply as a bag of weighted keywords instead of the rich representations of entities.

7 Conclusion and future work

Twitter is a real time pulse of the opinions of the people. Researchers have analyzed the twitter streams for different purposes: finding influential tweeters, opinion mining, categorizing tweets, summarizing tweets, etc. In some of these tasks, like opinion and sentiment mining, the classification of the twitters based on entities forms an important preprocessing step, as the accuracy of further analysis depends on this step. In this paper we address the task of classifying tweets based on entities, for which we use a simple entity representation. We realized an efficient classification process with the help of entity profiles, which we constructed using different information sources.

One can observe that the accuracy of our classification technique depends on the quality of the entity profiles. As future work, we would like to explore other techniques to further improve the quality of the entity profiles, including ensemble techniques[9], [21]. We would also like to explore dynamic ways of adapting the entity profiles, where the information from the twitter stream can be used to add or remove keywords from the entity profiles. Further we think that there is need for efficient quality metrics, similar to the ones used in information retrieval literature [3], in order to decide if a particular keyword is relevant or not, to the representation of entity.

References

1. Dbpedia. `http://dbpedia.org/`.
2. Linked data. `http://linkeddata.org/`.
3. Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
4. Ron Bekkerman and Andrew McCallum. Disambiguating web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web*, pages 463–470, 2005.
5. Enngin Demir Hakan Ferhatosmanoglu Bharath Sriram, David Fuhry and Murat Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the ACM SIGIR 2010 Posters and Demos*. ACM, 2010.
6. Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
7. Paolo Bouquet, Heiko Stoermer, and Daniel Giacomuzzi. Okkam: Towards a solution to the "identity crisis" on the semantic web. In *In Proceedings of SWAP 2006, the 3rd Italian Semantic Web Workshop*, pages 18–20, 2006.

8. Zhaoqi Chen, Dmitri V. Kalashnikov, and Sharad Mehrotra. Exploiting context analysis for combining multiple entity resolution systems. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 207–218, 2009.
9. Sungha Choi, Byungwoo Lee, and Jihoon Yang. Ensembles of region based classifiers. In *CIT '07: Proceedings of the 7th IEEE International Conference on Computer and Information Technology*, pages 41–46, Washington, DC, USA, 2007. IEEE Computer Society.
10. Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.
11. Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. Outtweeting the Twitterers - Predicting Information Cascades in Microblogs. In *3rd Workshop on Online Social Networks (WOSN'10)*, 2010.
12. David Heckerman. A tutorial on learning with bayesian networks. Technical report, Learning in Graphical Models, 1996.
13. B.J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 2009.
14. Dmitri V. Kalashnikov, Zhaoqi Chen, Sharad Mehrotra, and Rabia Nuray-Turan. Web People Search via Connection Analysis. *IEEE Transactions on Knowledge and Data Engineering* , 20(11):1550–1565, November 2008.
15. David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. pages 4–15. Springer Verlag, 1998.
16. Donald Metzler, Susan Dumais, and Christopher Meek. Similarity Measures for Short Segments of Text. In *Advances in Information Retrieval*, volume 4425 of *LNCS*, pages 16–27, 2007.
17. Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).
18. Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *GIS '09: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 42–51, New York, NY, USA, 2009. ACM.
19. Bilyana Taneva, Mouna Kacimi, and Gerhard Weikum. Gathering and ranking photos of named entities with high precision, high recall, and diversity. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *WSDM*, pages 431–440. ACM, 2010.
20. Qiong Yang, Xiaoqing Ding, and Xiaoou Tang. Incorporating generic learning to design discriminative classifier adaptable for unknown subject in face verification. *Computer Vision and Pattern Recognition Workshop*, 0:32, 2006.
21. Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. Towards better entity resolution techniques for Web document collections. In *1st International Workshop on Data Engineering meets the Semantic Web (DESWeb'2010) (co-located with ICDE'2010)*, 2010.