International Journal of Computer Science and Applications
© Technomathematics Research Foundation
Vol. 9 No. 2, pp. 88 - 115, 2012

Entity-based Classification of Twitter Messages

Surender Reddy Yerva*
EPFL IC LSIR
Lausanne, Switzerland,
surenderreddy.yerva@epfl.ch

Zoltán Miklós
EPFL IC LSIR
Lausanne, Switzerland,
zoltan.miklos@epfl.ch

Karl Aberer
EPFL IC LSIR
Lausanne, Switzerland,
karl.aberer@epfl.ch

Twitter is a popular micro-blogging service on the Web, where people can enter short messages, which then become visible to some other users of the service. While the topics of these messages varies, there are a lot of messages where the users express their opinions about some companies or their products. These messages are a rich source of information for companies for sentiment analysis or opinion mining. There is however a great obstacle for analyzing the messages directly: as the company names are often ambiguous (e.g. apple, the fruit vs. Apple Inc.), one needs first to identify, which messages are related to the company. In this paper we address this question. We present various techniques for classifying tweet messages containing a given keyword, whether they are related to a particular company with that name or not. We first present simple techniques, which make use of company profiles, which we created semi-automatically from external Web sources. Our advanced techniques take ambiguity estimations into account and also automatically extend the company profiles from the twitter stream itself. We demonstrate the effectiveness of our methods through an extensive set of experiments. Moreover, we extensively analyze the sources of errors in the classification. The analysis not only brings further improvement, but also enables to use the human input more efficiently.

Keywords: Twitter, Classification, Entity profile, Entity disambiguation

1. Introduction

Twitter[a] is a popular micro-blogging service on the Web, where people can enter short messages (a.k.a. tweets), which then become visible to other users. Twitter is currently one of the most popular sites of the Web: as of February 2010, Twitter users send 50 million

[*] EPFL IC LSIR, Bat. BC 118, Station 14, 1015 Lausanne, Switzerland
[a] `http://www.twitter.com`

messages per day [b]. While the subject of these varies, in many cases the messages express opinions about companies or their products. Since the service is very popular, the twitter messages form a rich source of information for companies about how their customers like their products. In the same way companies might learn what is the general perception of the company. There is however a great obstacle for analyzing the data directly: as the company names are often ambiguous, one needs first to identify, which messages are related to the company. This name ambiguity is not accidental, the choice of the company name is part of the branding and marketing strategy. Examples for such company and brand names from the technology industry are Apple ™ Inc., Orange® or BlackBerry®.

In this paper we focus on the problem of classifying twitter messages containing a given keyword, whether or not they are related to a given company. 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 we collected keywords from different sources (Web, User) automatically and in some cases manually. The company profiles essentially contain these keywords, which are related to the company in some way. With each profile we also maintain a set that contains unrelated keywords. With the help of these profiles we could construct a classifier.

Table 1. Tweets containing the keyword "apple"

T1	".. **installed** yesterdays **update released** by *apple*.."	T
T2	".. the *apple* **juice** was bitter.."	F
T3	".. it was easy when *apples* and **blackberries** were only **fruits**.."	T
T4	".. dropped my *apple*, mind u its not the **fruit**.."	T

Table 1 gives some examples of tweets containing the keyword "apple". Our task is to decide whether these messages are related to the company Apple Inc. or not. This task is not trivial, even for human inspectors. The human decision process relies on some specific keywords, which –together with the background knowledge– give hints for the decision. In the table, the bold words are examples for such possible hints. In our classification techniques, we try to construct profiles, which contain exactly these keywords. Note that in the sentences T3 and T4 the speaker exploits the multiple possible interpretations of the word "apple". (If one of them is the company Apple Inc. we try to classify the message as TRUE.)

Beyond this standard technique we construct more sophisticated classifiers as well. First we estimate the overall ambiguity of a company name, and include this information

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

in our classification decision. Moreover we do not use static profiles for the companies, rather dynamic ones, which we continually update from the twitter stream. This extension is essential and specific to our classification problem. The keywords appearing in the tweets are repeated with changing frequencies: for example if a company launches a new product, this new product name might appear more frequently in the twitter stream, and such keywords can be temporarily good indications that the message is related to the company. We conducted an extensive set of experiments using the WePS-3 dataset [c] and also through direct access to the twitter stream. The experiments show promising performance figures. Moreover, we extensively analyze the sources of errors in the classification. The analysis not only brings further improvement, but also enables to use the human input more efficiently.

The rest of the paper is organized as follows. Section 2 explains the problem more formally. Section 3 presents our basic classification technique, while Section 4 describes our more advanced techniques, where we involve ambiguity estimations and also active profiles. Section 5 contains the results of our extensive experimental evaluation. Section 6 elaborates on the reasons of errors in the classification and presents systematic techniques to minimize the effect of certain types of errors. Section 7 summarizes the related work and finally, Section 8 concludes the paper.

2. Model and Problem Statement

2.1. *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. We also use the term that a tweet belongs to a company, by which we mean the same. 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. We assume that some basic further information is available as input, such as the URL of the company $url(C)$, the language of the Web page.

2.2. *Model*

2.2.1. *Tweet Representation*

We represent 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,

[c] `http://nlp.uned.es/weps/weps-3` In fact, we are not using the training set of WePS-3, just the test set with the available ground truth, for evaluation purpose.

for example, RT,@username). We store a stemmed[d] version of keywords (unigrams and bigrams). Formally we have:

$$T_i = set\{wrd_j\}. \tag{1}$$

2.2.2. *Company Representation*

We represent each company entity as a profile, where a profile is a set of weighted keywords.

$$P_c = \{wrd_j : wt_j\} \tag{2}$$

with $wt_j \geq 0$ for positive evidence keywords (i.e. those words which suggest that the message should be related to the company) and $wt_j < 0$ for negative evidence keywords. We can consider the profile as two sets of weighted keywords. The set with positive weights constitute positive evidence keywords and the set with negative weights represent negative evidence keywords.

$$P_c.Set^+ = \{wrd_j : wt_j \mid wt_j \geq 0\} \tag{3}$$

$$P_c.Set^- = \{wrd_j : wt_j \mid wt_j < 0\} \tag{4}$$

The weights wt_j corresponding to word wrd_j essentially captures the conditional probability of the event that a message containing the keyword belongs (or does not belong) to the given company C. (For simplicity, we denote these events as C and $\overline{C}$).

$$P(wrd_j \mid C) = wt_j \text{ if } wt_j \geq 0, \tag{5}$$

$$P(wrd_j \mid \overline{C}) = |wt_j| \text{ if } wt_j < 0, \tag{6}$$

2.2.3. *Classification Process*

For the tweets classification task, we compare the tweet with the entity (i.e. company) profile. We make use of Naive Bayes Classifier [Heckerman (1999)], [Lewis (1998)] for our classification process. We assume the words appearing in a tweet independently contribute towards the evidence of whether the tweet belongs to the company, or not.

For each tweet T_i = $set\{wrd^i_j\}$ we compute the conditional probabilities $P(C \mid T_i)$ and $P(\overline{C} \mid T_i)$ for deciding if a tweet belongs to a company C or not. We make use of Bayes theorem for computing these terms.

$$\begin{aligned} P(C \mid T_i) &= \frac{P(C) * P(T_i \mid C)}{P(T_i)} = \frac{P(C) * P(wrd^i_1, \ldots, wrd^i_n \mid C)}{P(T_i)} \\ &= K_1 \prod_{j=1}^{n} P(wrd^i_j \mid C) \end{aligned} \tag{7}$$

[d] We used the Porter stemmer from the python based natural language toolkit, available at `http://www.nltk.org`

Similarly we have,

$$P(\overline{C} \mid T_i) = K_2 \prod_{j=1}^{n} P(wrd_j^i \mid \overline{C}) \tag{8}$$

where, $P(wrd_j \mid C)$ and $P(wrd_j \mid \overline{C})$ are the weights associated with the words wrd_j as described in previous section. Depending on whether $P(C \mid T_i)$ is greater than $P(\overline{C} \mid T_i)$ or not, the Naive Bayes Classifier decides whether the tweet T_i is related to the given company or not, respectively.

3. Basic Twitter classification

In this section we present a basic classification technique for twitter messages. This technique is an improved version of our classifier [Yerva *et al.* (2010a)], which we developed in the context of WePS-3 evaluation challenge. It is referenced with the name LSIR-EPFL in [Amigó *et al.* (2010)]. Our classifier is essentially a Naive Bayes classifier, which relies on constructed company profiles. In the following we give details about how we constructed the profiles from different information sources. We represent a company using basic profile, which is set of weighted keywords. We assume that for each company we are provided with the company name, an URL representing the company, the category to which the company belongs. For each information source we show how we extract the keywords, and discuss the advantages and disadvantages associated with that source.

Homepage Keywords For each company name, we assume that the company homepage URL is available. To extract relevant keywords from the homepage URL, we crawled all the relevant links up to a depth of level d(=2), starting from the given homepage URL. First we extracted all the keywords present on these relevant pages, then we removed all the stop-words, finally we store in the profile the stemmed version of these keywords. From this construction process one would expect that homepage provides us all the important keywords related to the company. However, since the construction is an automated process, it was not always possible to capture good quality representation of the company for various reasons like: the company webpages may use java-scripts, some use flash, some company pages contain irrelevant links, most of the webpages are non-standard home-pages etc. The collected keywords from this source contribute towards positive evidence.

Metadata Keywords HTML standards provides few meta tags[e], which enables a Web page to list set of keywords that one could associate with the Web page. We collect all such meta keywords 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. The metadata keywords contribute towards positive evidence.

[e] $http : //www.w3schools.com/html/html\_meta.asp$

Category Keywords 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. For example Apple Inc. belongs to “Computers Software and Hardware” category. One usually fails to find this kind of category related keywords on the homepage URLs. Further, we make use of WordNet[f], a network of words, to find all the terms linked to the category keywords. Thus by using this kind of source helps us associate keywords like: software,install, update, virus, version, hardware, program, bugs etc to a software company entity. This source of keywords contribute towards positive evidence.

GoogleSet/CommonKnowledge Keywords GoogleSet is a good source of obtaining “common knowledge” about the company. We make use of GoogleSets[g] 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 and their products with the competitors. We could for example associate Mozilla, Firefox, Internet Explorer, Safari keywords to Opera Browser entity from the keywords inferred from this source.

UserFeedback Positive Keywords The user himself enters the keywords which he feels are relevant to the company. The keywords we get from the user are of high quality, though they would be few in number. 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 Keywords 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 set of keywords 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[h] contains this information, at least for some entities. Finally this information could be gathered in a dynamic way i.e., using the keywords in all the tweets, that do not belong to the company. In fact, our more sophisticated classifier to be discussed in section 4 exploits this information. The unrelated keywords could also be obtained if we have training set for a particular company with tweets that do not belong to the company entity. Only keywords from this source contribute towards the negative evidence during the classification of tweet.

[f] `http://wordnet.princeton.edu/`

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

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

Table 2 shows the basic profile of “Apple Inc”[i] company entity.

Table 2. Apple Inc. Basic Profile

Positive Evidence Keywords
HomePage Source: iphone, ipod, mac, safari, ios, iphoto, iwork, leopard, forum, items, employees,itunes, credit, portable, secure, unix, auditing, forums, marketers, browse, genius, music, recommend, preview, type, tell, notif, phone, purchase, manuals, updates, fifa, 8GB, 16GB, 32GB …
Metadata Source: {empty}
Category Source: opera, code, brainchild, movie, trade, paper, freight, keyboard, merchandise, disk, language, microprocessor, move, web, monitor, show, instrument, board, lade, digit, shipment, food, cpu, moving-picture, fluid, consign, contraband, electronic, volume, peripherals, crt, resolve, yield, server, micro, magazine, 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, machine, deliver, job, output, release
GoogleSets Source: itunes, intel, belkin, 512mb, sony, hp, canon, powerpc, mac, apple, iphone, ati, microsoft, ibm
UserFeedback Source (Positive): iphone, ipod, itouch, itv, iad, itunes, keynote, safari, leopard, tiger, iwork, android, droid, phone, app, appstore, mac, macintosh
Negative Evidence Keywords
UserFeedback Source (Negative): fruit, tree, eat, bite, juice, pineapple, strawberry, drink

We associated a weight proportional to the quality of the source from which these words are extracted. More generally, if a training set is available one can use more sophisticated techniques. From the training set of the company, for each word, let N_r be the number of tweets containing this word and belong to the company. Similarly N_{nr} be the number of tweets in the training set containing this keyword but do not belong to the company. The weight of the keyword can be chosen proportional to $\frac{N_r}{N_r+N_{nr}}$. In this process, there could be many keywords in the profile, where there are no tweets in the training set containing these words. For all such words one can associate a weight proportional to the quality of the source from which these words are extracted, as in our simple case. This default weight for the keywords not present in the training set tweets, is similar to default weights usually used for an improved Naive Bayes Classifiers [Kim *et al.* (2002)].

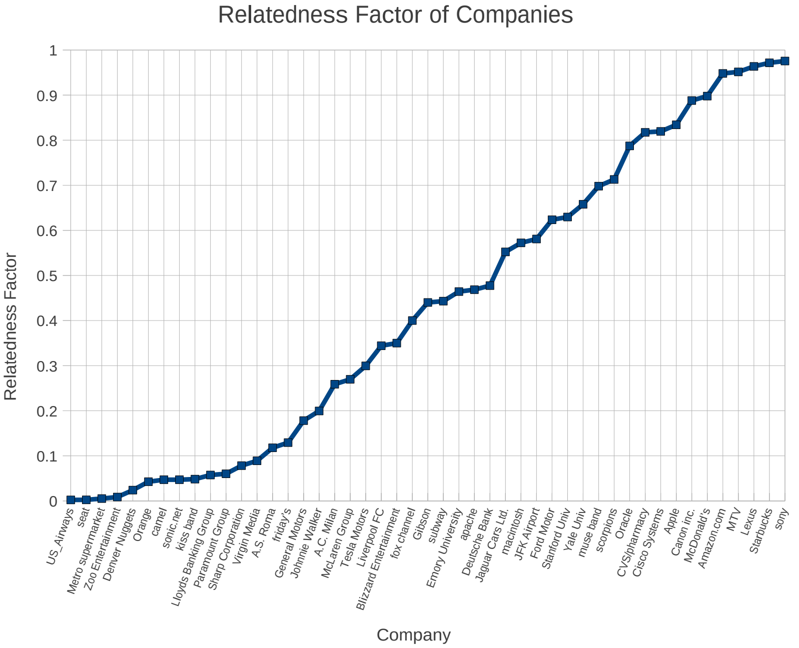

Fig. 1. Relatedness Factor of Companies

4. Improved techniques

4.1. *Relatedness-based Classification*

Based on the training set of size 50 tweets per company, we estimate the $relatedness$ factor of a company. We define this term as the percentage of tweets that really belong to the company.

$$relatedness = \frac{\text{\# of tweets in Training Set} \in \text{Company}}{\text{\# of tweets in the Training Set}} \tag{9}$$

Figure 1 shows the estimated $relatedness$ factor of the different companies in the test set. Companies with higher $relatedness$ factor (for example: Sony, Starbucks, MTV etc.), implies majority of the tweets containing the company keyword belong to the company. Similarly for companies with very low $relatedness$ factor (for example: Seat, Orange, Camel etc.), implies the majority of the tweets mentioning the company keyword do not refer to the company. Note that the $relatedness$ factor characterizes a company based on the dataset and it is independent of the entity profiles.

When classifying a tweet, we actually compare the words present in the tweet against the words present in the profile of a company. Since the number of words we have in the

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

profile are often limited and the possible set of words present in tweet is potentially infinite, in many cases, for many tweets, we do not find any overlap with the company profile. In such cases, it would be better to classify such tweets according to the $relatedness$ factor of the company. The knowledge of the $relatedness$ factor helped us to improve the accuracy of our classification. This technique particularly improves the performance in the cases, where the constructed company profiles are small or have low quality.

Once we know (i.e. estimate) the $relatedness$ factor of a company, there are two ways of classifying an unseen tweet. The first strategy is, if this factor is greater than 0.5, for all tweets we classify them as belonging to the company. This way of classifying helps us achieve an expected accuracy equal to the $relatedness$ factor. When the $relatedness$ factor of a company is less than 0.5, all the tweets are classified as not belonging to the company. In this case, we achieve an expected accuracy of 1 - ($relatedness$).

The second way is, for each tweet we classify the tweet belonging to the company with a probability equal to the $relatedness$ factor. In this way of classification, we would have tweets in both the classes: belonging to the company and not belonging to the company. The expected accuracy of this process can be shown to be a little lower than first case, but we gain some knowledge in this probabilistic classification which could be used for classifying future unseen tweets. We explain in more detail how we can infer some useful information using this method in the following section (Section 4.2).

Let us denote by N the number of tweets to be classified. With $p = relatedness$ factor, we have $p \times N$ tweets belonging to the company and $(1-p) \times N$ tweets not belonging to the company. When we decide with probability p that a tweet belongs to the company, we would be right with $p^2 \times N$ tweets as belonging to the company and $(1-p)^2 \times N$ tweets as not belonging to the company. So, in total the expected accuracy is given as:

$$\text{Expected Accuracy} = p^2 + (1-p)^2, \text{where } p \ = \ relatedness\text{-factor.} \tag{10}$$

We assume that the $relatedness$ factor of a given company does not change in time. We can make this assumption as these changes are relatively slow. One can observe dynamic changes of individual word frequencies which we handle using a different technique, that we explain in the next section.

4.2. *Active Stream Learning Based Classification*

In Section 3 we described how we constructed a basic profile of the company using few reliable sources (such as company homepage, category keywords, Google sets keywords, user feedback etc.) which give us list of keywords which help us decide if a tweet belongs the company. The basic profile is a good starting point for building an efficient classifier, however there are severe limitations of just using the basic profile, which we need to address in order to design better classifiers. In this section, we identify these limitations and propose novel techniques to overcome them.

The efficiency of the basic profile is limited by number of tweets in the test set that contain at-least few overlapping words from the basic profile. From the analysis of the test set tweets we observe that there is a significant percentage of tweets, which do not have any overlapping words with the corresponding basic profile keywords. The Figure 3 in Experiments section confirms this observation.

Some of the limitations of using only the basic profile include:

- The number of keywords in the basic profile are limited, while the number of words one could find in a twitter stream of the company are potentially infinite.
- The sources from which we gather the basic profile keywords are good for collecting positive evidence keywords but not so good for negative evidence keywords. It is possible, at least through human input and with the help of many Web sources, to associate all possible keywords related to a company. On the other hand it is relatively difficult to get a list of entities with which a company keyword could be confused. There is no single authoritative source on the web which lists all possible interpretations of a company name.
- The basic profile does not consider the characteristics of the words distribution in a tweet stream. The power law shown by word frequencies of tweet words, suggests which words should be present in the company profile so as to make an intelligent decision.
- The $relatedness$ factor of a company is useful information, which is completely ignored by a classifier that solely relies on the basic profiles.
- The limited user feedback is completely ignored by the basic profile. Usually it is difficult to involve humans in classifying the tweets, as there are numerous tweets in amount. Even for some number of tweets for which the user is willing to provide feedback, is not exploited by the basic profile.

Few observations made on the twitter streams, along with identifying $relatedness$ factor of the company helps us in overcoming many limitations of the basic profile based classifier. Here we discuss our observations and how we make use of them in developing more accurate classifier.

For each company we inspected the messages from the twitter stream which contain the given company name as a search keyword. For each company, by inspecting the twitter stream [j] (of about 2000 tweets), we studied the word frequency distributions. In general, we could observe power law of distributions for word frequencies. If we have a knowledge about all or top-k of these words, and if these words contribute as positive or negative evidence, then this should help us in classifying many more tweets from test set more accurately. Indeed, we applied such techniques.

The premise we use for improving over basic profile classifier is, to add more words to the positive and negative evidence profile. While adding these words we have to make sure they are of high quality and if they have more possibility of appearing in the future tweets. Some of the tweets which we are able to identify accurately using the basic profile, provide us more keywords, which can be used to resolve new unseen tweets. For example, assume

[j] `http://search.twitter.com/search.json?q=COMPANY-NAME`

our basic profile about Apple Inc. company contained only keywords $\{$iPhone, iPod, mac$\}$. Now when inspecting tweets from stream containing the "apple" keyword, we observe that there are many tweets mentioning "iPhone" and "iPad" together. Since we are able to classify all such tweets as belonging to the Apple Inc. company by the virtue of "iPhone" keyword, we can confidently associate "iPad" word also as a useful word which helps us associate future tweets containing only "iPad" keyword as belonging to Apple Inc.

As discussed in Section 3, in our representation the basic profile contains two sets of weighted keywords. The set with positive weights contribute as positive evidence while the negative weights set contribute as negative evidence. The weights of the words signify how confident the word helps in classifying the tweet as belonging to or not belonging to the company.

We proceed as follows (Algorithm 1). We start inspecting the twitter stream using this basic profile. Of the many tweets we inspect some percentage of tweets, which have overlap with the basic profile, can be accurately classified. All words co-occurring with profile keywords in these tweets can be added to the profile. The weights we associate with these newly identified keywords should depend on the words which made them as possible candidates and also on number of times they co-occurred.

Also when inspecting twitter stream, we would come across many tweets which do not have any overlap with the basic profile keywords. For all such tweets, we classify based on the $relatedness$ factor of the company. We end up with two sets of tweets: one set of tweets which we classify as belonging to the company and the other set as not belonging to the company. For both the sets, based on the word frequency distribution, we add all the keywords above certain threshold to the profile. The weight we associate with these words should depend on number of times the word appears and the $relatedness$ factor.

When there is feedback on some of the tweets by the user, this model is able to use the feedback very efficiently. All the tweets on which the user has responded, the active stream learning algorithm can ignore the basic profile-based and $relatedness$ factor-based decisions and give more weight-age to the user responded tweet keywords.

5. Experimental evaluation

Experimental setup

We performed our experiments on a 2GB RAM, Genuine Intel(R) T2500 @ 2.00 GHz CPU. Linux Kernel 2.6.24, 32-bit machine. We implemented our methods using matlab, java and python.

Dataset

We used the WePS-3 Dataset available at `http://nlp.uned.es/weps/weps-3/data` as our test set. This dataset contained about 47 companies, with each company having about 450 tweets. All the tweets corresponding to a company are annotated as belonging to or not belonging to the company. For each company we randomly selected 50 tweets out of about 450 tweets as our training set. We used the training set only for es-

Algorithm 1 Active Stream Learning

```
Input : Basic Profile: $P_0.Set^+, P_0.Set^-$
  Twitter Stream: $\Gamma = \{T_1, \ldots, T_n\}$
  R : $Relatedness$ factor of company
Init : Active Tweet Sets: $P_\triangle.Set^+ = \{\}, P_\triangle.Set^- = \{\}$
for all $T_i \in \Gamma$ do
  $score$ = SCORE($T_i, P_0.Set^+$) + SCORE($T_i, P_0.Set^-$)
  if $score > 0$ then
    $P_\triangle.Set^+$.add($T_i$,score)
  else if $score < 0$ then
    $P_\triangle.Set^-$.add($T_i$,score)
  else
    if $Math.radom(0,1) < Relatedness$ factor then
      $P_\triangle.Set^+$.add($T_i$,$Relatedness$)
    else
      $P_\triangle.Set^-$.add($T_i$,$Relatedness$)
    end if
  end if
end for
{ $P_\triangle.Set^+$,$P_\triangle.Set^-$ } = WordFreqAnalysis($P_\triangle.Set^+, P_\triangle.Set^-$)
Add Top-K keywords or all words above Threshold from $P_\triangle.Set^+$ to $P_0.Set^+$
Add Top-K keywords or all words above Threshold from $P_\triangle.Set^-$ to $P_0.Set^-$
return $P_0.Set^+, P_0.Set^-$
```

timating the $relatedness$ factor for each company. For constructing the active profiles, we gathered twitter streams for each company, using the query term shown in Table 5, from `http://search.twitter.com`. The number of tweets we investigated for active profiles varied from 600 to 9900 tweets.

Metrics

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 $accuracy$ metric to study the performance of our different classifiers.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

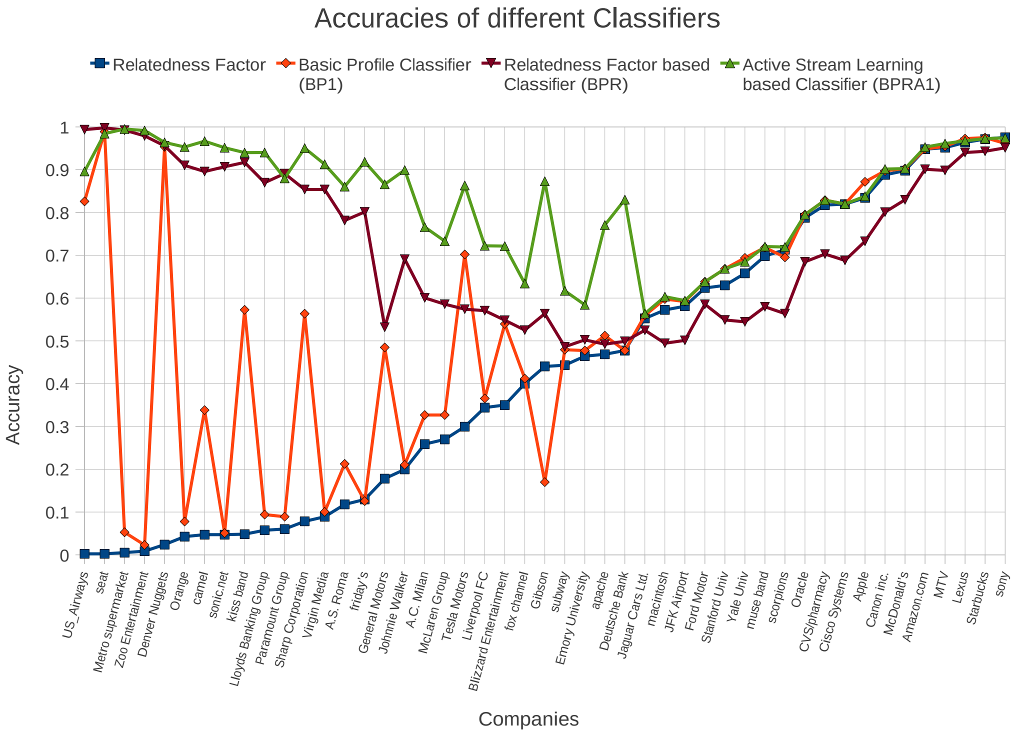

Fig. 2. Accuracies of different Classifiers

Different Classifiers

Our experiments make use of following different classifiers:

(1) Basic Profile-based Classifier(BP1): For each company we formed the basic profile, which included keywords from all the sources: homepage, category, metadata, google sets and user feedback.

(2) Basic Profile-based Classifier(BP2): In general we observed that keywords extracted from homepage source are of low quality compared to all other sources. So, we formed a second basic profile whose keywords are from high quality sources like category, metadata, google sets and user feedback.

(3) $Relatedness$ factor based Classifier (BPR): Based on the training set we estimated the $relatedness$ factor of each company. Using this factor the classifier classified all the tweets.

(4) Active Profile Classifier (BPRA1): We used high quality basic profile (BP2), which considered only high quality sources, for forming the active profile. This classifier based on the active profile classified all the tweets in the test set.

(5) Active Profile Classifier (BPRA2): In order to study the impact of the quality of basic

profile on the construction of active profile, we used basic profile (BP1) for forming the active profile. This classifier based on the active profile(BPRA2) is used to classify all the tweets in the test set.

(6) Active Profile Classifier (BPRA3): We earlier discussed that the quality of the active profile depends on how good the starting basic profile we use for its construction. For the active profile classifier BPRA3 we assume that the initial basic profile is empty, and go about constructing the active profile based only on the $relatedness$ factor decisions.

Please note that the classifiers (BPRA1), (BPRA2) and (BPRA3) internally make use of the estimated $relatedness$ parameter, as it is explained in Algorithm 1.

In the first set of experiments, we study how the different classifiers performed on the test set. The accuracy metric of the different classifiers : BP1, BPR and BPRA1 are shown in the Figure 2. We see that on average the $relatedness$ factor based classifier (BPR) and active profile based classifier (BPRA1) outperform the basic profile-based classifier (BP1). Also the BPRA1 classifier outperformed BPR classifier. On close observation of the Figure 2, we see that for the companies on the far-right that is with high $relatedness$ factor, the profile-based classifiers BP1 and BPRA1 are better than the classifier BPR. The reason is, the basic profile is already good enough to capture all the useful words associated with the company. The active profile does not improve much on the basic profile. Thus they both outperform the classifier (BPR). This is in tune with the argument in Section 4 that it is relatively easy to gather positive evidence keywords compared to the negative evidence keywords.

In the left side of the graph where the relatedness factor of the companies are low, we observe that BPR and BPRA1 clearly outperform BP1. It strongly suggests that the basic profile was not good enough to contain all the negative evidence keywords associated with the company. BPR is outperforming because it is exploiting the $relatedness$ factor estimate. While BPRA1 was able to efficiently identify all the supporting keywords which were not initially available in the basic profile.

The significant performance improvement of active profile-based classifier over the basic profile based classifier can be attributed to the fact that the active profile is able to identify many more keywords just by inspecting the twitter streams. In Figure 3 we show number of words in the profiles that overlap with the top 50 keywords of the test set. It confirms our observation that only small percentage of tweets in the test set overlap with the keywords in the basic profile. We also see that by use of active profile, there is significant percentage of overlap between the keywords in the test set and the active profile.

The quality of the active profile we construct depends on the quality of the basic profile that is used. In order to study how the different basic profiles affect the active profile based classifiers performance, we constructed many active profiles BPRA1,BPRA2 and BPRA3, each starting with a different quality basic profile. From the description of the different basic profiles, we see that the quality of BP2 classifier is better than BP1 classifier, which further are better than the empty basic profile. The average performance of each of the different classifiers is shown in the Table 3. From the table we observe that BPRA1 is better than BPRA2 which in turn is better than BPRA3 classifier. Thus we observe that as

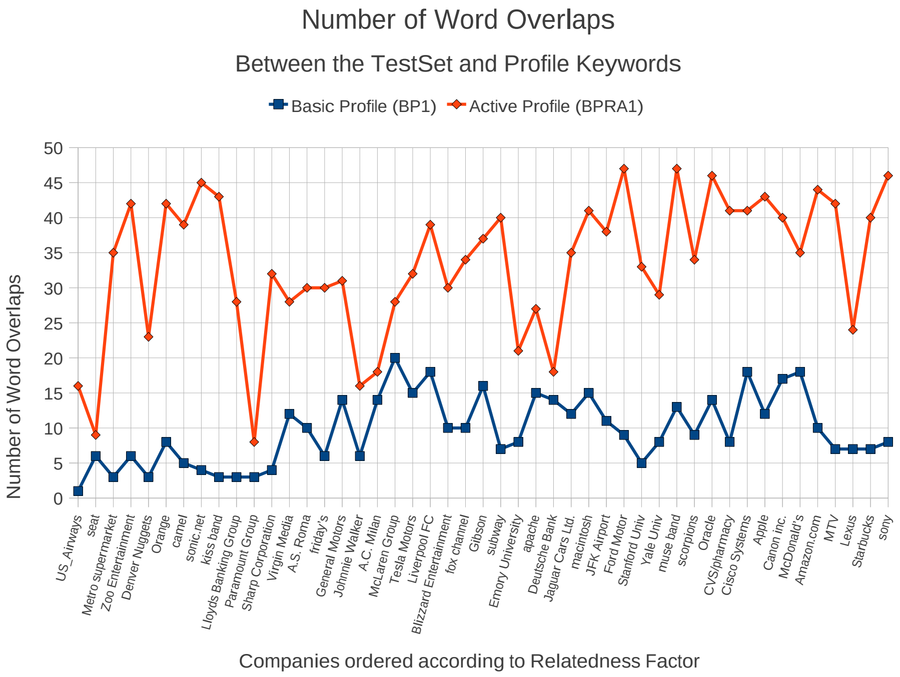

Fig. 3. Accuracies of different Classifiers

the basic profile quality deteriorates so does the performance of the corresponding active profile.

Table 3. Average Accuracy of Different Classifiers

Classifier	**Average Accuracy**
Basic Profile using all sources (BP1)	0.43
Basic Profile using only high quality sources (BP2)	0.46
$Relatedness$ factor based classifier (BPR)	0.73
Active Profile constructed using high quality Basic Profile-BP2 (BPRA1)	0.84
Active Profile constructed using normal quality Basic Profile-BP1 (BPRA2)	0.79
Active Profile constructed using the empty Basic Profile (BPRA3)	0.76

6. Performance Analysis and Further Improvements

We have introduced and evaluated various Twitter classification methods. In Section 3 we started with a simple classifier only relying on a basic profile, while in Section 4 we improved this method through the use of the $relatedness$ factor and updates from the active Twitter stream. In Section 5 we evaluated these methods. Our evaluation shows that the performance of these classifiers is still leaves some room for improvement, for some companies. In this section we look into the reasons for the under-performance and also propose principled techniques for improvements.

As a summary, our classifiers work as follows. A company profile in our setting is a set of weighted keywords. When a company profile is used for classifying an unseen tweet, the Naive-Bayes classification looks for overlapping keywords in the tweet message and in the company profile. The net sum of the weights of the overlapping words, determines if the tweet belongs to the company or not. For all the tweets which do not have any overlapping words with the profile, we classify those tweets based on the $relatedness$ factor of the company.

We first introduce some useful concepts for studying the performance of a classifier. The performance of a classifier, given a company profile on the test set collection of tweets, depends on how well the keywords of the test set collection overlap with the company profile keywords and how accurate are the weights in the company profile. Thus, to improve the performance of classifiers, we need "better" profiles, that is profiles that contain a high number of relevant keywords which also appear in the test set collection, with as accurate weights as possible.

For our performance analysis, we define following concepts:

Perfect Profile : P_c : We define the Perfect Profile, P_c, of a company as the profile that can be formed using the words inferred from the entire test set. The weights associated with these words reflect the distribution of words in the entire test set collection.

Eventually, when one uses this profile for classifying the tweets in the test set, with the given classification method we will have the best possible performance. The performance of the classifier that uses the Perfect Profile is an upper bound for the accuracy level of the classifier with any other profile.

Current Profile : P_i : It is the profile that is formed using the different techniques proposed in the earlier sections (Sections 3 and 4) that is eventually used by the classifier.

Next we look into the performance differences of the Current Profile P_i w.r.t. the Perfect Profile P_c

6.1. *Comparison of the Current Profile and the Perfect Profile of a Company*

We summarize the performance of Current Profile in relation to the performance of Perfect Profile in the Figure 4. We observe that Current Profile is doing as good as Perfect Profile for the companies with either very low or very high $relatedness$ factor. For these companies, the Current Profile is able to capture the required keywords accurately using the mentioned techniques. However, the Current Profile still lags behind Perfect Profile for the

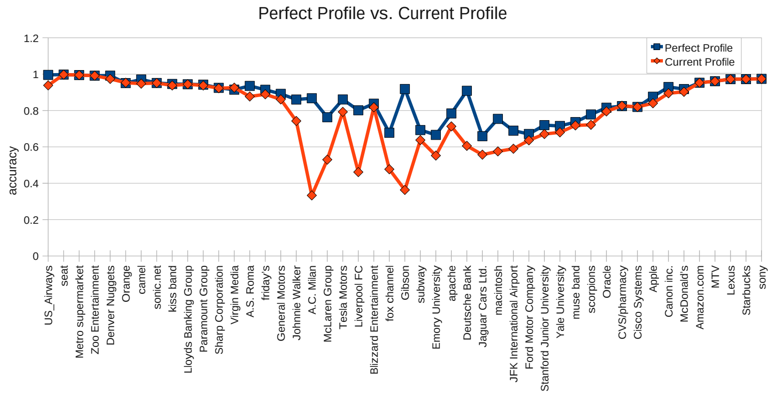

Fig. 4. Comparison of accuracies of Current Profile vs. Perfect Profile.

companies with mid-range $relatedness$ factor. If we want to further improve the classification performance, we need to look into the reasons for the under performance of Current Profile for companies with mid-range $relatedness$ factor.

In Figure 5, we show the comparison of Perfect Profile against Current Profile of a mid-range $relatedness$ factor company (Company name: “Emory University”). The words on the x-axis are arranged in a decreasing order of their occurrence frequency in the test-set collection. The top graph shows the Perfect Profile, with blue-bars referring to the positive weights and red-bars referring the negative weights. The height of the bars indicate associated weight. The lower graph represents the Current Profile of the company. Once again the blue-bars and red-bars indicate positive and negative weights respectively. The green-bars indicate positive weights but their corresponding weights in the Perfect Profile is negative. Similarly yellow-bars indicate negative weights while their corresponding weights in the Perfect Profile is positive. The green and yellow bars in a way contribute towards the reduced performance of the classifier.

Figure 5 helps us understanding the possible reasons for the under-performance of Current Profile in comparison to the Perfect Profile. First, we observe that there are certain words in Perfect Profile, whose corresponding weights in the Current Profile is zero. The Current Profile does not contain any information about these words that are occurring in the Perfect Profile. The reason could be that, when the profile is constructed, those words are not encountered. So, the Current Profile will not be able to classify the tweets containing those words accurately. Second, we observe some words acting as “positive evidence” (i.e. information indicating that the keyword in the message is related to the company) in Perfect Profile are acting as “negative evidence”, indicated by the yellow-bars, and similarly some words acting as “negative evidence” in Perfect Profile are acting as “positive evidence”, indicated by the green-bars. All such words also contribute to some error in the classification. Thirdly, there could be an error because of differences in the weights of words in the Perfect Profile and the Current Profile.

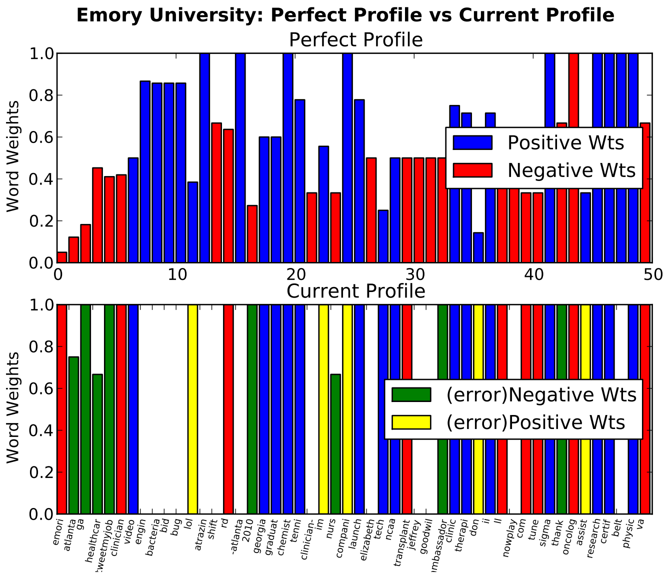

Fig. 5. Comparison of a company's profiles (Current Profile vs. Perfect Profile)

6.2. *Error Groups*

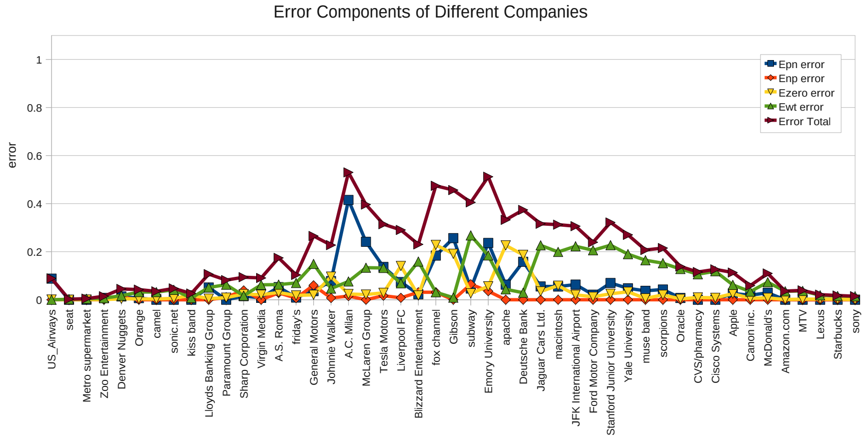

Fig. 6. Different error components contributing towards total classification error

On comparing the Current Profile with Perfect Profile, we have seen the different ways in which the errors could occur. Based on the observations we define three different error groups as follows.

Missing Words Error: (E_{zero}): The Current Profile, under consideration, may not contain some words appearing in the Perfect Profile, i.e. the frequent words that are appearing in the test-set collection. The classifier with the Current Profile in this case would classify all such tweets using the $relatedness$ factor of the company. In this case, the classification error occurs because of these $relatedness$ factor-based decisions. We denote the fraction of incorrect decisions of this type as E_{zero}, that can be computed as follows:

$$E_{zero} = \sum_i (1 - relatedness) \left(\frac{\text{\# of Tweets containing } wrd_i}{\text{\# of Tweets in Test Set}} \right) \tag{12}$$

where wrd_i are the missing words i.e., the words which appear in Perfect Profile but not in Current Profile.

Wrongly Placed Words Error: $E_{PN}(E_{NP})$ is the error caused because of words, which are supposed to be acting as positive (negative) evidence are instead of acting as negative (positive) evidence. The Current Profile classifies all such tweets containing this misplaced words with a confidence proportional to the weights of the misplaced words. So the error introduced will be proportional to the weights of the misplaced words.

$$E_{NP} = E_{PN} = \sum_i \left(\frac{1 + \|wt_i\|}{2} \right) \left(\frac{\text{\# of Tweets containing } wrd_i}{\text{\# of Tweets in Test Set}} \right) \tag{13}$$

where wrd_i are the misplace words i.e, words which are acting as positive (negative) evidence in active-profile are acting as negative (positive) evidence in Current Profile, and wt_i is the weight of the wrd_i in Current Profile.

Words Weights Error: E_{wt} is the error caused because of the differences in the weights of words in the Current Profile and the Perfect Profile. The tweets containing these words (wrd_i) are classified with a confidence of wt_i, weight associated with the word in Current Profile, instead of a confidence of wt_i^p, the weight associated with the word in Perfect Profile.

$$E_{wt} = \sum_i \left(\frac{\|wt_i - wt_i^p\|}{2} \right) \left(\frac{\text{\# of Tweets containing } wrd_i}{\text{\# of Tweets in Test Set}} \right) \tag{14}$$

The above described different error groups, for all the companies, are shown in Figure 6. We see that the majority of the errors is in the middle of the graph, corresponding to the companies with mid-range $relatedness$ factor. We can further see the different components: E_{zero}, E_{PN}, E_{NP} and E_{wt} contribution towards the total error.

6.3. *Reducing the Error Components*

In this section we discuss methods and tradeoffs for reducing errors (of different types defined in Section 6.2).

6.3.1. *Reducing the Missing Words Error (E_{zero})*

We have seen that we construct the profiles using the static information sources (for example, homepages, etc.), that we then extend with keywords from the active twitter streams.

This learning mechanism helps us increase the overlap of words between the Current Profile and the Perfect Profile. It is natural that the longer we inspect the active twitter stream, the higher is the probability of learning new words. Thus the size of the active stream that we inspect, has direct impact on the number of new words that we include in the profile.

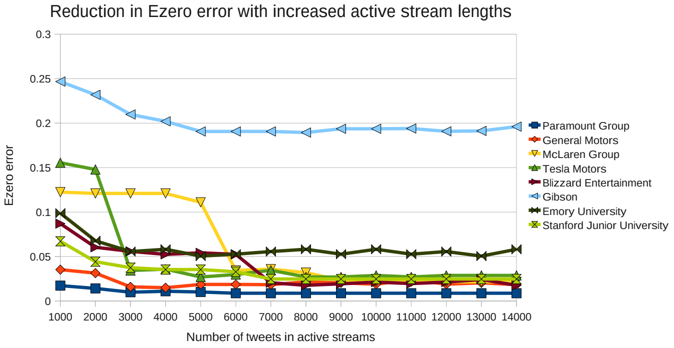

Fig. 7. Reduction of Missing Words Error (E_{zero}) component of selected companies

When the Missing Words Error E_{zero} component, is significant we should try increasing the length of active stream of inspection. We conducted an experiment in which we formed the Current Profile using active streams of increasing length (from the size of 1000 to 14000 tweets per company). Figure 7 shows the impact on the Missing Words Error (E_{zero}), for some mid-range $relatedness$ factor companies, with the increasing the active twitter stream length, we see that the E_{zero} component reduces as the active twitter stream length increases. We observe that even though the error E_{zero} reduces, it never reduces to absolute zero, implying that inspected twitter streams are not containing the words one is expected to find in the test-set collection.

In the next figure we will show the summarized performance for all the companies. Figure 8 shows the reduction in E_{zero} component when the Current Profile uses longer active twitter stream (average length of 8K tweets) instead of a smaller active twitter stream (average length of 2K tweets). We observe the error E_{zero} reduction for the mid-range $relatedness$ factor companies.

6.3.2. *Reducing the Wrongly Placed Words Error (E_{PN} and E_{NP})*

With the previous technique we see that we can increase the overlap of words between the Current Profile and Perfect Profile, but this still does not ensure that we are using the newly found words from the stream correctly. We discuss two possible techniques for reducing the Wrongly Placed Words Error component, with their associated costs.

First, we can make use of stricter controls when deciding if a new word should be acting as positive or negative evidence. We usually identify new words when they are co-occurring with the already existing profile keywords. We can associate a weight for the newly found

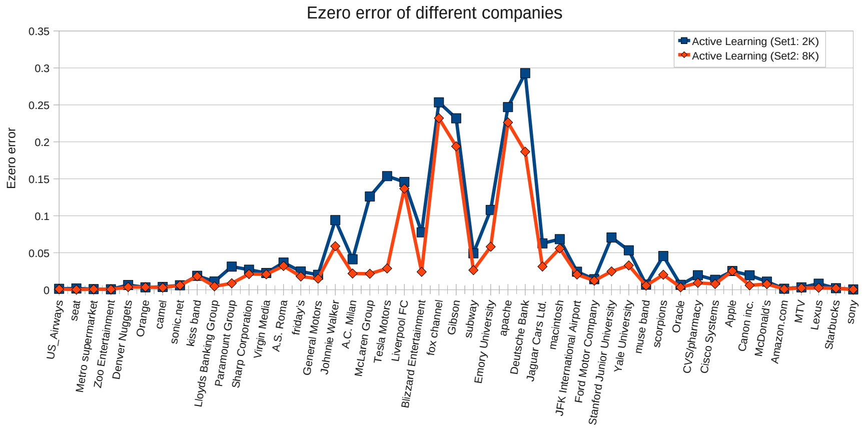

Fig. 8. Comparison of Missing Words Error (E_{zero}) component of all companies for two sets of active stream tweets

words, based on the quality of the words which identified them and also how frequently the newly found word is occurring. One can have stricter controls policy for adding keywords to the profiles, for example by only adding those new words whose weight is above certain predefined threshold. In the experiments section we have already shown that starting with high quality profile, we usually make less error with adding the newly collected words. If we chose very strict control, like very high threshold, we may run in the risk of missing many useful new words, which in turn can increase the error E_{zero} component.

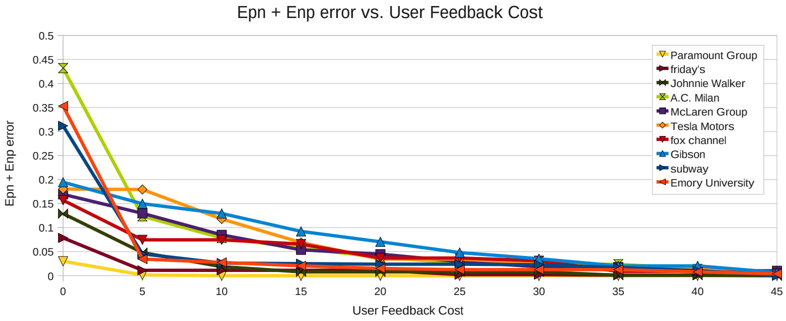

Fig. 9. Comparison of (E_{PN} and E_{NP}) error component of select set of mid-range $relatedness$ factor companies.

Another way of reducing the E_{PN} and E_{NP} error, is to make use of user feedback. We can either make use of user feedback on a selected subset of tweets or on a selected set of frequently occurring keywords. In the remaining of this paper, we make use of the user feedback. We present a set of keywords to the user, who has to evaluate whether they are related to the given company. We treat the number of words to which the user gives feedback as the associated human cost. We conduct an experiment in which we study the impact of error E_{PN} and E_{NP} with respect to the user feedback (cost). Figure 9 shows the impact on the error E_{PN} + E_{NP}, with the increased cost of user feedback, for some selected set of mid-range $relatedness$ factor companies. We see the error reduces at the expense of user feedback. If we have limited budget of human feedback, we should be careful in choosing only those subset of words which can have maximum impact on the overall performance. This is one of the strength of our approach: based on the error analysis, we can chose only those word which are occurring frequently but whose associated weights are smaller than the chosen threshold. In this way we can "optimally" use the costly human input. (In fact, we did not conduct our experiments with human users directly, rather we considered the ground truth as human input. The ground truth itself was created through human effort, see [Amigó *et al.* (2010)].)

6.3.3. *Reducing the Words Weights Error (E_{wt})*

While it is clear how to reduce the errors E_{zero} and E_{PN}+E_{NP} by inspecting longer active twitter streams and efficiently using the human feedback, it is really difficult to reduce the error due to differences in the weights of words in the Current Profile and Perfect Profile. The weights are obtained through heuristic techniques (see Section 3), as no good training set is available. For reducing this error E_{wt} we could construct a training set that represents well the test set, however obtaining a good training set may be difficult.

6.4. *Error reduction techniques impact on the overall accuracy performance*

After seeing the different ways of reducing the individual error components, now we present the impact on the overall accuracy. The following table shows the accuracy performance of different profiles. As Table 4 shows, the error correction techniques explained above further improve the accuracy of our classification techniques. The results using the Current Profile are approaching the ones of Perfect Profile, one could even further improve them, if needed. There are certainly a limitations how close we can get, because of the Words Weights Error component (E_{wt}).

Table 4. Overall accuracy of classification using different error reduction techniques

Different Profiles	**Overall Accuracy**
Perfect Profile	0.87
Current Profile	0.79
Current Profile combined with error E_{zero} reduction technique	0.81
Current Profile combined with error E_{pn}+E_{np} reduction technique	0.83
Current Profile combined with error E_{zero} and error E_{pn}+E_{np} reduction techniques	0.84

7. Related work

The classification of tweets has already been addressed in the literature, in different contexts. Some of the relevant works include [Sriram *et al.* (2010)], [Sankaranarayanan *et al.* (2009)], [Pak and Paroubek (2010)], [Jansen *et al.* (2009)].

In [Sriram *et al.* (2010)], 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.

The authors in [Sankaranarayanan *et al.* (2009)] have built a news processing system based on Twitter. From the twitter stream they have built 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 also build a naive Bayes classifier for distinguishing relevant news tweets from irrelevant ones. They construct the classifier from a training set (that is different from our case). 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 [Jansen *et al.* (2009)] and [Pak and Paroubek (2010)], 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 [Taneva *et al.* (2010)] 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 [Bekkerman (2005)], [Chen *et al.* (2009)], [Kalashnikov *et al.* (2008)], [Yerva *et al.* (2010b)], 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 [Bishop (2006)] 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.* [Yang *et al.* (2006)] 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 [Bishop (2006)] and [Choi *et al.* (2007)] for the tweets classification task.

There are many ways to represent entities. In the Okkam [Miklós *et al.* (2010)] project, which aimed to enable the Web of entities by offering an global entity identification service, an entity is internally represented as a set of attribute-value pairs, along with the meta information related to the evolution of entity. In DBpedia[k] and in Linked Data[l] 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.

In [Perez-Tellez *et al.* (2011)] the authors address the problem of company identification in the micro-blogs by resorting to clustering techniques as a parallel approach to designing classifiers. They propose techniques to improve the representation of a twitter message through term expansion, in a process to enrich the semantic similarity hidden behind the lexical structure.

Authors in [Dan *et al.* (2011)] look into similar problem in a different setting. They address the problem of filtering twitter messages for Social TV purposes. They are concerned if a tweet message is about some popular TV show (Lost, Survivor, Friends etc). Their approach, somewhat similar to ours, is of bootstrapping a model with smaller training set, developing a more sophisticated model using large dataset of unlabeled messages and further using domain specific features to obtain a final classifier. However, their focus was on developing a generic classifier that can be used on any unseen TV show in the training set.

We summarize the different classifiers proposed for the WePS-3 challenge task [Amigó *et al.* (2010)].

The approach presented in [Kalmar (2010)] uses data extracted from the company Website as surrogate training data. This data is used to create a initial model, which is then used to bootstrap a model from the Tweets. The model is iteratively refined with subset of tweets which were confidently classified by the model. The features used are the co-occurring words in each tweet and the relevance of them was calculated according to the Point-wise Mutual Information(PMI) value. Although it seems to be an interesting approach, the results shown provided a lot of scope for improvement. This system -even though it has low on overall accuracy- had decent F-score for relevant tweets, suggesting that a bootstrapping step can be very useful for company names with high ambiguity.

The authors in [Cumberas *et al.* (2010)] based their approach on linguistic aspects like recognizing named entities, extracting external information and making use of predefined rules. They use the well-known Name Entity Recognizer (NER) included in GATE (General Architecture for Text Engineering) for recognizing all the entities in their Tweets. They also use the Web page of the organization, Wikipedia and DBpedia to extract the company related information. Predefined rules are then applied to determine if a Twitter message

[k] `http://dbpedia.org/`
[l] `http://linkeddata.org/`

belongs to an organization or not. The performance of the classifier varied across various companies. It is difficult to predict for what kind of companies this classifier performs well.

The research presented in [Yoshida *et al.* (2010)] proposes two-phase system. In the first phase, they divide the organizations in the training set into 3 or 4 categories depending on the ratio of positive tweets to negative-tweets. In the second-phase, based on simple rules, the classification is done based on the category specific features extracted from the tweets. Their approach is based on the observation that the ratio of positive or negative (if the tweet is related to the organization or not) has a strong correlation with the types of organization names i.e. “organization-like names” have high percentages of tweets related to the company and when compared to “general-word-like names”. Their system performance demonstrated high precision for positive examples and high recall for negative examples.

Another approach is described in [Tsagkias and Balog (2010)], where the focus is on working with organization independent features and not relying on any external information sources. They trained the well-known J48 decision tree classifier using as features the company name, content value such as the presence of URLs, hash-tags or is-part-of-a-conversation (through re-tweeting, denoted in the messages with “RT”), content quality such as ratio of punctuation and capital characters and organizational context. This approach is quite interesting but heavily relies on the availability of training set. In our work we did not exploit the presence of hash-tags or re-tweeting behavior of users.

The basic profile classifier, discussed in Section 3, is based on the LSIR-EPFL classifier [Yerva *et al.* (2010a)], which was the winner of WePS-3 evaluation challenge. The LSIR-EPFL classifier essentially makes use of different information sources on the Web to create an entity profile. We used these profiles for classifying the tweets. We further extended the basic techniques in [Yerva *et al.* (2011a)]. The current paper is a long version of [Yerva *et al.* (2011a)], that gives further details on the work and introduces systematic performance analysis. The same dataset and company profiles were also used in an another line of research on designing quality-aware similarity functions for Web data, in [Yerva *et al.* (2011b)].

8. Conclusion and future work

We studied how to classify Twitter messages containing a keyword, whether they are related to a given company, whose name coincides with the keyword. We proposed several techniques. First we presented a simple Naive Bayes classifier, which relies on automatically or semi-automatically constructed profiles. The company profiles contain two sets of keywords, which indicate whether a tweet containing this keyword is related to the company or not. We then extended this basic technique in two ways. First we developed a method, which takes estimations of the general ambiguity level of the problem into account. We have also introduced a technique that updates our company profiles actively from the twitter stream.

The main advantage of our technique is that it opens the possibility to estimate the accuracy of our classification decision. Indeed, we have exploited this possibility: we analyzed the sources of lower accuracies and we introduced methods to systematically address these problems. In this way we can minimize the uncertainty that is involved in the classification decision. We demonstrated how to localize the cases, where the human input is necessary, that is usually expensive to obtain.

In this way we can handle also the dynamic frequency changes in the use of words in the twitter language. Such changes arise naturally when a company temporarily receives media attention (e.g. if they launch a new product). Our experiments show systematic improvements as we extend our classifier with the described techniques. Though we demonstrated our techniques of entity-based classification on twitter messages, these techniques readily apply for other data sources like comments on social networks or blogs. Equally, one could apply the technique for other types of entities (for which we can obtain similar profiles) as well.

9. Acknowledgements

This work is partially supported by the Network of Excellence Project **PlanetData** (contract no. ICT-257641). We are grateful to the anonymous referees; their comments helped to improve the presentation of the paper.

References

Amigó, E., Artiles, J., Gonzalo, J., Spina, D., Liu, B., and Corujo, A. (2010). Weps3 evaluation campaign: Overview of the on-line reputation management task. In *CLEF (Notebook Papers/LABs/Workshops)*.

Bekkerman, R., and McCallum, A. (2005). Disambiguating Web appearances of people in a social network. In *Proceedings of the 14th International Conference on World Wide Web*, pages 463–470.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Chen, Z., Kalashnikov, D. V., and Mehrotra, S. (2009). Exploiting context analysis for combining multiple entity resolution systems. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 207–218.

Choi, S., Lee, B., and Yang, J. (2007). 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. IEEE Computer Society.

Cumbreras, M. A. G., Vega, M. G., Santiago, F. M., and Perea-Ortega, J.M.(2010). SINAI at WePS-3: Online Reputation Management. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*.

Dan, O., Feng, J., and Davison, B. (2011). A Bootstrapping Approach to Identifying Relevant Tweets for Social TV.

Heckerman, D.(1999). *Learning in graphical models*, chapter A tutorial on learning with Bayesian networks, pages 301–354. MIT Press.

Jansen, B.J., Zhang, M., Sobel, K., and Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 60(11):2169–2188.

Kalashnikov, D. V., Chen, Z., Mehrotra, S., and Nuray-Turan, R. (2008). Web People Search via Connection Analysis. *IEEE Transactions on Knowledge and Data Engineering* , 20(11):1550–1565, November.

Kalmar, P.(2010). Bootstrapping Websites for Classification of Organization Names on Twitter. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*.

Kim, S. B., Rim, H. C., Yook, D. S., and Lim, H. S. (2002). Effective methods for improving naive bayes text classifiers. In Mitsuru Ishizuka and Abdul Sattar, editors, *PRICAI 2002: Trends in Artificial Intelligence*, volume 2417 of *Lecture Notes in Computer Science*, pages 479–484. Springer Berlin / Heidelberg.

Lewis, D. D. (1998). Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, pages 4–15. Springer Verlag.

Miklós, Z., Bonvin, N., Bouquet, P., Catasta, M., Cordioli, D., Fankhauser, P., Gaugaz, J., Ioannou, E., Koshutanski, H., Mana, A., Niederée, C., Palpanas, T., and Stoermer, H. (2010). From Web Data to Entities and Back. In *The 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10)*, volume 6051 of *LNCS*, pages 302–316. Springer.

Pak, A., and Paroubek, P. (2010). 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. European Language Resources Association (ELRA).

Perez-Tellez, F., Pinto, D., Cardiff, J., and Rosso, P. (2011). On the Difficulty of Clustering Microblog Texts for Online Reputation Management. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 146–152, Portland, Oregon, June. Association for Computational Linguistics.

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., and Sperling, J. (2009). 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. ACM.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., and Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceedings of the ACM SIGIR 2010 Posters and Demos*. ACM press.

Taneva, B., Kacimi, M., and Weikum, G. (2010). 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 press.

Tsagkias, M., and Balog, K. (2010). The university of amsterdam at weps3. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*.

Yang, Q., Ding, X., and Tang, X. (2006). Incorporating generic learning to design discriminative classifier adaptable for unknown subject in face verification. *Computer Vision and Pattern Recognition Workshop*, 0:32.

Yerva, S. R., Miklós, Z., and Aberer, K. (2010a). It was easy, when apples and blackberries were only fruits. Padua, Italy. WePS-3, colocated with CLEF.

Yerva, S. R., Miklós, Z., and Aberer, K. (2010b). 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)*.

Yerva, S. R., Miklós, Z., and Aberer, K. (2011a). What have fruits to do with technology? The case of Orange, Blackberry and Apple. In *International Conference on Web Intelligence, Mining and Semantics* (WIMS'2011),Sogndal,Norway.

Yerva, S. R., Miklós, Z., and Aberer, K. (2011b). Quality-aware similarity assessment for entity matching in Web data. In *Information Systems Journal* 2011, Elsevier.

Yoshida, M., Matsushima, S., Ono, S., Sato, I., and Nakagawa, H. (2010). ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*.

Appendix

Table 5. WePS-3 Testset Companies Information

ID	Company Entity	Query Term	Company URL
1	Amazon.com	Amazon	http://www.amazon.com
2	Apache	apache	http://www.apache.org/
3	Apple	Apple	http://www.apple.com
4	Blizzard Entertainment	Blizzard	http://www.blizzard.com
5	camel	camel	http://en.wikipedia.org/wiki/Camel_(cigarette)
6	Canon inc.	canon	http://www.usa.canon.com/home
7	Cisco Systems	Cisco	http://www.cisco.com/
8	CVS/pharmacy	CVS	http://www.cvs.com/CVSApp/user/home/home.jsp
9	Denver Nuggets	nuggets	http://en.wikipedia.org/wiki/Denver_Nuggets
10	Deutsche Bank	Deutsche	http://en.wikipedia.org/wiki/Deutsche_Bank
11	Emory University	emory	http://www.emory.edu/
12	Ford Motor Company	ford	http://www.ford.com
13	fox channel	fox	http://www.fox.com/
14	friday’s	friday’s	http://www.tgifridays.com/
15	Gibson	Gibson	http://www.gibson.com
16	General Motors	GM	http://www.gm.com/
17	Jaguar Cars Ltd.	jaguar	http://www.jaguar.com/
18	J. F. Kennedy Int. Airport	jfk	http://www.jfkiat.com/
19	Johnnie Walker	johnnie	http://www.johnniewalker.com/en-us/home
20	kiss band	kiss	http://www.kissonline.com/
21	Lexus	Lexus	http://www.lexus.com/
22	Liverpool FC	Liverpool	http://www.liverpoolfc.tv/
23	Lloyds Banking Group	Lloyd	http://www.lloydsbankinggroup.com/
24	macintosh	mac	http://www.apple.com/mac/
25	McDonald’s	McDonald’s	http://www.mcdonalds.com
26	McLaren Group	McLaren	http://mclaren.com/home
27	Metro supermarket	Metro	http://www.metro.ca/
28	A.C. Milan	ACMilan	http://www.acmilan.com/index.aspx
29	MTV	MTV	http://www.mtv.com/
30	Muse band	muse	http://muse.mu/
31	Oracle	oracle	http://www.oracle.com/index.html
32	Orange	Orange	http://www.orange.com/en_EN/
33	Paramount Group	Paramount-Group	http://www.paramount-group.com/
34	A.S. Roma	Roma	http://en.wikipedia.org/wiki/A.S._Roma
35	Scorpions	scorpions	http://www.the-scorpions.com/english/
36	Seat	seat.com	http://www.seat.com
37	Sharp Corporation	sharp	http://www.sharp.eu
38	Sonic.net	sonic	http://sonic.net/
39	Sony	sony	http://www.sony.com/
40	Stanford Junior University	stanford	http://www.stanford.edu/
41	Starbucks	Starbucks	http://www.starbucks.com/
42	Subway	subway	http://www.subway.com
43	Tesla Motors	tesla	http://www.teslamotors.com/
44	US_Airways	Usairways	http://www.usairways.com/
45	Virgin Media	VirginMedia	http://www.virginmedia.com
46	Yale University	Yale	http://www.yale.edu/
47	Zoo Entertainment	zoo	http://en.wikipedia.org/wiki/Zoo_Entertainment