Filter Keywords and Majority Class Strategies for Company Name Disambiguation in Twitter*

Damiano Spina, Enrique Amigó, and Julio Gonzalo

UNED NLP & IR Group
Juan del Rosal, 16
28040 Madrid, Spain
{damiano,enrique,julio}@lsi.uned.es
http://nlp.uned.es

Abstract. Monitoring the online reputation of a company starts by retrieving all (fresh) information where the company is mentioned; and a major problem in this context is that company names are often ambiguous (*apple* may refer to the company, the fruit, the singer, etc.). The problem is particularly hard in microblogging, where there is little context to disambiguate: this was the task addressed in the WePS-3 CLEF lab exercise in 2010. This paper introduces a novel *fingerprint* representation technique to visualize and compare system results for the task. We apply this technique to the systems that originally participated in WePS-3, and then we use it to explore the usefulness of *filter keywords* (those whose presence in a tweet reliably signals either the positive or the negative class) and finding the majority class (whether positive or negative tweets are predominant for a given company name in a tweet stream) as signals that contribute to address the problem. Our study shows that both are key signals to solve the task, and we also find that, remarkably, the vocabulary associated to a company in the Web does not seem to match the vocabulary used in Twitter streams: even a manual extraction of filter keywords from web pages has substantially lower recall than an oracle selection of the best terms from the Twitter stream.

1 Introduction

Monitoring the online reputation of a company starts by retrieving all (fresh) information where the company is mentioned; and a major problem in this context is that company names are often ambiguous. For instance, the query "Apple" retrieves information about Steve Jobs' company, but also about apples (the fruit) Fiona Apple (the singer), etc. The problem becomes particularly hard in microblogging streams such as `Twitter.com`, because the available context to disambiguate each tweet is much smaller than in other media.

* This research was partially supported by the Spanish Ministry of Education via a doctoral grant to the first author (AP2009-0507) and the Spanish Ministry of Science and Innovation (Holopedia Project, TIN2010-21128-C02).

P. Forner et al. (Eds.): CLEF 2011, LNCS 6941, pp. 50–61, 2011.

Our work deals with the challenge of providing an unsupervised solution for this problem, i.e. a system that accepts a company's name (plus a representative URL) as input and provides a binary classification of tweets (which mention the company's name) as related or non-related to the company. In practice, this could be a filtering component for services such as `SocialMention.com`, where the query "apple" produces aggregated figures of *strength*, *sentiment*, *passion*, *reach*, etc. for all sources mentioning apple, whether they refer to the company, the fruit, the singer, etc.

This research is based upon two intuitive observations:

(i) **Filter Keywords:** Manual annotation can be simplified by picking up special keywords (henceforth called *filter keywords*) that isolate positive or negative information. For instance, "ipod" is a positive filter keyword for apple tweets[1], because its presence is a highly reliable indicator that the tweet is about the apple company. Reversely, "crumble" is a negative filter keyword for apple, because it correlates with non-related tweets. The intuition is that automatic detection of such filter keywords can be a valuable information to provide an automatic solution to the problem.

(ii) **Majority Class:** The ratio of positive/negative tweets is extremely variable between company names, and does not follow a normal distribution; on the contrary, it seems to have a skewed distribution (at least when considering a short time frame): typically, either most tweets are about the company, or most tweets are unrelated to the company. Therefore, predicting which of the situations hold for a certain company name might be a valuable input for algorithmic solutions to the problem.

Our goal is to provide quantitative evidence supporting (or rejecting) our intuitions. We will use the WePS-3 Task 2 test collection[2] [1], which is, to our knowledge, the first dataset built explicitly to address this problem. This paper also introduces the *fingerprint* representation technique, a visualization of system results that is particularly useful to understand the behavior of systems in classification/filtering scenarios where the test cases have variable class skews.

In Section 2 we start by briefly discussing the state of the art, and introducing the *fingerprint* representation. In Section 3 we test our two hypotheses, and in Section 4 we discuss how to locate filter keywords automatically.

2 State of the Art

Disambiguation of company names is a necessary step in the monitorization of opinions about a company. However, this problem is not tackled explicitly in most research on the topic, where it is normally assumed that query terms are not ambiguous in the retrieval process. The disambiguation task has been explicitly addressed in the WePS-3 evaluation campaign [1]. In this section we revisit the results of that campaign using a novel *fingerprint* representation technique that helps understanding and comparing system results.

[1] We will refer to microblog entries as tweets – being Twitter the most prominent example of microblogging network – in the remainder of the article.

[2] `http://nlp.uned.es/weps/weps-3`

2.1 WePS-3: Test Collection and Systems

The Online Reputation Management task of the WePS-3 evaluation campaign consists of filtering Twitter posts containing a given company name depending of whether the post is actually related with the company or not. The WePS-3 ORM task dataset consists of 52 training and 47 test cases, each of them comprising a company name, its URL, and a set of tweets consisting of an average of 435 tweets manually annotated as related/non-related to the company. A total of five research groups participated in the campaign. The best two systems were LSIR [7] and ITC-UT [8].

The LSIR system builds a set of profiles for each company, made of keywords extracted from external resources such as WordNet or the company homepage, as well as a set of manually defined keywords for the company and the most frequent unrelated senses for the company name. These profiles are used to extract tweet-specific features that are added to other generic features that give information about the quality of the profiles to label the tweets as related or unrelated with an SVM classifier. The ITC-UT system is based on a two step classification. Firstly, it predicts the class of each query/company name according to the ratio of related tweets of each company name and secondly applies a different heuristic for each class, basically based on the PoS tagging and the named entity label of the company name.

The SINAI system [3] also uses a set of heuristic rules based on the occurrence of named entities both on the tweets and on external resources like Wikipedia, DBPedia and the company homepage. The UVA system [6] does not employ any resource related to the company, but uses features that related with the use of the language on the collection of tweets (URLs, hashtags, capital characters, punctuation, etc.). Finally, the KALMAR system [4] builds an initial model based on the terms extracted from the homepage to label a seed of tweets and then uses them in a bootstrapping process, computing the point-wise mutual information between the word and the target's label.

In the WePS exercise, accuracy (ratio of correctly classified tweets) was used to rank systems. The best overall system (LSIR) obtained 0.83, but including manually produced filter keywords. The best automatic system (ITC-UT) reaches an accuracy of 0.75 (note that 0.5 is the accuracy of a random classification), and includes a query classification step in order to predict the ratio of positive/negative tweets. These results motivate us to analyze in depth the potential contribution of filter keywords and majority class signals.

2.2 A *Fingerprint* Visualization to Compare Systems

One of our intuitions is that predicting which is the majority class for a given microblogging stream is a substantial step towards solving the problem. The reason is that the ratio of related tweets does not follow a normal distribution: in general, for a given company name and a (small) period of time, either most

tweets refer to the company, or most tweets are unrelated. Even in the WePS-3 dataset, where organizers made an effort to include company names covering all the spectra of positive ratio values, the ratio is very low or very high for many companies.

In this context, average performance measures (accuracy or F) are not sufficiently informative of the system's behavior. That's our main motivation to propose a *fingerprint* representation technique to visualize system results. Figure 1 illustrates our method, which consists of displaying the accuracy of the system (vertical axis) for each company vs. the ratio of related (positive) tweets for the company (horizontal axis). Each dot in the graph represents one of the test cases (i.e. the accuracy of the system for the set of tweets containing the name of one of the companies in the dataset).

The advantage of this representation is that the three basic baselines (all true, all negative and random) are displayed as three fixed lines, independently of the dataset. The performance of the "all true" baseline classification corresponds exactly with the proportion of true cases, and therefore is the $y = x$ diagonal in the graph. The "all negative", correspondingly, is represented by the $y = 1 - x$ diagonal. And, finally, the random baseline is the horizontal line $y = 0.5$.

Note that, for averaged measures – such as accuracy or F –, the results of the baselines depend on the dataset: for instance, the "all true" baseline depends on the average number of true cases in the corpus used for evaluation. As our fingerprint representation is constant for that baselines, it is easier to identify when a system is having a baseline-like approach, and for which subset of the data.

Figure 1 shows the fingerprint representation of the best runs per participant. The first observation is that the "all true" and "all false" baseline lines seem to have a "magnetic" attraction on the dots; in other words, somehow most systems have a tendency to make a "winner takes all" decision on each of the test cases. The second observation is that the fingerprint representation makes a clear distinction between each system's behavior. SINAI is the system that most clearly tends to apply a winner takes all strategy, with a strong preference for the "all false" choice. KALMAR results cluster around the "all true" baseline, with deviations towards randomness. The two best systems, LSIR manual and ITC-UT, have a tendency towards the "all false" baseline when the true ratio is lower than 0.5, and towards the "all true" baseline when the true ratio is greater than 0.5. In the area around 0.5, LSIR is able to improve on both baseline strategies, with the dots spreading over a "glass of martini" shaped area. Finally, UVA is the only system which has no baseline behavior at all, with accuracy values which seem largely independent on the related tweets ratio.

The fingerprint representation of WePS-3 systems proves to be very useful to understand systems' behavior, and highlights the importance of the related tweets ratio: the best systems behave as if they were guessing which is the majority class for each test case, and most of the rest tend to behave like one of the baselines.

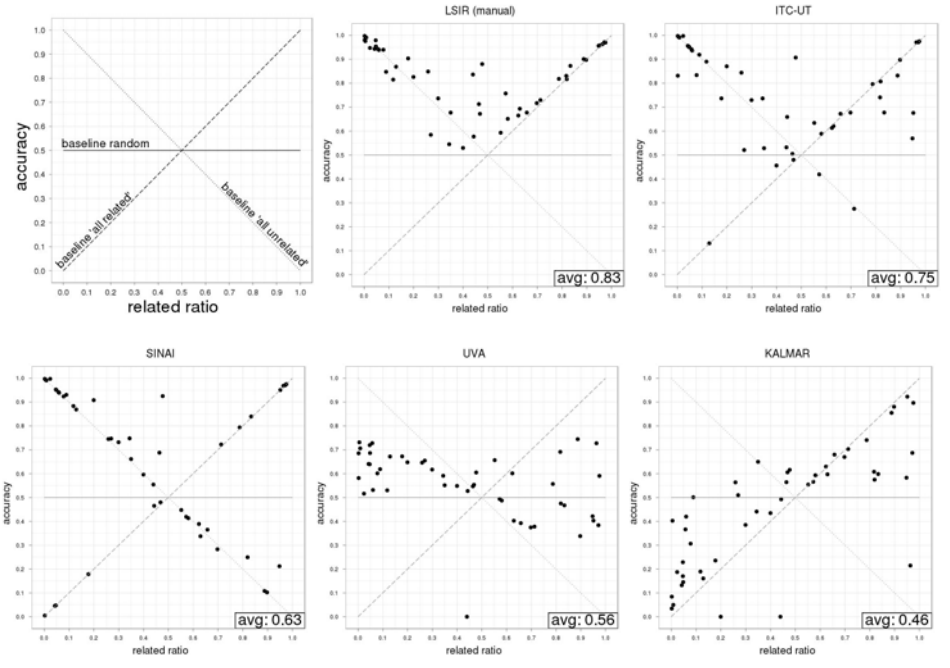


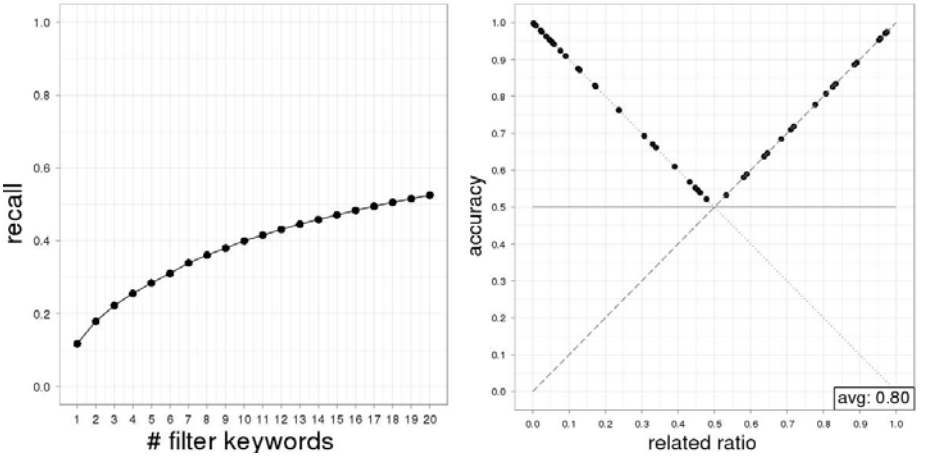
Fig. 1. Fingerprint technique and visualization of the best runs of participants in the WePS-3 exercise

3 Hypotheses Testing

3.1 Upper Bound Performance of Filter Keywords

A positive/negative filter keyword is an expression that, if present in a tweet, indicates a high probability that it is related/non-related to the company. The most useful filter keywords are those with a high recall, i.e., those which appear in as many tweets as possible. As all tweets in the WePS-3 collection are manually annotated as related/non-related to their respective company name, we can find exactly how many filter keywords there are (by definition, filter keywords are those terms that only appear in either the positive or the negative tweets), and how much recall they provide. Figure 2a shows the recall of the first n filter keywords (for $n = 1 \ldots 20$) in the test collection. Recall at step n is the proportion of tweets covered by adding the keyword that filters more tweets among those which were not still covered by the first n-1 keywords. We will hereafter refer to these perfect keyword selection as *oracle keywords*.

The graph shows that, in average, the best five oracle keywords cover around 30% of the tweets, and the best ten reach around 40% of the tweets. That is, only the best five discriminative terms directly cover around 130 out of 435 tweets in each stream, which could in turn be used to build a supervised classifier



(a) (b)

Fig. 2. Upper bounds of the two hypotheses: filter keywords (a) and majority class (b)

to be applied on the remaining tweets. This indicates that filter keywords are potentially a relevant source of information to address the problem.

A priori, the natural place to find filter keywords is the Web: the company's web domain, references to this domain in Wikipedia, ODP, etc. and the Web at large. Using the company's URL and web search results for the company name, a human annotator performed a manual selection of positive and negative keywords for all the companies in WePS-3 corpus. Note that the annotator did not have access to the tweets in the corpus.

Remarkably, manual keywords extracted from the Web (around 10 per company) only reach 14.61% coverage of the tweets (compare with 39.97% for 10 oracle keywords extracted from the tweets themselves), with an accuracy of 0.86 (lower than expected for manually selected filter keywords). This seems an indication that the vocabulary and topics of microblogging are different from those found in the Web. Our experiments in Section 4 corroborate this finding.

3.2 Upper Bound Performance of Winner-Takes-All Strategy

In the hypothesis that the tweet stream will be skewed (most tweets will be either related or unrelated to the company), a system that, given a company name, simply guesses which of the two situations is more likely and applies a "winner takes all" classification, may already reach a high average accuracy[3].

[3] Of course this is not a solution to the problem from a practical perspective; if there are just a few relevant tweets, we will miss them and we will have no data to analyze. But knowing the majority class may allow us, for instance, to select that class for a tweet in the absence of other relevant signals.

Let us consider an upper bound performance for this strategy, in which this decision is always taken correctly. Results on the WePS-3 testbed are shown in Figure 2b. All points are placed on the upper part of the diagonals, because they are the best possible individual choice between "all positive" or "all negative" for each of the test cases. The average accuracy is 0.80, which is almost comparable with the best manual WePS-3 participation, that obtained an accuracy of 0.83 [7], and superior to the best automatic system of the WePS-3 campaign (with 0.75 accuracy) [8]. These results confirm that an accurate determination of the majority class can be crucial to solve the problem.

Notice that the distribution of related ratios across company names in the WePS-3 corpus streams was enforced to a nearly uniform distribution [1]. In a real scenario, the dots in the fingerprint visualization would be more concentrated in the left and right areas, the effect of predicting the majority class would be even stronger, and therefore the upper bound in real world conditions would be even higher.

3.3 Upper Bound Combination of Strategies

An interesting question is whether filter keywords are enough to predict the majority class. The leftmost graphs in Figure 3 show the results when tagging all tweets for a given company as related or non related depending on the majority class in oracle and manual keywords respectively. Notably, the resulting graph for oracle keywords is almost identical to the *winner-takes-all* upper bound, and results in the same overall accuracy (0.80). Manual keywords, on the other hand, tend to behave as the "all related" baseline, with an average accuracy of 0.61. The reason is probably that there are less manual negative keywords than positive.

The central graphs show the results for the *winner-takes-remainder* strategy, which consists of applying the winner-takes-all strategy only to those tweets that were not previously classified by some of the filter keywords. This straightforward combination of filter keywords and winner-takes-all strategies gives 0.85 accuracy for 20 oracle keywords.

It is also interesting to compare the *winner-takes-all* and the *winner-takes-remainder* approaches with a standard bootstrapping method: we have represented tweets as bag-of-words (produced after tokenization, lowercase and stopword removal) with a binary weighting function for the terms in the instance (occurrence); then we have employed a C4.5 decision tree classification model[4]. For each test case, we use the tweets retrieved by the keywords as seed (training set) in order to classify automatically the rest of tweets. The obtained average accuracy is 0.87 for oracle keywords, only 0.02 absolute points above the *winner-takes-remainder* approach; the graph shows that the improvement resides in cases with a related ratio around 0.5, i.e. the cases where it is more likely to have enough training samples for both classes. Table 1 displays results

[4] We also tried with other machine learning methods, such as linear SVM and Naive Bayes, obtaining similar results.

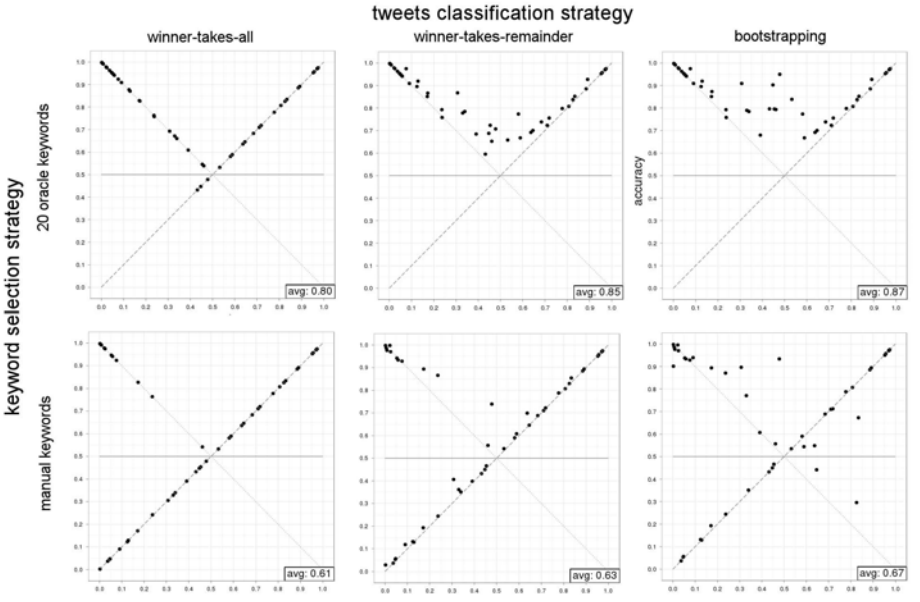


Fig. 3. Fingerprints for each of the three different classification strategies when applying 20 oracle keywords or manual keywords

Table 1. Quality of the three different classification strategies when applying oracle/manual filter keywords

keyword selection strategy	seed set recall	seed set accuracy	overall accuracy winner-takes-all	overall accuracy winner-takes-remainder	overall accuracy bootstrapping
5 oracle keywords	28.45%	1.00	0.80	0.81	0.81
10 oracle keywords	39.97%	1.00	0.80	0.83	0.85
15 oracle keywords	47.11%	1.00	0.80	0.84	0.86
20 oracle keywords	52.55%	1.00	0.80	0.85	0.87
manual keywords	14.61%	0.86	0.61	0.63	0.67

for different amounts of filter keywords: the bootstrapping strategy ranges from 0.81 (with five keywords) up to 0.87 with 20 keywords.

4 Automatic Discovery of Filter Keywords

Our goal is to discover automatically the terms which are most strongly associated to the company name (*positive* filter keywords) or to the alternative meanings of the company name (*negative* keywords), and to discard those which are not discriminative (*skip* terms). To that end, we first discuss a number of potentially helpful term features, and then apply various machine learning and heuristic algorithms that operate on those features.

Table 2. Notation used to describe term features

Item	Description
t, t_i	term
c	(ambiguous) name that identifies a company (e.g. `jaguar`)
T	set of tweets in the WePS-3 collection[a]
T_c	set of tweets in the collection for a given company name c.
$df_t(T_c)$	document frequency of term t in the collection T_c.
$df_{\text{web}}(q)$	number of total hits returned by the Yahoo! Search BOSS API (`http://developer.yahoo.com/search/boss/`) for the query q.
M	an approximation of the size of the search engine index ($30 \cdot 10^9$).
$domain_c$	domain of the web site given as reference for the company c.
$wikipedia(q)$	set of Wikipedia pages returned by the MediaWiki API (`http://www.mediawiki.org/wiki/API`) for the query q.
$dmoz(q)$	set of items (composed by an URL, a title, a description and a category) returned by searching q on the Open Directory Project (`http://www.dmoz.org/search`)

[a] For each company name, the dataset to which its belongs is only used (either training or test dataset).

4.1 Term Features

Table 2 summarizes the notation used to explain the features. The terms represented are those that are not stopwords and appear at least in five different tweets on T_c, given a company c. For each of these terms, 13 features grouped in three types were computed: *Collection-based features* (*col_**), *Web-based features* (*web_**), and *Features expanded by co-occurrence* (*cooc_**): Given a feature f, a new feature (meant to address the sparseness problem) is computed as the Euclidean norm (1) of the vector composed by $f_{t_i} * w(t, t_i)$ for the five terms most co-occurrent with t in the set of tweets T_c (2), where f_{t_i} is the value of feature f for the term t_i and $w(t, t_i)$ is the the grade of co-occurrence between the terms t and t_i.

$$cooc\_agg(t, f) = \sqrt{\sum_{i \in cooc_t} (f(t_i) * w(t_i))^2} \tag{1}$$

$$cooc_t = \text{ set of the five most co-occurrent terms with } t. \tag{2}$$

Table 3 describes the 13 features computed to represent the terms.

4.2 Keyword Discovery

We experiment with three approaches to keyword discovery. The first one, *machine learning*, consists of training a positive-negative-skip classifier over the training corpus in WePS-3 by using the features described previously. We combine two classifiers; positive versus others, and negative versus others. Then, we state one threshold for each classifier. Terms which are simultaneously under/over both thresholds are tagged as skip terms. We have tried with several machine learning methods using Rapidminer [5]: Multilayer Perceptron with Backpropagation (Neural Net), C4.5 and CART Decision Trees, Linear Support

Table 3. Features computed to represent the terms

Feature	Description
col_c_df = $df_t(T_c)/\lvert T_c \rvert$)	Normalized document frequency over the collection of the company tweets.
col_c_specificity = $df_t(T_c)/df_t(T)$	Ratio of document frequency on the company collection T_c over the document frequency on the collection T.
col_hashtag	Number of occurrences that the term appears as hashtag on T_c.
web_c_assoc = $\frac{df_{\text{web}}(t \text{ OR } c)/df_{\text{web}}(c)}{df_{\text{web}}(t)/M}$	Represents the association, according to the search counts, between the term t and a company name c.
web_c_ngd = $\frac{\max(\log(f(c)),\log(f(t))-\log(f(t \text{ AND } c))}{M-\min(\log(f(t)),\log(f(c))}$ where $f(x) = df_{\text{web}}(x)$	The Normalized Google Distance [2] (applied to the Yahoo! search engine) between a term t and a company name c
web_dom_df = $\frac{df_{\text{web}}(t \text{ AND site:}domain_c)}{df_{\text{web}}(\text{site:}domain_c)}$	Normalized document frequency of the term in the website of the company.
web_dom_assoc = $\frac{\text{web\_dom\_df}}{df_{\text{web}}(t)/M}$	Analogous to web_c_assoc, using the website domain instead of the company name c.
web_odp_occ	Number of occurrences of the term on all the items retrieved in $dmoz(domain_c)$.
web_wiki_occ	Number of occurrences of the term on the first 100 results in $wikipedia(domain_c)$.
cooc_c_assoc, cooc_c_ngd, cooc_dom_df, cooc_dom_assoc	Features col_c_assoc, web_c_ngd, web_dom_df and web_dom_assoc expanded by co-occurrence.

Vector Machines (SVM) and Naive Bayes. According to the ROC curves of those classifiers, Neural Nets are slightly superior.

The second approach (*heuristic*) is inspired by an analysis of the signal provided by each of the features, and consists of a heuristic which involves the two most informative features. First, we define a threshold to remove skip terms which are not specific enough with respect to the tweet set (using the *col_c_spec* feature). Then we state, using the feature that measures association with the website (*cooc_dom_assoc*) a minimal value to consider a keyword positive and a maximal value under which keywords are considered negative. These three thresholds were manually optimized using the training data set. Finally, the *hybrid* approach consists of applying machine learning as in the first method, but using only the two features applied in the heuristic approach.

After detecting the filter keywords automatically, we classify the tweets that contain only negative or only positive keywords. Then, in order to classify the rest of tweets, we apply the same three alternative methods described in Section 3: winner-takes-all, winner-takes-remainder and bootstrapping.

Table 4 shows the results of our experiments. The best automatic system (machine learning to discover keywords and bootstrapping with the tweets annotated using that keywords) gives an accuracy of 0.73, which is much better than using manual keywords (0.67) and close to the best automatic result in the competition (0.75). Note that the winner-takes-remainder strategy, which is a direct combination of the two strategies (filter keywords plus winner-takes-all)

Table 4. Results for automatic keyword detection strategies

keyword selection strategy	seed set recall	seed set accuracy	overall accuracy winner-takes-all	overall accuracy winner-takes-remainder	overall accuracy bootstrapping
m. learning	57.81%	0.72	0.64	0.68	**0.73**
heuristic	27.42%	0.79	0.64	0.65	0.71
hybrid	38.64%	0.78	0.70	0.72	0.72

reaches 0.72; this is a strong baseline which shows that both signals are indeed useful to solve the problem.

Discovery of filter keywords has proved to be challenging using signals from the Web: the accuracy of the resulting seed set ranges between 0.72 and 0.79, with a large recall only in the case of the machine learning strategy. As with the manual selection of keywords, the biggest problem is finding suitable negative keywords. Overall, this result reinforces the conclusion that the characterization of companies in Twitter, in terms of vocabulary, is different from the characterization in Web pages.

5 Discussion and Conclusions

This paper makes a few novel contributions to the problem of company name disambiguation in microblog entries: (i) we introduce a *fingerprint* visualization technique to understand and compare systems' behavior (ii) we show that finding *filter keywords* and determining the majority class in a set of tweets are two relevant sources of information to solve the problem; and (iii) we have seen that the vocabulary that characterizes a company in Twitter is substantially different from the vocabulary associated to the company in its home page, in Wikipedia, and in the Web at large.

We have seen that in the WePS-3 dataset, five optimal keywords directly classify around 30% of the tweets in average (providing valuable information to characterize the positive and negative classes), and assigning the true majority class to all tweets (*winner-takes-all* strategy) gives .80 accuracy (which is higher than the best published automatic result on this test collection).

We have also set upper and lower bound classification performances using only these two signals to disambiguate tweets:

- **Upper bound:** Using 20 *oracle* (perfect) filter keywords and a winner-takes-all guess based on the majority class in the tweets that contain the filter keywords we reach .85 accuracy without any additional information or algorithmic machinery.
- **Lower bound:** If we replace oracle keywords with automatically extracted filter keywords we reach .72 accuracy, which is close to the best published result on WePS-3 dataset (.75).

Remarkably, the contribution of the naive winner-takes-all strategy to the lower bound is not easy to improve. If we replace it with a Machine Learning process (using tweets with filter keywords as training set) we reach .73 accuracy,

which represents only a 1.4% improvement. Our experiments also indicate that, although both signals are useful to solve the problem, extracting filter keywords and guessing the majority class are challenging tasks. In the case of filter keywords, we have seen that the vocabulary that characterizes a company in the Web has only a moderate overlap with its microblogging counterpart. In fact, a manual keyword selection from Web sources only reaches 14.61% coverage with 0.86 accuracy. Descriptions of the company in its home page, in Wikipedia, and in the Web at large, provide useful but weak signals to the problem of filter keyword identification, and it is far from trivial to build an accurate classifier with them. Negative keywords (those which signal tweets which are not related to the company) are particularly hard to discover automatically. Guessing the majority class seems to be, apparently, a simpler problem, but in practice it turns out to be equally hard.

Our future work naturally addresses two questions: how to find filter keywords (and determine the majority class) more accurately, and how to integrate both types of information in a full-fledged algorithmic solution to the problem. Given that we have only used these two signals to disambiguate so far, we believe that merging them with other useful signals should provide an optimal solution to the problem.

References

1. Amigó, E., Artiles, J., Gonzalo, J., Spina, D., Liu, B., Corujo, A.: WePS-3 Evaluation Campaign: Overview of the Online Reputation Management Task. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)
2. Cilibrasi, R.L., Vitanyi, P.M.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering (2007)
3. García-Cumbreras, M.A., García-Vega, M., Martínez-Santiago, F., Peréa-Ortega, J.M.: SINAI at WePS-3: Online Reputation Management. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)
4. Kalmar, P.: Bootstrapping Websites for Classification of Organization Names on Twitter. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)
5. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid prototyping for complex data mining tasks. In: SIGKDD 2006: Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining (2006)
6. Tsagkias, M., Balog, K.: The University of Amsterdam at WePS3. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)
7. Yerva, S.R., Miklós, Z., Aberer, K.: It was easy when apples and blackberries were only fruits. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)
8. Yoshida, M., Matsushima, S., Ono, S., Sato, I., Nakagawa, H.: ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)