Expert Systems with Applications 40 (2013) 4986–5003

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Discovering filter keywords for company name disambiguation in twitter ☆

Damiano Spina *, Julio Gonzalo, Enrique Amigó

UNED NLP & IR Group, C/ Juan del Rosal 16, 28040 Madrid, Spain

ARTICLE INFO

Keywords:
Twitter
Online reputation management
Name disambiguation
Filtering

ABSTRACT

A major problem in monitoring the online reputation of companies, brands, and other entities is that entity names are often ambiguous (*apple* may refer to the company, the fruit, the singer, etc.). The problem is particularly hard in microblogging services such as Twitter, where texts are very short and there is little context to disambiguate. In this paper we address the filtering task of determining, out of a set of tweets that contain a company name, which ones do refer to the company. Our approach relies on the identification of *filter keywords*: those whose presence in a tweet reliably confirm (positive keywords) or discard (negative keywords) that the tweet refers to the company.

We describe an algorithm to extract filter keywords that does not use any previously annotated data about the target company. The algorithm allows to classify 58% of the tweets with 75% accuracy; and those can be used to feed a machine learning algorithm to obtain a complete classification of all tweets with an overall accuracy of 73%. In comparison, a 10-fold validation of the same machine learning algorithm provides an accuracy of 85%, i.e., our unsupervised algorithm has a 14% loss with respect to its supervised counterpart.

Our study also shows that (i) filter keywords for Twitter does not directly derive from the public information about the company in the Web: a manual selection of keywords from relevant web sources only covers 15% of the tweets with 86% accuracy; (ii) filter keywords can indeed be a productive way of classifying tweets: the five best possible keywords cover, in average, 28% of the tweets for a company in our test collection.

1. Introduction

The vast use of social media to share facts and opinions about entities, such as companies, brands and public figures has generated the opportunity – and the necessity – of managing the online reputation of those entities. Online Reputation Management consists of monitoring – and handling – the opinion of Internet users (also referred to as electronic word of mouth, eWOM) on people, companies and products, and is already a fundamental tool in corporate communication (Dellarocas et al., 2004; Hoffman, 2008; Pollach, 2006; Wilson, 2003).

Over the last years, a wide variety of tools have been developed that facilitate the monitoring task of online reputation managers.[1] The process followed by these tools typically consists of three tasks:

1. **Retrieval of potential mentions.** The user gives as input a set of keywords (e.g. the company name, products, CEO's name…), and the service uses these keywords to retrieve documents and content generated by users from different sources: broadcast news sources, social media, blogs, site reviews, etc.
2. **Analysis of results.** Retrieved documents are automatically processed in order to get relevant information to the user: sentiment, authority and influence, background topics, etc.
3. **Results visualization.** Analyzed data is presented to the user in different ways: ranking documents, drawing graphics, generating tag clouds, etc.

☆ This research was partially supported by the Spanish Ministry of Education (FPU Grant No. AP2009-0507), the Spanish Ministry of Science and Innovation (Holopedia Project, TIN2010-21128-C02), the Regional Government of Madrid and the ESF under MA2VICMR (S2009/TIC-1542) and the European Community's FP7 Programme under Grant agreement No. 288024 (LiMoSINe).

* Corresponding author. Tel.: +34 913988106.
E-mail addresses: damiano@lsi.uned.es (D. Spina), julio@lsi.uned.es (J. Gonzalo), enrique@lsi.uned.es (E. Amigó).
URL: http://nlp.uned.es (D. Spina).

[1] At the time of writing this paper, some popular reputation management tools are trackur (http://www.trackur.com/), BrandsEye (http://www.brandseye.com/), Alterian SM2(http://socialmedia.alterian.com/) or SocialMention (http://socialmention.com), among others.

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.03.001

A major problem concerning the first task (retrieval of potential mentions) is that brand names are often ambiguous. For instance, the query "Ford" retrieves information about the motor company, but also might retrieve results about Ford Models (the modeling agency), Tom Ford (the film director), etc.

One might think that the query is too general, and the user should provide a more specific query, such as "ford motor" or "ford cars". In fact, some tools explicitly suggest the user to refine possibly ambiguous queries.[2]

This approach has two main disadvantages: (i) users have to make an additional effort when defining unambiguous queries and (ii) query refinement harms recall over the mentions of the brand in the Web, which can be particularly misleading in an online reputation management scenario.

Note that filtering out the mentions that do not refer to the monitored entity is also crucial when estimating its visibility. Quantifying the number of mentions on the Web about an entity, and how this number changes over time, is essential to track marketing or Public Relationship campaigns. When the entity name is ambiguous, indicators given by tools such as Google Trends[3] or Topsy[4] can be misleading.

We think that a component capable of filtering out mentions that do not refer to the entity being monitored (specified by the user as a keyword plus a representative URL) would be a substantial enhancement of current online reputation management tools, and would also facilitate the analysis of the online presence/visibility of a brand.

Among the most important information sources monitored by reputation managers are microblogging services (Comm and Robbins, 2009; Jansen et al., 2009), As of today, Twitter[5] is the most popular microblogging service and provides a communication environment that deserves serious attention as a form of eWOM. Users send short posts (*tweets*) where they can describe things of interest and express attitudes that they are willing to share with others.

The work presented in this paper deals with the challenge addressed by the Online Reputation Management Task of WePS-3 (Amigó et al., 2010). Given a set of Twitter entries containing an (ambiguous) company name, and given the home page of the company, the task is to filter out irrelevant information, providing a binary classification of tweets as related or unrelated to the company. Notice that ambiguity resolution is particularly challenging in Twitter: tweets are minimal (140 characters at most) and little context is available for resolving name ambiguity.

The main objective of this paper is to validate an intuitive observation derived by the set-up and the analysis of the results of the WePS-3 Online Reputation Management Task (Amigó et al., 2010). The observation is that manual annotation can be simplified by picking up special keywords –henceforth called *filter keywords*– that reliably signal positive or negative information. For instance, "ipod" is a positive filter keyword for the Apple company, because its presence is a highly reliable indicator that the tweet is about the company. Reversely, "crumble" is a negative filter keyword for Apple, because it correlates with unrelated tweets. The intuition is that automatic detection of such filter keywords can be a valuable signal to design an automatic solution to the problem.

Our goal is to provide quantitative evidence supporting (or rejecting) our intuition, and to answer some related questions:

- Where should we look for filter keywords in order to find them automatically?
- How much recall can we expect from such keywords?
- Can we use the notion of filter keywords to build a binary classifier that solves the task?

In order to answer these questions we will use the WePS-3 Task 2 test collection,[6] which is, to our knowledge, the first dataset built explicitly to address this problem.

The paper is organized as follows: in Section 2 we start by introducing some background concepts and discussing the State of the Art: we review some related work on Twitter, we summarize previous work on Named Entity Disambiguation and Automatic Keyword Extraction (which is one of the techniques we want to apply to the problem). Then we focus on the goals and results of the WePS-3 Online Reputation Management Task, which is the test collection used in our experiments.

In Section 3 we detail and validate our Filter Keywords hypothesis by computing its upper bounds on our test collection. In Section 4, we study how to automatically discover filter keywords. Then, in Section 5 we discuss how to perform the classification task by using automatically extracted filter keywords. Finally, we discuss our results in Section 6 and conclude in Section 7.

2. Related work

Filtering out mentions that do not refer to a given company name can be seen as a named entity disambiguation problem. Named Entity Disambiguation (NED) have been receiving a lot of attention from the Natural Language Processing (NLP) research community in the last decade (Artiles, 2009; Bagga and Baldwin, 1998; Bunescu and Pasca, 2006; Cucerzan, 2007; Grishman and Sundheim, 1996; Ji et al., 2010). Most recent work tackle this problem in contexts where disambiguation is more challenging, as in the case of Twitter: the textual context is reduced (limited to 140 characters) and language is used as chat-speak or SMS style, where the use of the phonetic spelling and acronyms for common phrases is vast (Gouws et al., 2011; Kaufmann and Kalita, 2010).

In this section we provide some background about Twitter and its related datasets. Then we introduce some previous work on NED, focusing on scenarios where NED is crucial such as Entity Linking, Document enrichment by Wikipedia links and Web People Search. Related work about automatic keyword extraction – one of the techniques we want to apply to our problem – is also presented, and then we describe the definition and the results of the WePS-3 Online Reputation Management Task – which is the test collection used in our experiments.

2.1. Twitter

Twitter[7] is a relatively new social networking site (Krishnamurthy et al., 2008) which stands out as the prototypical microblogging service. Its particularity is that posts do not exceed 140 characters and there are no privacy conditions. Therefore, Twitter reflects opinions in real time and is very sensitive to burstiness phenomena (Boyd et al., 2010; Kwak et al., 2010; Milstein et al., 2008).

Tweets are particularly challenging for Natural Language Processing tasks, given that (i) tweets are shorts (i.e. 140 characters), and (ii) users post text using a nonstandard language with similarities to SMS style (Kaufmann and Kalita, 2010; Laboreiro et al., 2010, 2011).

[2] Alterian SM2 service: http://www.sdlsm2.com/social-listening-for-business/industry/.

[3] http://www.google.com/trends.

[4] http://www.topsy.com.

[5] http://twitter.com.

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

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

2.1.1. Twitter-related tasks

Up to now, most of the work on Twitter has focused on analyzing the microblogging phenomenon (Boyd et al., 2010; Java et al., 2007; Kwak et al., 2010), modeling the propagation of information in the social network (Krishnamurthy et al., 2008; Wu et al., 2011; Yang and Leskovec, 2010) and analyzing the content of tweets (Cheng et al., 2010; Hurlock and Wilson, 2011; Jansen et al., 2009; Sakaki et al., 2010; Sriram et al., 2010).

Some previous work address the problem of tweet classification for different purposes such as sentiment analysis (Jansen et al., 2009; Pak and Paroubek, 2010) and information filtering (Sriram et al., 2010). Sriram et al. (2010) classify tweets in five categories (i.e. news, events, opinions, deals, and private messages) in order to improve information filtering. They compare different tweet representations, including bag-of-words and features like the author name, the presence of mentions of users on the tweet and features regarding the writing style (i.e. opinionated words, currency and percentage signs). Then, they train a Naive Bayes classification model to evaluate the different representations, obtaining high accuracy scores when combining bag-of-words with the other features.

Jansen et al. (2009) use a multinomial Bayes model to determine the sentiment of a set of tweets. The model is built using a lexicon of around 200,000 unigrams and bigrams of words and phrases that have a probability distribution to determine the sentiment of a topic. The system calculates the probability of each tweet by checking the occurrence of polarized words and then picks the class with the greatest probability in a winner-takes-all scenario. In this work we use a similar approach, in the manner that we test, among others, the winner-takes-all strategy in order to classify all the tweets for a company name either as related or unrelated.

Zhao et al. (2011b) propose an automatic keyphrase extraction model to summarize topics in Twitter. Firstly, they identify topics using Twitter-LDA (Zhao et al., 2011a), which assumes a single topic assignment for each tweet. Secondly, they use a variation of Topical PageRank (Liu et al., 2010) where edges between two words are weighted taking into account co-occurrences in tweets assigned to the same topic. Then, they generate phrase candidates by looking for combinations of the top topic keywords that co-occur as frequent phrases in the text collection. Finally, topical keyphrases are ranked by using a probabilistic model that takes into account (i) the specificity of a phrase given a topic and (ii) the retweet ratio of tweets containing a keyphrase.

To the best of our knowledge, the Online Reputation Management Task on WePS-3 held at CLEF 2010 was the first campaign of NLP-centered tasks over Twitter. TREC 2011 and TREC 2012 held a track about Twitter: TREC Microblog Track.[8] Here, the problem addressed is a realtime search task (Realtime Adhoc Task): given a query at a specific time, systems should return a list both recent and relevant tweets (Soboroff et al., 2012).

2.1.2. Twitter datasets

There are several Twitter datasets that are suitable for a variety of research purposes.

A corpus of 900.000 tweets has been provided by the *Content Analysis in Web 2.0* Workshop (CAW 2.0),[9] to tackle the problem of text normalization on user generated contents. Yang and Leskovec (2011) use a dataset of 580 million Tweets to identify temporal patterns over the content published on the tweets, and Kwak et al. (2010) use a representative sample of 467 million tweets from 20 million users covering a 7 month period from June 1 2009 to December 31 2009 to study the information diffusion and the topological characteristics of Twitter.[10] Cha et al. (2010) built a dataset comprising 54,981,152 users, connected to each other by 1,963,263,821 social links and including a total of 1,755,925,520 tweets to analyze users' influence and how to measure it on the Twittersphere.

Finally, the Tweets2011 corpus is the dataset that will be used on the TREC Microblog Track. The corpus is a representative sample of the twittersphere, that includes also spam tweets. It consists of approximately 16 million tweets over a period of 2 weeks (24th January 2011 until 8th February, inclusive), which covers both the time period of the Egyptian revolution and the US Superbowl, among others.

To the best of our knowledge, WePS-3 Task 2 test collection (described in Section 2.4), is the first dataset specifically built to address the problem of disambiguation of organization names on tweets. Recently, the RepLab evaluation campaign (Amigó et al., 2012) held in CLEF 2012, has addressed the same problem but in a multilingual scenario: the RepLab dataset contains tweets written in English and Spanish.

2.2. Named Entity Disambiguation

In Natural Language Processing, Named Entity Disambiguation (NED) is a step further from Named Entity Recognition. The latter consists on identifying and classifying phrases in a text that refer to people, places or organizations, as well as temporal expressions or numeric quantities (Grishman and Sundheim, 1996; Nadeau and Sekine, 2007). The former involves the association of mentions in one or more texts (also called references or surface forms) of an entity with the concrete object that they are actually referencing (Bagga and Baldwin, 1998). For instance, in the sentence: *"The Big Apple's new Apple retail store was officially opened today."* the two occurrences of the word *Apple* refer to two different entities. The former refers to New York City (nicknamed as "The Big Apple"[11]), while the latter refers to the consumer electronics and software company Apple Inc[12].

Word Sense Disambiguation (WSD) is another NLP task related to NED, which deals with the polysemy of common words, and consists of assigning the appropriate sense to an occurrence of a word in a given context (Agirre and Edmonds, 2006). While in WSD the senses of a word are often looked up in a dictionary, thesaurus or lexical resource such as WordNet (Edmonds and Cotton, 2001), we will see that in NED the use of Wikipedia as knowledge source is widely extended.

In the following, we review three main problems –closely related with ours– in which entity disambiguation has been applied: Entity Linking, Document Enrichment by Linking to Wikipedia Articles and Web People Search. Finally, we conclude this section describing some previous work about named entity disambiguation on Twitter.

2.2.1. Entity linking

Entity linking consists of associating a mention in a text with the corresponding entity in a Knowledge Base. In the Knowledge Base Population scenario (KBP) (Dredze et al., 2010; Ji and Grishman, 2011; Ji et al., 2010; McNamee and Dang, 2009), this is a first step to discover facts about entities and augmenting a knowledge base with these facts and with newly discovered entities. Given an entity name and a background document where the name occurs, systems typically perform three steps to link the entity to the knowledge base: (i) query expansion (enrich the query by mining the Wikipedia structure or resolving co-reference in the back-

[8] https://sites.google.com/site/microblogtrack.

[9] http://caw2.barcelonamedia.org/node/41.

[10] These datasets are no longer available as per request from Twitter.

[11] http://en.wikipedia.org/wiki/Big_Apple.

[12] http://en.wikipedia.org/wiki/Apple_Inc.

ground document); (ii) candidate generation (find all possible entries in the knowledge base that the query might link to) and (iii) candidate ranking (rank candidate entities by computing similarity between the represented query and the entities, and fixing a threshold to decide when the entity does not exist in the knowledge base).

2.2.2. Document enrichment by linking to Wikipedia articles

There are a variety of systems that automatically annotate a document by linking names appearing in the document with Wikipedia articles (Bunescu and Pasca, 2006; Kulkarni et al., 2009; Mihalcea and Csomai, 2007; Milne and Witten, 2008; Strube and Ponzetto, 2006).

In general, named entity disambiguation in these systems is carried out in three steps:

- **Mention or surface form representation.** In this step, the context of the mention to disambiguate is defined. The most common representations used are vector space model (Bunescu and Pasca, 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007), the set of named entities that occur in the text (Gentile et al., 2010), and the resolved Wikipedia links of unambiguous entities next to the mention (Ferragina and Scaiella, 2010; Kulkarni et al., 2009; Meij et al., 2012; Milne and Witten, 2008).
- **Candidates entities retrieval and representation.** The system retrieves all possible entities that could be referenced by the mention from the knowledge base (i.e. Wikipedia pages) and represents each entity as a bag-of-words from the page (Bunescu and Pasca, 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007), extracting features from the page structure (such as the categories which the page belongs to) (Bunescu and Pasca, 2006; Cucerzan, 2007; Gentile et al., 2010), or syntactic features (Mihalcea and Csomai, 2007) or also exploiting the hyperlink structure of Wikipedia, retrieving all pages that link to the candidate entity page (Ferragina and Scaiella, 2010; Kulkarni et al., 2009; Meij et al., 2012; Milne and Witten, 2008).
- **Best candidate selection.** In the final step, the best candidate entity is selected by computing a distance or similarity function between the surface form and each of the candidate entities. The most common functions are cosine (Bunescu and Pasca, 2006) and other vector similarity functions (Cucerzan, 2007; Mihalcea and Csomai, 2007), random walk graph models to compute semantic relatedness (Gentile et al., 2010) and finally relatedness or coherence functions that involve all the entity links made in the text (Ferragina and Scaiella, 2010; Kulkarni et al., 2009; Milne and Witten, 2008).

Unlike document enrichment, our task focuses on a single organization name and does not require linking every mention of entities in the collection, but just deciding whether each mention refers or not to the entity.

2.2.3. Web People Search

Web People Search is defined as the task of clustering a set of web pages, which are the result of a Web search for a person name, in as many groups as entities sharing that name (Artiles, 2009). This was the original goal of the Web People Search campaign (WePS) (Artiles et al., 2007), complemented in subsequent editions by the tasks of person attribute extraction (Artiles et al., 2009b, 2010) and company name disambiguation in Twitter (Amigó et al., 2010). In the web people search scenario, Hierarchical Agglomerative Clustering seems to be the most competitive (and most frequently used) clustering technique (Artiles et al., 2007, 2009b, 2010; Gooi and Allan, 2004). Documents are typically represented as bag-of-words (Artiles et al., 2007; Bagga and Baldwin, 1998; Kalashnikov et al., 2007). Other works use smaller portions of the document, such as sentences where the ambiguous name occurs or pre-defined windows of words (Artiles et al., 2009b; Gooi and Allan, 2004; Mann, 2006). Named entities are also a frequently used feature for people name disambiguation (Artiles et al., 2007, 2009b; Kalashnikov et al., 2007), and biographical features (e.g. title, organization, email address, phone number, etc.) are used to a lesser extent (Al-Kamha and Embley, 2004; Mann and Yarowsky, 2003; Wan et al., 2005). The most common similarity measure in this scenario is cosine (Artiles et al., 2007, 2009b; Bagga and Baldwin, 1998), while some works also use Kullback–Leibler divergence (Gooi and Allan, 2004; Martinez-Romo and Araujo, 2009).

2.2.4. Named Entity Disambiguation in Twitter

Most Named Entity Disambiguation techniques have been applied to disambiguate entities on reasonably long texts such as news articles or blog posts. However, little work has been done on NED over microblogging posts (Ferragina and Scaiella, 2010; Meij et al., 2012; Michelson and Macskassy, 2010). In this scenario, disambiguation is harder due to fact that texts are short (limited by 140 characters) and hence the context of a mention is minimal.

Ferragina and Scaiella (2010) propose a system capable of annotating short texts (including tweets) by linking entities to Wikipedia pages. The system relies on the hyperlink structure of Wikipedia, exploiting the links between pages and the anchors texts of the links. When the system receives a text, it detects the anchors on the text and retrieve the possible senses for each anchor. Ambiguous senses are disambiguated by a collective agreement function among all senses associated to the anchors detected on the text (Kulkarni et al., 2009), and taking advantage of the unambiguous anchors to boost the selection of these senses for the ambiguous anchors (Milne and Witten, 2008). They do not evaluate the accuracy of their disambiguation component over tweets. However, they report that almost 95% of the 5,000 analyzed tweets have at least 3 phrases with an entry in Wikipedia (which is not necessarily an entity). This result shows that Wikipedia have a high coverage as a catalog of senses for tweet disambiguation.

Different from Ferragina and Scaiella (2010), Meij et al. (2012) uses supervised machine learning techniques to refine a list of candidate Wikipedia concepts that are potentially relevant to a given tweet. The candidate ranking list is generated by matching n-grams in the tweet with anchor texts in Wikipedia articles, taking into account the inter-Wikipedia link structure to compute the most probable Wikipedia concept for each n-gram.

Michelson and Macskassy (2010) focus on discovering the topics of interest for a particular Twitter user. Given the stream of tweets corresponding to the user, they firstly find the Wikipedia page of the entities mentioned on tweets and secondly they build a topic profile from the high-level categories that cover these entities. Entity disambiguation is performed by calculating the overlap between the terms on the tweet and the terms on the page of each candidate entity. In this scenario, the accuracy of the disambiguation process is not critical, since the system takes into account the categories that occur frequently across all the entities found in order to produce a topic profile of a stream.

2.3. Automatic Keyphrase Extraction

We include here an overview of Automatic Keyphrase Extraction techniques, as this is a technique that plays a crucial role in our approach to company name disambiguation on Twitter.

Automatic Keyphrase Extraction is the task of identifying a set of relevant terms or phrases that summarize and characterize one or more given documents (Witten et al., 1999). Most of the literature about automatic keyphrase extraction is focused on (well-written) technical documents, such as scientific and medical articles, since the keywords given by the authors can be used as gold standard (Frank et al., 1999; Kim et al., 2010; Mihalcea and Tarau, 2004; Milios et al., 2003; Turney, 1999; Witten et al., 1999). Some authors address automatic keyword extraction as a way of automatically summarizating web sites (Zhang et al., 2004a, 2004b, 2005, 2007). In Zhang et al. (2007) different keyword extraction methods are compared, including tf*idf, supervised methods, and heuristics based on both statistical and linguistic features of candidate terms. While Zhang et al. (2007) study the automatic keyword extraction from website descriptions, in our work we explore a semi-supervised keyword extraction approach that extract filter keywords over a stream of tweets.

Automatic keyword extraction is typically used to characterize the content of one or more documents, using features intrinsically associated with that documents. In order to detect both positive and negative filter keywords, however, we need to look into external resources in order to discriminate between related and unrelated keywords. Thus, automatic keyword extraction methods are not directly applicable to our filter keyword approach.

2.4. The WePS-3 Online Reputation Management Task

Disambiguation of company names in text streams (and in particular in microblog posts) is a necessary step in the monitoring of opinions about a company. However, it is not tackled explicitly in most research on the subject. Rather than this, most previous work assume that query terms are not ambiguous in the retrieval process. The disambiguation task has been explicitly addressed in the WePS-3 evaluation campaign (Amigó et al., 2010). In this section we summarize the outcome of that campaign, analyzing the test collection and comparing system results.

2.4.1. Task Definition and Test Collection

The Online Reputation Management task of the WePS-3 evaluation campaign consists of filtering Twitter posts containing a given company name, depending on whether the post is actually related with the company or not. The task is defined to deal with the scenario where an online system accepts any company name as input, and has to learn to disambiguate entries about that company without supervision (i.e. without a set of previously disambiguated tweets). Therefore, the set of organization names in the training corpora is different from the set of companies in the test set.

For each organization in the dataset, systems are provided with the company name c (e.g. apple) used as query to retrieve the stream of tweets to annotate T_c, and a representative URL (e.g. http://www.apple.com) that univocally identifies the target company. The input information per tweet consists of a tuple containing: the tweet identifier, the organization name, the query used to retrieve the tweet, the author identifier, the date and the tweet content. Systems must label each tweet as *related* (i.e. the tweet refers to the company) or *unrelated* (the tweet does not refer to the given company).

The WePS-3 ORM task dataset comprises 52 training and 47 test cases, each of them including a company name, its URL, and an average of 435 tweets manually annotated as related/unrelated to the company. The dataset has been annotated using the Mechanical Turk crowdsourcing marketplace,[13] with the following manual annotation options: *related*, *unrelated* and *undecidable*. Each hit has been redundantly annotated by five Mechanical Turk workers. A total of 902 annotators have participated in the annotations of 43,730 tweets. Finally, an agreement analysis was done in order to decide the final annotation for each tweet.

An interesting property of the dataset is that there is a great variability of the degree of ambiguity across the training and test cases. That is, there are companies with low occurrence in tweets (e.g. Delta Holdings, Zoo Entertainment), companies with medium ambiguity (e.g. Luxor Hotel and Casino, Edmunds.com) and companies with high presence in tweets (e.g. Yamaha, Lufthansa).

2.4.2. WePS-3 Results

A total of five research groups participated in the campaign. The best two systems were LSIR (Yerva et al., 2010) and ITC-UT (Yoshida et al., 2010). 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 (García-Cumbreras et al., 2010) 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 (Tsagkias and Balog, 2010) does not employ any resource related to the company, but uses features that involve the use of the language in the collection of tweets (URLs, hashtags, capital characters, punctuation, etc.). Finally, the KALMAR system (Kalmar, 2010) 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 (being 0.5 the accuracy of a random classification), and includes a query classification step in order to predict the ratio of positive/negative tweets.

2.4.3. Other work using WePS-3 datasets

Recently, Yerva et al. (2012) have explored the impact of extending the company profiles presented in Yerva et al. (2010) by using the related ratio and considering new tweets retrieved from the Twitter stream. By estimating the degree of ambiguity per entity from a subset of 50 tweets per entity, they reach 0.73 accuracy.[14] Using this related ratio and considering co-occurrences with a given company profile, the original profile is extended with new terms extracted from tweets retrieved by querying the company name in Twitter. The expanded profile outperforms the original, achieving 0.81 accuracy.

Zhang et al. (2012) presents a two stages based disambiguation system. They combine supervised and semi-supervised methods. Firstly, they train a generic classifier using the training set, similarly to Yerva et al. (2010). Then, they use this classifier to annotate

[13] http://www.mturk.com.

[14] Note that, unlike the original formulation of the WePS-3 task, this is a supervised system, as it uses part of the test set for training. Hence, their results cannot be directly compared with the results in our work.

a seed of tweets in the test set, using the Label Propagation algorithm to annotate the remainder tweets in the test set. Using Naïve Bayes and Label Propagation they achieve a 0.75 accuracy, that matches the performance of the best automatic system in WePS-3.

The RepLab evaluation campaign held at CLEF 2012 addressed, as a subtask, the same filtering problem introduced in WePS-3. The main difference was that while WePS-3 dataset contains only tweets written in English, the RepLab collection (Amigó et al., 2012) contains tweets both in English and Spanish.

Finally, the WePS-3 ORM task dataset has been extended with manual annotations for the task of entity profiling in microblog posts, that includes identifying entity aspects and opinion targets (Spina et al., 2012).

2.5. Wrap up

There are two main findings on entity name disambiguation that motivate our research:

1. **Use of filter keywords.** Artiles et al. (2009a) studied the impact of query refinement in the Web People Search clustering task and concluded that

   "although in most occasions there is an expression that can be used as a near-perfect refinement to retrieve all and only those documents referring to an individual, the nature of these ideal refinements is unpredictable and very unlikely to be hypothesized by the user" (Artiles et al., 2009a).

This is both a positive indication – there are keywords able to isolate relevant information well – and a suggestion that finding optimal keywords automatically might be a challenging task.

Another evidence in favor of using filter keywords is that the best results on the WePS-3 ORM Task were achieved by a system that used a set of manually produced –both positive and negative– filter keywords (Yerva et al., 2010). One of the goals of our work is to analyze query refinement in the scenario of the Company Name Disambiguation on Twitter. In other words, we explore the impact of defining a set of keywords to filter both related and unrelated tweets to a given company.

1. **Use of knowledge bases to represent candidate entities.** Both entity linking and document enrichment systems use knowledge bases in order to characterize the possible entities that a mention may refer to. Most systems use Wikipedia as knowledge base (Bunescu and Pasca, 2006; Cucerzan, 2007; Dredze et al., 2010; Ferragina and Scaiella, 2010; Gentile et al., 2010; Ji et al., 2010; Kulkarni et al., 2009; Mihalcea and Csomai, 2007; Milne and Witten, 2008). In this direction, we believe that looking at Wikipedia pages related to the company to disambiguate could give useful information to characterize positive filter keywords. We also explore the use of other resources such as Open Directory Project (ODP), the company website and the Web in general. Note, however, that not every company is listed in ODP or has an entry in Wikipedia.

3. Potential benefits of filter keywords for name disambiguation in Twitter

A positive/negative filter keyword is an expression that, if present in a tweet, indicates a high probability that the tweet is related/unrelated to the company. In this section, we investigate the upper bound of the filter keyword strategy, in two ways: firstly, we consider manual annotations in the WePS-3 collection to derive *oracle* (optimal) keywords; and secondly, manually extracting keywords from representative Web pages about the company. Finally, we study how to use these filter keywords to solve the WePS-3 ORM task.

3.1. Upper Bound Performance of Filter Keywords

The most useful filter keywords are those with a high coverage, i.e., those which appear in as many tweets as possible.

As all tweets in the WePS-3 collection are manually annotated as related/unrelated 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. Fig. 1 shows the coverage of the first n filter keywords (for $n = 1 \ldots 20$) in the test collection.

Coverage 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 this optimal 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 cover around 40% of the tweets. Note that only five discriminative terms directly cover, in average, 130 out of 435 tweets in each stream, and those could in turn be used to build a supervised classifier for the rest of the tweet stream. This indicates that filter keywords are, potentially, a relevant source of information to address the problem.

In principle, the natural place to find filter keywords is the Web: the company's web domain and reference 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, we performed a manual selection of positive and negative keywords for all the companies in the WePS-3 corpus. Note that the annotator inspected pages in the web search results, but did not have access to the tweets in the corpus.

Tables 1 and 2 show some examples for positive and negative keywords, respectively. Note that in the set of Oracle keywords there are expressions that a human would hardly choose to describe a company (at least, without previously analyzing the tweet stream). For instance, the best positive oracle keywords for the *Fox Entertainment Group* do not include intuitive keywords such as `tv` or `broadcast`; instead, they include keywords closer to breaking news (`leader`, `denouncing`, etc.).

Looking at negative keywords (Table 2), we can find occasional oracle keywords that are closely related with the vocabulary used in microblogging services, such as `followdaibosyu`, `nowplaying`

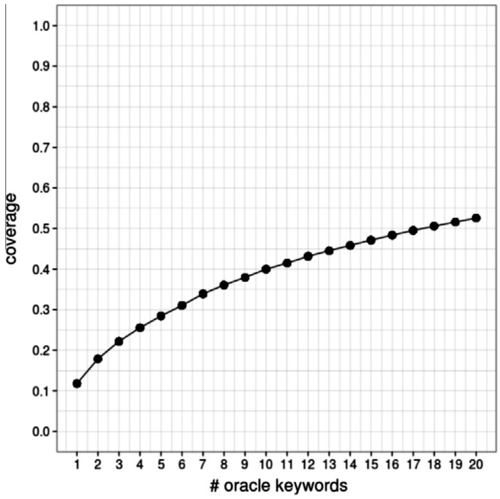


Fig. 1. Upper bound of the filter keywords strategy.

Table 1
Differences between oracle and manual positive keywords for some of the company names on the test collection.

Company name	Oracle positive keywords	Manual positive keywords
amazon	sale, books, deal, deals, gift	electronics, apparel, books, computers, buy
apple	gizmodo, ipod, microsoft, itunes, auction	store, ipad, mac, iphone, computer
fox	money, weather, leader, denouncing, viewers	tv, broadcast, shows, episodes, fringe, bones
kiss	fair, rock, concert, allesegretti, stanley	tour, discography, lyrics, band, rockers, make up design

Table 2
Differences between oracle and manual negative keywords for some of the company names on the test collection.

Company name	Oracle negative keywords	Manual negative keywords
amazon	followdaibosyu, pest, plug, brothers, pirotta	river, rainforest, deforestation, bolivian, brazilian
apple	juice, pie, fruit, tea, fiona	fruit, diet, crunch, pie, recipe
fox	megan, matthew, lazy, valley, michael	animal, terrier, hunting, Volkswagen, racing
kiss	hug, nowplaying, lips, day, wanna	french, Meg Kevin, bang bang, Ryan Kline

Table 3
Quality of seed sets and the bootstrapping classification strategy when applying oracle/manual filter keywords.

Keyword selection strategy	Seed set		Bootstrapping overall accuracy
	Coverage (%)	Accuracy	
5 oracle keywords	28	1.00	0.81
10 oracle keywords	40	1.00	0.85
15 oracle keywords	47	1.00	0.86
20 oracle keywords	53	1.00	0.87
manual keywords	15	0.86	0.67

or `wanna`, while intuitive manual keywords like `wildlife` for jaguar are unlikely to occur in the Twitter collection.

Remarkably, manual keywords extracted from the Web (around 10 per company) only reach 15% coverage of the tweets (compare with 40% coverage using 10 oracle keywords extracted from the tweet stream), with an accuracy of 0.86 (which is 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. Using keywords to solve the task

So far, we have seen that keywords do not cover all tweets in the collection, but a part of them. Then, we need an additional step in order to complete the task: given a seed of tweets, annotate the tweets that remain uncovered by filter keywords on each test case. To this aim, we use a standard bootstrapping method. Tweets are represented as bag-of-words (produced after tokenization, lowercase and stop word removal) and term occurrence is used as weighting function; then we have employed a C4.5 Decision Tree classification model[15] Quinlan (1993) –with its default parameters– using the implementation provided by the Rapidminer toolkit (Mierswa et al., 2006). For each stream, we use the tweets retrieved by the keywords as seed (training set) in order to classify automatically the rest of tweets.

Table 3 displays results for different amounts of filter keywords: the bootstrapping strategy ranges from 0.81 (with 5 keywords) up to 0.87 with 20 keywords. On the other hand, using the tweets covered by manual keywords as training set, the bootstrapping achieves only a 0.67 accuracy.

In order to better understand the results, Fig. 2 shows the *fingerprint* representation (Spina et al., 2011) for manual keywords and 20 oracle keywords. This visualization technique consists of displaying the accuracy of the system (vertical axis) for each company/test case (dots) vs. the ratio of related (positive) tweets for the company (horizontal axis). The three basic baselines (all true, all negative and random) are represented as three fixed lines: $y = x$, $y = 1 - x$ and $y = 0.5$, respectively. The fingerprint visualization method helps in understanding and comparing systems' behavior, specially when class skews are variable over different test case.

Using 20 oracle keywords (see Fig. 2(b)), the obtained average accuracy is 0.87. The fingerprint 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. Manual keywords, on the other hand, lead to annotations that tend to stick to the "all related" or "all unrelated" baselines, which indicates that they tend to describe only one class, and then the learning process is biased.

In summary, our results validate the idea that filter keywords can be a powerful tool in our filtering task, but also suggest that they will not be easy to find: descriptive web sources that can be attributed to the company do not lead to the keywords that are most useful or accurate in the Twitter domain.

4. Automatic discovery of filter keywords

Our next step is now is to discover automatically the terms which are most strongly associated to the company name (*positive* filter keywords) and to the alternative meanings of the company name (*negative* filter keywords), and to discard those which are not discriminative (*skip* terms). For this, we take a machine learning approach, in which training data corresponds to a set of company names and test data to a different set of companies. Thus, the learning process must be able to generalize across companies. Each term is represented by features that take into account the company's website, Wikipedia, ODP, the Web at a large and the WePS-3 collection itself.

In this section we start discussing the features that we propose to represent terms; then, we perform a statistical analysis of the features, and finally we report the results of experiments with our dataset.

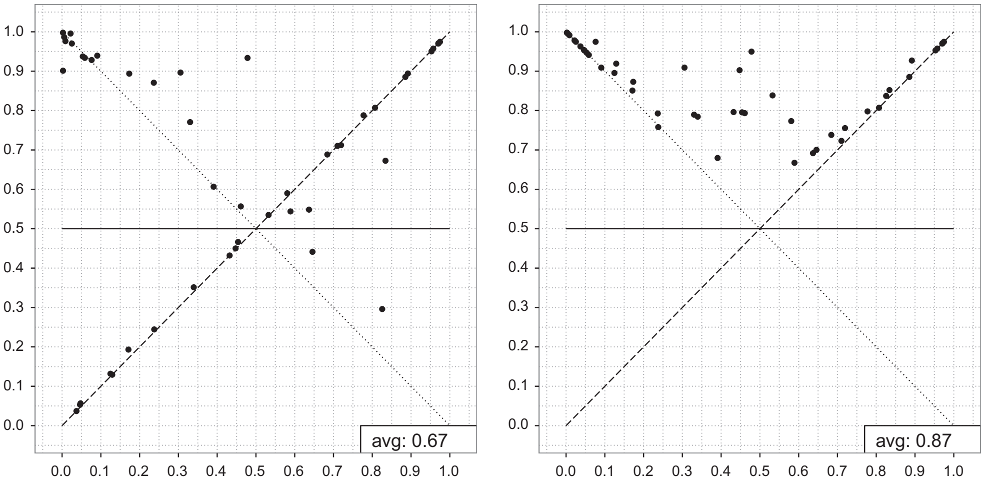
4.1. Term features

Table 4 summarizes the notation that we will use in this section to describe the features. We will only work with terms which are not stop words and appear at least in five different tweets on the set T_c, given a company c.

For each of these terms, we have considered 18 features grouped in three classes:

1. **Collection-based features** (col_*): Terms that co-occur frequently with the (ambiguous) company name c, or terms written as hashtags should have more probability to be (positive/negative) keywords than others. These features combine information about the occurrence of the term in the collection: document frequency in the whole corpus, document frequency in

[15] We also tried with other machine learning methods, such as linear SVM and Naive Bayes, obtaining similar results; therefore we only report results on C4.5 for the sake of clarity.



(a) manual keywords

(b) 20 oracle keywords

Fig. 2. Fingerprints for the bootstrapping classification strategy when applying manual keywords (a) or 20 oracle keywords (b).

Table 4
Notation used to describe the features used to represent terms.

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.
$\mathrm{df}_t(T_c)$	document frequency of term t in the collection T_c.
$\mathrm{df}_{\mathrm{web}}(q)$	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 website 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, only the dataset to which the company belongs is used (either training or test).

the set of tweets for the company, how many times the term occurs as a twitter hashtag, and the average distance between the term and the company name in the tweets.

(a) **col_c_df**: Normalized document frequency in the collection of the tweets for the company T_c:

$$\frac{df_t(T_c)}{|T_c|} \qquad (1)$$

(b) **col_c_specificity**: Ratio of document frequency in the tweets for the company T_c over the document frequency in the whole corpus T:

$$\frac{df_t(T_c)}{df_t(T)} \qquad (2)$$

(c) **col_hashtag**: Number of occurrences of the term as a hashtag (e.g. `#jobs`, `#football`) in T_c.

(d) **col_c_prox_avg**, **col_c_prox_sd**, **col_c_prox_median**: Mean, standard deviation and median of the distance (number of terms) between the term and the company name in the tweets.

2. **Web-based features** (*web_∗*): These features are computed from information about the term in the all the Web (approximated by search counts), the website of the company, Wikipedia and the Open Directory Project (ODP).[16]

(a) **web**_bf c_bf assoc: Intuitively, a term which is close to the company name has more chances to be a keyword (either positive or negative) than more generic terms. This feature represents the association, according to the search counts, between the term t and a company name c.

$$\frac{\mathrm{df}_{\mathrm{web}}(t\,\mathrm{OR}\,c)/\mathrm{df}_{\mathrm{web}}(c)}{\mathrm{df}_{\mathrm{web}}(t)/M} \qquad (3)$$

(b) **web_c**_bf ngd: The Normalized Google Distance (Cilibrasi and Vitanyi, 2007) (applied to the Yahoo! search engine), which is a measure of semantic distance between two terms from the search counts. Then, for a term t and a company name c, the Google Normalized Distance is given by (4):

$$\frac{\max(\log(f(c)), \log(f(t))) - \log(f(t \text{ AND } c))}{M - \min(\log(f(t)), \log(f(c)))}$$
$$\text{where } f(x) = \mathrm{df}_{\mathrm{web}}(x) \qquad (4)$$

(c) **web_dom_df**: Frequent terms in the company website should be meaningful to characterize positive keywords. web_dom_df is the normalized document frequency of the term in the website of the company.

$$\frac{\mathrm{df}_{\mathrm{web}}(t \text{ AND } site : domain_c)}{\mathrm{df}_{\mathrm{web}}(\mathrm{site} : domain_c)} \qquad (5)$$

(d) **web_dom_assoc**: degree of association of the term with the website in comparison with the use of the term in the web. This feature is analogous to web_c_assoc, using the website domain instead of the company name c.

[16] Some companies in the Weps-3 collection have a Wikipedia page as reference page instead of the company website. In these cases, the feature web_dom_df (that is also the numerator in the feature web_dom_assoc) is computed as the presence of the term t in the Wikipedia page. Also, the query used to get the values of the features web_odp_occ and web_wiki_occ is the title of the Wikipedia page.

$$\frac{\mathrm{df}_{\mathrm{web}}(t \text{ AND } site : domain_c)/\mathrm{df}_{\mathrm{web}}(\text{site} : domain_c)}{\mathrm{df}_{\mathrm{web}}(t)/M} \tag{6}$$

(e) **web_odp_occ**: Number of occurrences of the term in all the items in *dmoz*($domain_c$). Each item is composed by an URL, a title, a description and the ODP category to which it belongs.

(f) **web_wiki_occ**: Number of occurrences of the term in the first 100 results in *wikipedia*($domain_c$). In order to filter pages returned by the API that could be unrelated to the company, only pages that contain the string $domain_c$ are considered.

3. **Features expanded with co-occurrence**: In order to avoid false zeros in web-based features, we expand some of the previous term features with the value obtained by the five most co-occurrent terms. Given a feature *f*, a new feature is computed as the Euclidean norm (7) of the vector with components $f_{t_i} * w(t, t_i)$ for the five most co-occurrent terms with *t* in the set of tweets T_c (8), where f_{t_i} is the web-based feature value *f* for the term t_i and $w(t,t_i)$ is the the grade of co-occurrence of each term (9):

$$cooc\_agg(t,f) = \sqrt{\sum_{i \in cooc_t} (f(t_i) * w(t_i))^2} \tag{7}$$

$$cooc_t = \text{set of the five terms which most co} - \text{occur with } t \tag{8}$$

$$w(t,t_i) = \frac{|\text{co} - \text{occurrences}_{T_c}(t,t_i)|}{|T_c|} \tag{9}$$

$$f(t_i) = \text{value of the feature } f \text{ for the term } t_i$$

This formula is applied to web_c_assoc, web_c_ngd, web_dom_df, web_dom_assoc, web_odp_occ and web_wiki_occ, resulting in the features enumerated below:

(a) **cooc_c_assoc** =*cooc_agg*(*t*, web_c_assoc)
(b) **cooc_c_ngd** =*cooc_agg*(*t*, web_c_ngd)
(c) **cooc_dom_df** =*cooc_agg*(*t*,web_dom_df)
(d) **cooc_dom_assoc** =*cooc_agg*(*t*, web_dom_assoc)
(e) **cooc_odp_occ** =*cooc_agg*(*t*,web_odp_occ)
(f) **cooc_wiki_occ** =*cooc_agg*(*t*,web_wiki_occ)

4.2. Feature analysis

The first step for the feature analysis is to develop a gold standard set of positive and negative keywords.

In order to get sufficient training data and to deal with possible miss-annotations in the corpus, we set a precision of 0.85 of a term in a related/unrelated set of tweets as a feasible threshold to annotate a term as a keyword. Those terms with precision lower than 0.85 in both classes are labeled as *skip* terms (10):

$$label(term) = \begin{cases} \text{positive} & \text{if } \frac{|related(T_c)|}{|T_c|} > 0.85 \\ \text{negative} & \text{if } \frac{|unrelated(T_c)|}{|T_c|} > 0.85 \\ \text{skip} & \text{otherwise} \end{cases} \tag{10}$$

where *related*(T_c) and *unrelated*(T_c) are respectively the set of the tweets annotated as related and unrelated in the collection T_c.

Labeling all suitable terms of the WePS-3 training dataset we end up with a total of 6410 terms, where 34% were labeled as positive keywords, 44% as negative keywords and the remaining 22% as skip. The test dataset, on the other hand, produces a total of 4653 candidate terms, where 33% were labeled as positive keywords, 40% as negative keywords and 27% as skip.

In order to study feature behavior, we calculate the distribution of each feature in the three classes: positive, negative and skip. We rely on box/whisker plots to show these distributions and differences or similarities between classes (see Fig. 3). Each box/whisker plot shows the distribution of values of a feature for the three classes. The bottom and top of the box are the 25th and 75th percentile (the Q_1 and Q_3 quartiles, respectively), and the band near the middle is the 50th percentile (the median, Q_2). The whiskers extend to the most extreme data point (1.5 times the length of the box away from the box: $-1.5 \cdot \mathrm{IQR}$ and $1.5 \cdot \mathrm{IQR}$, where $\mathrm{IQR} = |Q_3 - Q_1|$).

These plots help visualizing the range of values for each feature, as well as where most of the values lie, allowing for a qualitative analysis of the features.

We can see that features col_hashtag (Fig. 3(c)), web_odp_occ (Fig. 3(g)) and cooc_odp_occ (Fig. 3(i)) are not informative, because almost all of their values are zero. There are less than 1% of the terms in the test set that occur at least one time as hashtag. Also, less than 1% are terms that appear in descriptions and titles of ODP search results.

Features describing term - company distance seem to capture differences between keyword and skip terms: both negative and positive keywords, generally occur closer to the company name than skip terms. While positive and negative keywords share similar median and standard deviation (Fig. 3(d) and (f)) of proximity to the company name, average distance for positive keywords is slightly smaller than for negative keywords (Fig. 3(e)).

Features col_c_df, col_c_specificity, web_c_assoc, web_c_ngd and their expanded (by co-ocurrence) versions cooc_c_assoc and cooc_c_ngd were defined to discriminate filter keywords from skip terms. The most discriminative feature seems to be col_c_specificity (Fig. 3(b)).

On the other hand, features web_wiki_occ, web_dom_df, web_dom_assoc, cooc_dom_df,cooc_dom_assoc and cooc_wiki_occ were designed to distinguish between positive and negative filter keywords. At a first glance, positive and negative keywords have different distributions in all the features. Skip terms, on the other hand, tend to have distributions similar to those of positive keywords. The features cooc_dom_assoc (Fig. 3(r)) and cooc_wiki_occ (Fig. 3(j)) seem to be the best to discriminate positive keywords from negative and skip terms.

Remarkably, features expanded by co-occurrence seem to be more informative than the original features, which tend to concentrate on low values (the median is near zero). When expanding the original values by co-occurrence, positive terms receive higher values more consistently.

In order to quantitatively evaluate the quality of features, we compute the Mann–Whitney U test (Mann and Whitney, 1947), which is a non-parametric test used in statistical feature selection when a normal distribution of the features cannot be assumed. The p-value could be used to rank the features, since the smaller the value of the p-value, the more informative the feature is (Guyon et al., 2006).

Table 5 shows the p-value and the rank of each feature for the U test. The most remarkable aspect of this table is that – in agreement with the boxplot analysis – the *col_c_specificity* feature discriminates between filter keywords and skip terms better than other features. In addition, the feature *cooc_dom_assoc*, which measures the association between the term and the company website, is the best feature to discriminate between positive and negative filter keywords. These results confirm our assumptions that salient terms in the set of tweets of the company tend to be discriminative and salient terms associated with the company in tweets are also associated with the company website.

Although the features analyzed above are signals that help differentiating between positive, negative and skip terms, it seems that the vocabulary that characterizes a company in microblog streams is different from the vocabulary associated to the company in its website, in ODP entries or in Wikipedia.

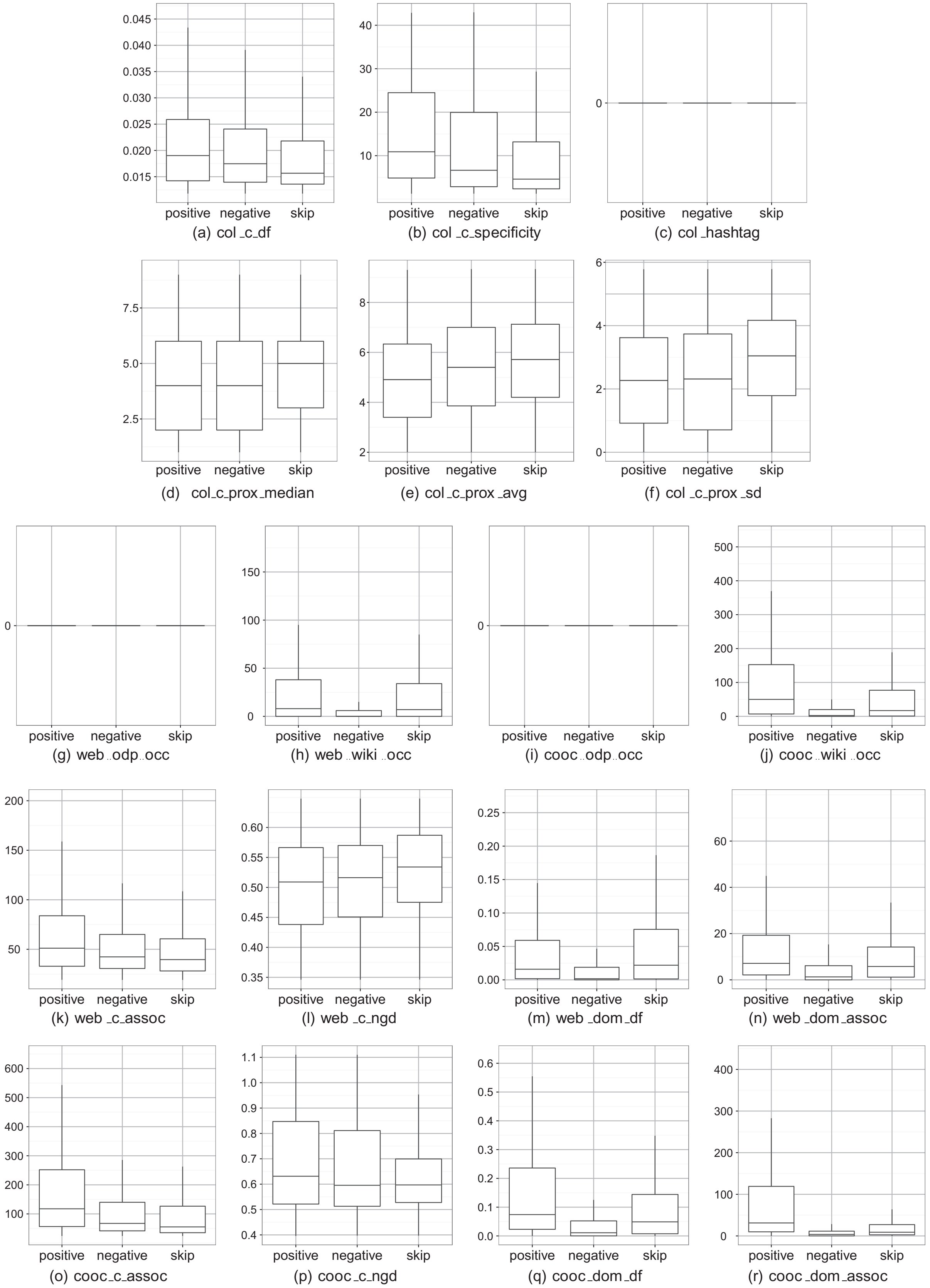


Fig. 3. Box-plots representing the distribution of each of the features in the positive, negative and skip classes. The bottom and top of the box are the Q_1 and Q_3 quartiles, respectively, and the band near the middle is the Q_2 quartile –i.e., the median. The whiskers extend to the most extreme data point (1.5 times the length of the box away from the box: $-1.5 \cdot \text{IQR}$ and $1.5 \cdot \text{IQR}$, where $\text{IQR} = |Q_3 - Q_1|$).

4.3. Keyword discovery

The features described above have been combined in three different ways to automatically discover filter keywords. The first one, (*machine learning-all features*), consists of training a positive–negative-skip classifier over the training corpus in WePS-3 by using all the features. We combine two classifiers: positive versus others and negative versus others, using the confidence thresholds learned by the classifiers. Terms which are simultaneously under/over both thresholds are tagged as skip terms.

The second approach, (*heuristic*), is inspired by the analysis of the signal provided by each of the features (in the previous section). It is a simple heuristic which looks at only the best two features according to the Mann–Whitney U test: first, we define a threshold to remove skip terms according to the specificity w.r.t. the collection of the tweets for the company (*col_c_specificity* feature). Then we state, for the feature that measures association with the website (*cooc_dom_assoc* feature) a lower bound to capture positive filter keywords and an upper bound to capture negative filter keywords. These three thresholds have been manually optimized using the training data set.

Finally, we also explored a third option: we apply machine learning using only the best two features instead of the whole feature set. We will refer hereafter to this method as *machine learning-2features*.

We have experimented with several machine learning methods using Rapidminer (Mierswa et al., 2006): Neural Nets, C4.5 and CART Decision Trees, Linea Support Vector MAchines (SVM) and Naive Bayes. All methods have been used with “out-of-the-box” parameters.

All the terms labeled over the WePS-3 training dataset were used to train the models. In the same way, terms extracted from the test dataset were used as test set. Table 6 shows the values of the Area Under the ROC Curve (AUC) of each of the binary classifiers evaluated. AUC is an appropriate metric to measure the quality of binary classification models independently of the confidence threshold (Fawcett, 2006).

We analyzed three different subsets of features to represent the terms: (i) using all but the six features expanded by co-occurrence, (ii) using only the best two features (those used by the *heuristic* and *machine learning-2features* classifiers), and (iii) using all the features.

The results obtained are similar for all models, except for C4.5 and SVM that in several cases do not provide any useful information for classification (AUC = 0.5). Keeping out the “expanded by co-occurrence“ features, the performance is, in general, lower for all the algorithms. This corroborates the results of our previous feature analysis.

In the following experiments, we focus on the Neural Net algorithm to train both (positive versus others and negative versus other) classifiers, because it is consistently the best performing algorithm according to the AUC measure.

For each of the feature combinations described at the beginning of this section (*machine learning-all features*, *heuristic* and *machine learning-best 2 features*), below we analyze the obtained results. The methods were trained using terms from the WePS-3 training dataset and evaluated with the WePS-3 test set.

Table 7 shows the confusion matrix obtained for the *machine learning-all features* method. The precision for the positive and negative classes is 62% and 56%, respectively, while recall is 52% and 72%. In order to obtain a few representative keywords, this recall levels are good enough; but the precision may compromise the final accuracy of the filtering process.

Table 8 shows the confusion matrix for the *heuristic* method, that only uses the col_c_specificity and cooc_dom_assoc features. This method is more precision-oriented than *machine learning - all features*: precision values of positive and negative class are higher (68% and 75%), but recall is significantly lower (26% and 19%).

Finally, Table 9 shows the contingency matrix for the *machine learning-2features* method, that represents terms with the features col_c_specificity and cooc_dom_assoc and uses the Neural Net machine learning algorithm to build the model.

As expected, its performance lies between *machine learning-all features* and *heuristic* methods, with a precision higher than the former (65% and 68%) and a recall higher than the latter (29% and 21%).

These results indicate that automatic detection of keywords is plausible and challenging at the same time. Which of the three approaches is better for our problem depends on their performance on the final task: tweet classification. In the next section we explore how to use these filter keywords to classify tweets.

5. Automatic tweet classification

After detecting the filter keywords automatically, we directly classify the subset of tweets that contain only negative or only positive keywords. Then, the *bootstrapping* strategy described in Section 3.2 is used to complete the task.

Table 5
U test p-value and ranking position of the features, comparing filter keywords (both positive and negative) with skip terms and comparing positive with negative filter keywords.

	Filter keywords vs. Skip terms		Positive vs Negative filter keywords	
	p-value	Rank	p-value	Rank
col_c_df	2.11e−19	7	4.55e−31	8
col_c_specificity	8.18e−50	**1**	4.40e−22	9
col_hashtag	1.49e−02	15	5.40e−04	12
col_c_prox_sd	1.87e−33	2	4.20e−02	15
col_c_prox_avg	2.03e−20	6	4.18e−02	14
col_c_prox_median	5.79e−14	8	1.62e−01	16
web_c_assoc	4.76e−06	12	8.39e−19	10
web_dom_df	6.67e−30	4	5.33e−92	6
web_dom_assoc	7.14e−14	9	7.01e−138	4
web_c_ngd	5.12e−12	10	3.38e−08	11
web_odp_occ	1.96e−01	16	3.63e−01	18
web_wiki_occ	1.68e−20	5	4.86e−115	5
cooc_c_assoc	2.91e−30	3	2.04e−54	7
cooc_dom_df	1.53e−05	13	1.42e−189	3
cooc_dom_assoc	8.35e−01	18	**7.27e−233**	**1**
cooc_c_ngd	3.54e−07	11	2.41e−01	17
cooc_odp_occ	3.15e−01	17	3.92e−03	13
cooc_wiki_occ	2.36e−03	14	2.13e−211	2

Table 6
Area Under the ROC Curve (AUC) values of the five classification models and the three feature sets used to classify positives and negatives keywords (best AUC values in boldface).

Machine learning algorithm	Not expanded by co-occurrence features		2 Best features		All features	
	pos	neg	pos	neg	pos	neg
Neural Net	**0.68**	**0.67**	**0.73**	**0.72**	**0.75**	**0.73**
CART Dec. Trees	0.58	0.61	0.72	0.71	0.63	0.64
Linear SVM	0.50	0.50	0.73	0.71	0.50	0.50
Naïve Bayes	0.64	0.64	0.71	0.71	0.72	0.72
C4.5 Dec.Trees	0.50	0.61	0.50	0.50	0.59	0.66

Table 7
Confusion matrix for the *machine learning-all features* classifier (precision and recall for Positive and Negative classes in boldface).

		Actual class			Class precision (%)
		Positive	Negative	Skip	
Predicted class	Positive	790	190	304	**62**
	negative	483	1334	583	**56**
	skip	242	330	375	40
	Class Recall (%)	**52**	**72**	30	

Table 8
Confusion matrix for the *heuristic* classifier (precision and recall for Positive and Negative classes in boldface).

		Actual class			Class precision (%)
		Positive	Negative	Skip	
Predicted class	Positive	391	60	122	**68**
	Negative	23	352	94	**75**
	Skip	1102	1453	1056	29
	Class recall (%)	**26**	**19**	83.02	

Table 9
Confusion matrix for the *machine learning-2features* classifier (precision and recall for Positive and Negative classes in boldface).

		Actual class			Class precision (%)
		Positive	Negative	Skip	
Predicted class	Positive	438	102	139	**65**
	Negative	85	399	101	**68**
	Skip	993	1364	1032	30
	Class recall (%)	**29**	**21**	81	

Ii is also interesting to compare the *bootstrapping* strategy with the naïve *winner-takes-all* baseline –that directly classifies all the tweets as related or unrelated depending on which is the dominant class in the seed of tweets– and the *winner-takes-remainder* strategy, which consists of applying the winner-takes-all strategy only to those tweets that were not covered by some of the filter keywords.

Fig. 4 shows the fingerprint of each of the combinations tested and Table 10 shows the results. The best automatic method, which combines (`machine learning-all features` to discover keywords and bootstrapping with the tweets annotated using that keywords) gives an accuracy of 0.73, which is higher than using manual keywords from the Web (0.67) and is close to the best automatic result reported in the WePS-3 competition (0.75). In addition, the bootstrapping process almost doubles the coverage (from 58% to 100%) with only 2.7% of accuracy loss.

In general, the more tweets are covered by filter keywords (seed coverage), the lower is the loss in accuracy: the `heuristic` keyword selection covers 27% of tweets with .79 accuracy and achieves a .71 accuracy with the bootstrapping process, while `machine learning-2 best features` covers 39% of the tweets with 0.78 accuracy and finishes with 0.72 accuracy. Remarkably, the bootstrapping process outperforms the winner-takes-all and winner-takes-remainder baselines in all the cases.

An interesting question is how our approach - which does not use training data from the companies in the test set - compares with a truly supervised counterpart (i.e. one which uses the same machine learning algorithm and the same bag-of-word features, but uses perfect training material – taken from the test set – for each of the companies). To this aim, we carried a 10-fold validation on the test set of the machine learning algorithm. This supervised upper bound achieves 0.85 accuracy, that is only 14% higher than our unsupervised algorithm (0.73). In Fig. 5 we can see the fingerprint representation of the supervised upper bound. This fingerprint is similar to the 20 oracle keyword's fingerprint shown in Section 3.2.

Discovery of filter keywords has proved to be challenging using signals from the Web: the accuracy of the resulting seed set ranges between 0.75 and 0.79, with a potentially useful coverage (58%) in the case of the *machine learning - all features*. Overall, this result reinforces the conclusion that the characterization of companies in Twitter, in terms of vocabulary, is probably different from the characterization that can be derived from the Web.

6. Discussion

6.1. How much of the problem is solved?

In order to shed light on the trade-off between quality and quantity of filter keywords, here we analyze the relation between accuracy and coverage of the tweets classified by considering different sets of filter keywords. Fig. 6 shows the coverage/accuracy curves for oracle, manual and automatic filter keywords.

Curves were generated as follows:

Oracle keywords. At step n, we consider the n^{th} positive/negative oracle keywords that maximizes accuracy and - in case of ties - coverage of tweets (i.e., in the case of two keywords hav-

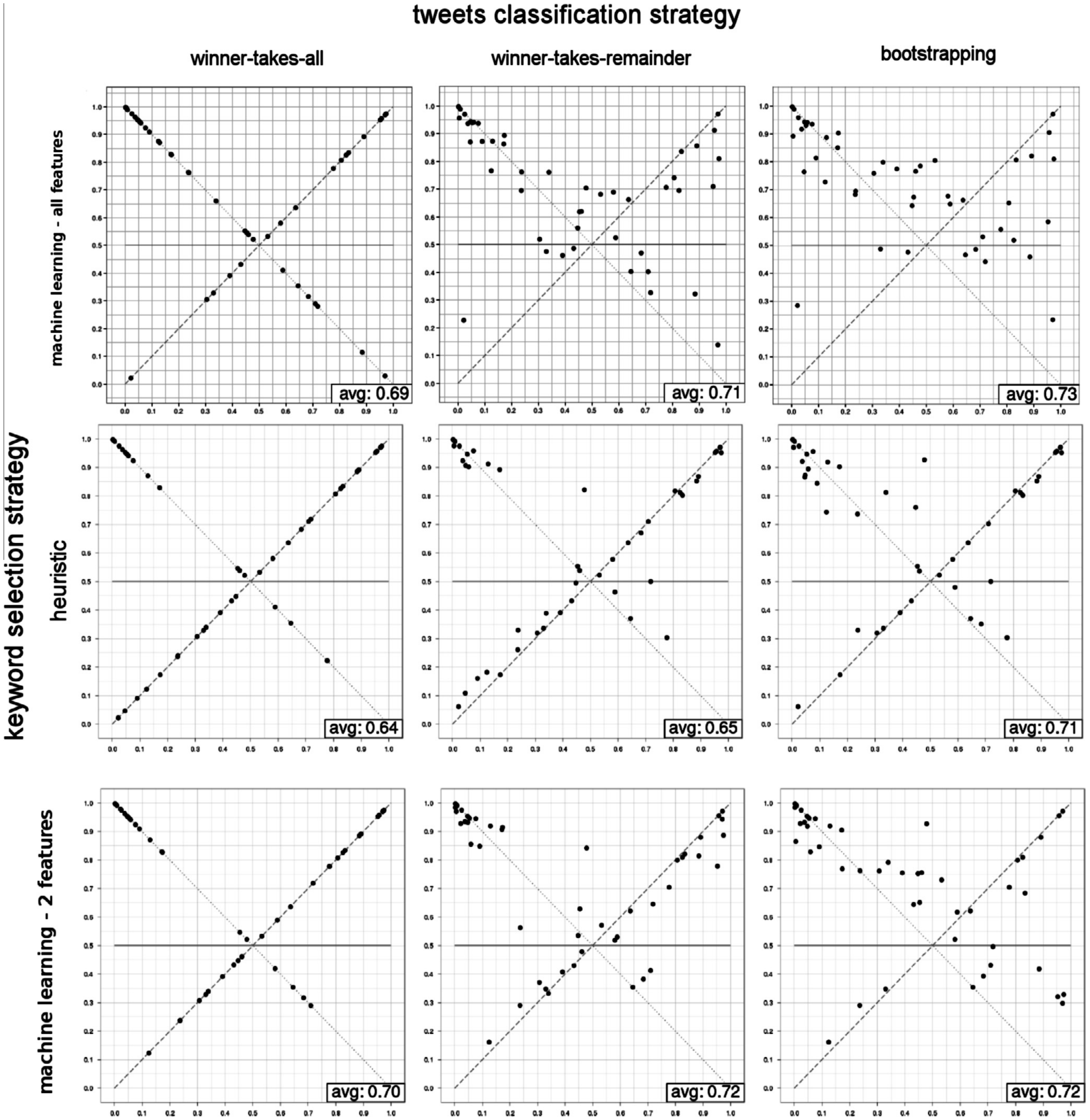


Fig. 4. Fingerprints for each of the keyword selection strategies combined with each of the different tweet classification strategies.

Table 10
Results for automatic keyword detection strategies (wta = winner-takes-all, wtr = winner-takes-remainder). Statistical significance w.r.t. the `ml-all features` selection strategy was computed using two-tailed Student's t-test. Significant differences are indicated using ▲ (or ▼) for $\alpha = 0.01$ and △(or ▽) for $\alpha = 0.05$.

Keyword selection strategy	Seed set		Overall accuracy		
	Coverage (%)	acc.	wta	wtr	Bootstrapping
20 oracle keywords	$53^{\blacktriangle}$	$1.00^{\blacktriangle}$	$0.80^{\triangle}$	$0.85^{\blacktriangle}$	$0.87^{\blacktriangle}$
manual keywords	$15^{\blacktriangledown}$	0.86	0.61	0.63	0.67
supervised bootstr.					$0.85^{\blacktriangle}$
m. learning – all feat.	58	0.75	0.69	0.71	**0.73**
heuristic	$27^{\blacktriangledown}$	0.79	0.64	0.65	0.71
m. learning – 2 feat.	$39^{\blacktriangledown}$	0.78	0.70	0.72	0.72

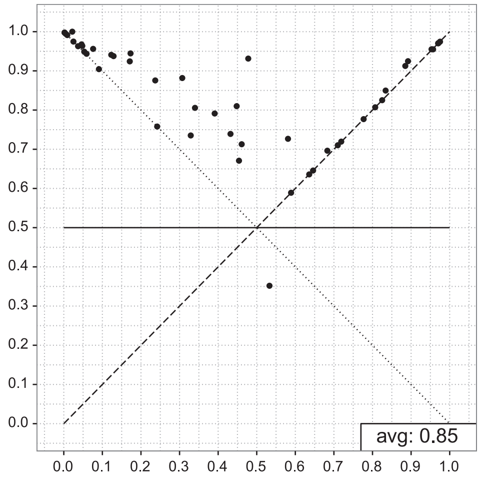


Fig. 5. Fingerprint for the bootstrapping upper bound (10-fold cross-validation).

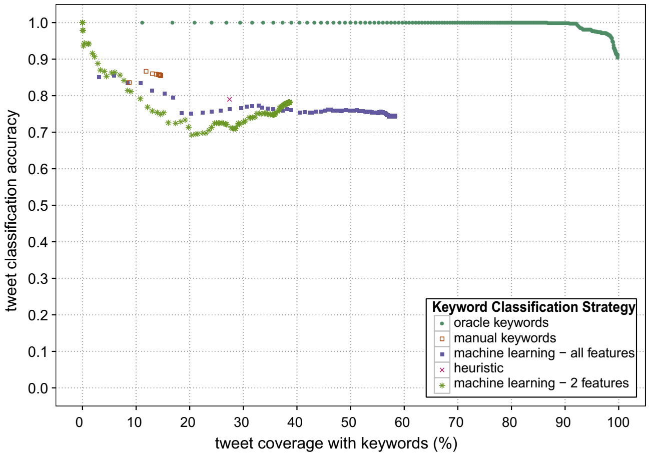


Fig. 6. Coverage/accuracy curves for oracle, manual and automatic filter keywords.

ing same accuracy, the one that covers more tweets is considered first).

Manual keywords. At each step n we consider the n^{th} positive/negative manual keywords that maximize coverage of tweets.

Machine learning – all features. The set of terms considered in this analysis are those that were classified as positive or negative by using the confidence thresholds learned by the two classifiers in the method "machine learning - all features". Skip terms are (i) those classified as skip by both binary classifiers; (ii) those classified simultaneously as positive and negative keywords. Then, we use the maximum of the confidence scores returned by the two classifiers (i.e., max (conf (positive),conf (negative))) to sort the keywords. The keyword with highest confidence score is added at each step. The point in the curve with highest coverage corresponds to the classifier used in the experiments explained in Section 4.

Machine learning-2 features. This curve is generated by using the two classifiers learned in "machine learning - 2 features", similarly to the curve generated for "machine learning - all features".

Heuristic. Since this classifier consists of manually defining thresholds using the training set, it doesn't provide any confidence score for the test cases. Hence, in the graphic it is represented as a single point (×).

The curve for Oracle keywords provides a statistical upper bound of how many tweets can be directly covered using filter keywords. Considering the best 100 oracle keywords for each test case/company name, it is possible to directly tag 85% of the tweets with 0.99 accuracy. On the other hand, a more realistic upper bound is given by manual keywords. Here, we can observe how the accuracy remains stable around 0.85, while the coverage grows from 10% to 15% approx. In the best possible case, with more keywords the curve would continue as the line $y = 0.85$. Note that manual keywords have been annotated by inspecting representative Web pages (from Google search results) rather than inspecting tweets.

Therefore, an automatic keyword classifier cannot achieve an accuracy above 0.85. Considering this, our automatic approaches establish a strong lower bound of 0.7 accuracy. In conclusion, it seems that a filter keyword classifier should have reach an accuracy between 0.7 and 0.85 to be competitive.

6.2. Comparing systems with different metrics

We have seen that related/unrelated tweets are not balanced in most of the test cases in WePS-3, and the proportion does not follow a normal distribution (extreme values seem to be as plausible as values around the mean). Because of this, accuracy may be not sufficient to understand the quality of systems, and that's why we have complemented it with the fingerprint representation (Spina et al., 2011). In this section, we evaluate (and compare) results with the most popular alternative evaluation metrics found in the literature.

Considering the confusion matrix given by each system, where TP = true related tweets, FP = false related tweets, TN = true unrelated tweets, and FN = false unrelated tweets, we compute the following metrics, in addition to accuracy:

Normalized Utility. Utility has been used to evaluate document filtering tasks in TREC (Hull et al., 1998, 1999) and is commonly used assigning a relative α weight between true positives and false positives:

$$u(S,T) = \alpha \cdot TP - FP$$

As in the TREC-8 filtering task (Hull and Robertson, 1999), here Utility is normalized by means of the following scaling function:

$$u_s^*(S,T) = \frac{max(u(S,T),U(s)) - U(s)}{MaxU(T) - U(s)}$$

where $u(S,T)$ is the original utility of system output S for topic T, $U(s)$ is the utility of retrieving s non-relevant documents, and $MaxU(T) = \alpha \cdot (TP + FN)$ is the maximum possible utility score for topic T. In this paper, we set $\alpha = 2$ and $U(s) = -25$.

lam%. $lam\%$ (logistic average misclassification percentage) has been used in TREC to evaluate the problem of spam detection (Cormack and Lynam, 2005). It was defined as the geometric mean of the odds of $hm\%$ (ham misclassification percentage) and $sm\%$ (spam misclassification percentage). More precisely, $lam\%$ is defined as

$$lam\% = logit^{-1}\left(\frac{logit(hm\%) + logit(sm\%)}{2}\right)$$

where

$$hm\% = \frac{FN}{FN + TP} \quad sm\% = \frac{FP}{FP + TN}$$

$$logit(x) = log\left(\frac{x}{1-x}\right) \quad logit^{-1}(x) = \frac{e^x}{1+e^x}$$

Note that lam% is an error-based metric – i.e., maximum scores represent minimum quality.
One remarkable property of this metric is that, when a system has a non-informative behavior –that is, it classifies the documents randomly, or classifies all documents to a single class– lam% score is around 0.5.

Reliability & **Sensitivity.** Reliability and Sensitivity (Amigó et al., 2012) have been recently proposed as two complementary measures to evaluate document organization tasks involving classification, clustering and ranking. It was the official metric used in RepLab held at CLEF 2012 (Amigó et al., 2012). When evaluating a binary classification task, Reliability corresponds to the product of the precision of the classes, and Sensitivity to the product of the recall of both classes:

$$R = \frac{TP}{TP + FP} \cdot \frac{TN}{TN + FN} \qquad S = \frac{TP}{TP + FN} \cdot \frac{TN}{TN + FP}$$

and

$$F_1(R, S) = 2 \cdot \frac{R \cdot S}{R + S}$$

As *lam%*, Reliability & Sensitivity also penalizes systems that do not provide any useful information. Contrary to *lam%*, that assigns a low but not the minimum score, Reliability & Sensitivity gives zero –the minimum score– to these systems.

F_1 measure. The most standard combination of Precision and Recall is F_1, or balanced F measure. Here we focus on the "related" class, where

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

and

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table 11 reports the results for the baselines, the WePS-3 systems and our proposed systems for the metrics described above. All metrics were macro-averaged by topics, and undefined scores were considered as zero values.

Results show that, according to Reliability & Sensitivity, our best automatic system `ml-all features + bootstrapping` achieves the same score as the WePS-3 LSIR semi-automatic system (0.27) – which is the best result at the competition and involves manual processing – and outperforms the best automatic system in WePS-3 (ITC-UT = 0.20), with a 35% of relative improvement. In terms of *lam%*, the SINAI system achieves the best automatic score of 0.35, followed by ITC-UT & `ml-all features + bootstrapping` that reaches 0.37 *lam%*. Note that *lam%* and R& S penalize non-informative/baseline-like behaviors. Because of this, the `winner-takes-all` systems and the "all (un) related" baselines get the worst scores in these metrics.

According to utility, ITC-UT is still the best automatic system (0.52). Our best runs are between 0.47 and 0.49, being `ml-2 features + bootstrapping` the best of them. Finally, F_1 rewards systems that tend to return all tweets as related. Indeed, the best score given by an automatic system is achieved by the "all_related" baseline, that has perfect recall and enough precision to get the highest score.

There are a number of empirical observations that can be made on this comparison of metrics:

Table 11
Results for proposed systems, WePS-3 systems and baselines compared with different evaluation metrics. Best automatic runs are in boldface. (ml = machine learning, wta = winner-takes-all, wtr = winner-takes-remainder). Statistical significance w.r.t. the `ml-all feat.+bootstrapping` run was computed using two-tailed Student's t-test. Significant differences are indicated using ▲ (or ▼) for $\alpha = 0.01$ and △(or ▽) for $\alpha = 0.05$.

System	Accuracy	Utility	lam%	**F$_1$**(R,S)	**F$_1$**
Gold standard	1.00$^{\blacktriangle}$	1.00$^{\blacktriangle}$	0.00$^{\blacktriangle}$	1.00$^{\blacktriangle}$	1.00$^{\blacktriangle}$
supervised bootstr.	0.85$^{\blacktriangle}$	0.69$^{\blacktriangle}$	0.28$^{\blacktriangle}$	0.30$^{\blacktriangle}$	0.62$^{\blacktriangle}$
WePS-3: LSIR (manual)	0.83$^{\blacktriangle}$	0.66$^{\blacktriangle}$	0.28$^{\blacktriangle}$	0.27	0.62$^{\blacktriangle}$
`ml-all feat.+bootstr.`	0.73	0.47	0.37	**0.27**	0.49
`ml-2 feat.+wtr`	0.72	0.49	0.43	0.16$^{\triangledown}$	0.49
`ml-2 feat.+bootstr.`	0.72	0.47	0.43	0.17$^{\triangledown}$	0.50
`heuristic + bootstr.`	0.71	0.45	0.42	0.11$^{\blacktriangledown}$	0.46
`ml-all feat.+wtr`	0.71	0.44	0.39$^{\blacktriangledown}$	0.21$^{\blacktriangledown}$	0.43$^{\blacktriangledown}$
`ml-2 feat.+wta`	0.70	0.48	0.50$^{\blacktriangledown}$	0.00$^{\blacktriangledown}$	0.39$^{\triangledown}$
`ml-all feat.+wta`	0.69$^{\triangledown}$	0.40$^{\triangledown}$	0.50$^{\blacktriangledown}$	0.00$^{\blacktriangledown}$	0.27$^{\blacktriangledown}$
`heuristic + wtr`	0.65$^{\blacktriangledown}$	0.46	0.42	0.10$^{\blacktriangledown}$	0.46
`heuristic + wta`	0.64$^{\blacktriangledown}$	0.44	0.50$^{\blacktriangledown}$	0.00$^{\blacktriangledown}$	0.39$^{\triangledown}$
WePS-3: ITC-UT	**0.75**	**0.52**	0.37	0.20	0.49
WePS-3: SINAI	0.64$^{\triangledown}$	0.38$^{\triangledown}$	**0.35**	0.12$^{\blacktriangledown}$	0.30$^{\blacktriangledown}$
WePS-3: UvA	0.58$^{\blacktriangledown}$	0.22$^{\blacktriangledown}$	0.46$^{\blacktriangledown}$	0.17$^{\blacktriangledown}$	0.36$^{\blacktriangledown}$
WePS-3: KALMAR	0.47$^{\blacktriangledown}$	0.35$^{\blacktriangledown}$	0.43	0.16$^{\triangledown}$	0.48
baseline: all unrelated	0.57$^{\blacktriangledown}$	0.20$^{\blacktriangledown}$	0.50$^{\blacktriangledown}$	0.00$^{\blacktriangledown}$	0.00$^{\blacktriangledown}$
baseline: random	0.49$^{\blacktriangledown}$	0.20$^{\blacktriangledown}$	0.49$^{\blacktriangledown}$	0.16$^{\blacktriangledown}$	0.37$^{\blacktriangledown}$
baseline: all related	0.43$^{\blacktriangledown}$	0.40	0.50$^{\blacktriangledown}$	0.00$^{\blacktriangledown}$	**0.52**

- In general, high $F_1(R,S)$ implies high accuracy, but not vice versa: $F_1(R,S)$ is a stricter metric, at least in this dataset.
- Metrics such as *lam%* and $F_1(R,S)$ are suitable to identify baseline-like behaviors, while F_1 is not.
- ITC-UT and `ml-features + bootstrapping` perform consistently well across metrics.
- Different metrics illustrate different aspects of the behavior of systems: If we need to penalize non-informative behavior, we should look at results with *lam%* or $F_1(R,S)$. Accuracy and utility directly show misclassification errors, but are sensitive to collections where class skews are variable over different test cases, such as our dataset.

6.3. Web vs. Twitter

In our experiments, we have found two results indicating that the vocabulary that characterizes a company on Twitter substantially differs from the the vocabulary associated to the company on the Web:

- **Low recall of manual keywords.** As we saw in Section 3.1, manually selecting around 10 salient terms from Web search results retrieved using the company name and its representative URL only covers 15% of the tweets.
- **Web-based features are useful but inconclusive.** Analyzing the features (see Section 4.2), we found that web-based features that may discriminate positive from negative keywords tend to receive low values. Moreover, the low quality (⩽0.75 AUC) of the automatic classification of filter keywords indicates that building an accurate classifier with features extracted from the Web is challenging to say the least (see Section 4.3).

In order to reach a better understanding of the problem, we have explored the association between the best 10 oracle keywords for each tweet stream and its occurrences in both the company's homepage and its Wikipedia article.[17] The terms from each page have been extracted using the `lynx -dump` *url* Linux command.

[17] We manually extended the input data of each organization on the WePS-3 dataset with its Wikipedia page (or its homepage in the cases which the Wikipedia page is provided as the representative page).

Table 12
Percentage of the 10 best *oracle* keywords extracted from the tweet stream covered by the company's homepage, its Wikipedia article and both.

Filter keywords	Homepage (%)	Wikipedia (%)	Both (%)
Related *oracle* keywords	36	68	33
Unrelated *oracle* keywords	9	19	6

Table 12 shows the average percentage of the best 10 oracle keywords that occur on the company's homepage, on the Wikipedia page, and both.

Overall, the only substantial overlap is for positive keywords in Wikipedia, indicating that representative Web pages are not the ideal place to look for effective filter keywords in Twitter.

Note that the overlap of related oracle keywords with the company's Wikipedia page roughly doubles the overlap with its homepage. The same thing happens with unrelated keywords: almost 20% on Wikipedia and almost 10% on the homepage. The percentage of oracle keywords that occur both in the homepage and in the Wikipedia article is similar to the homepage alone, indicating that Wikipedia basically extends the keywords already present in the homepage.

In summary, exploring the nature of filter keywords leads us to the conclusion that the vocabulary characterizing a company in Twitter is substantially different from the vocabulary associated to the company in its homepage, in Wikipedia, and apparently in the Web at large. These findings are in line with the "vocabulary gap" that has been shown between Twitter and other Web sources such as Wikipedia or news comments (Tsagkias et al., 2011). One way of alleviating this problem is using co-occurrence expansion of web-based features, which allows to better recognize automatically filter keywords. While the company's Wikipedia article seems to have more coverage of (perfect) filter keywords than the company's homepage, further investigation is needed on how to automatically infer the company's Wikipedia page from its homepage URL in order to extract additional keyword features from it.

7. Conclusion

In this paper we tackled the problem of company name disambiguation in Twitter, defined in WePS-3 as a binary classification problem. We have studied the use of filter keywords: expressions that, if present in a tweet, indicate a high probability that it is related/unrelated to the company.

In our experiments, automatically discovered filter keywords are able to classify a subset of 30%-60% tweets with an accuracy range of .75–.79.

We defined features that characterize terms in the Twitter dataset, the company's website, ODP, Wikipedia and the searchable Web. We found that (i) term specificity in the tweet stream of each company name is a feature that discriminates between filter keywords and skip terms and (ii) the association between the term and the company website is useful to differentiate positive vs. negative filter keywords, specially when it is averaged by considering its most co-occurrent terms. Tweets classified by these filter keywords can be used to feed a supervised machine learning process to obtain a complete classification of all tweets for an overall accuracy of 0.73. In comparison, a 10-fold validation of the same machine learning algorithm provides an accuracy of 0.85, i.e., our unsupervised algorithm has a 14% loss with respect to its supervised counterpart.

We also found that, in average, the best five optimal keywords can directly classify around 30% of the tweets. Nevertheless, keywords defined by a human by inspecting web search results relevant to the company name only cover 15% of the tweets and accuracy drops to 0.86.

Exploring the nature of filter keywords also led us to the conclusion that that the there is a gap between the vocabulary characterizing a company in Twitter and the vocabulary associated to the company in its homepage, in Wikipedia, and apparently in the Web at large.

Note that all our experimentation is based on the WePS-3 dataset, which is an English-only dataset. As immediate future work, we plan to explore cross-lingual strategies to deal with multilingual data and so be able to test our approach on the RepLab 2012 dataset, which is the other test collection that fits our problem.

References

Agirre, E., & Edmonds, P. (2006). *Word sense disambiguation: Algorithms and applications*. Springer.

Al-Kamha, R. & Embley, D. (2004). Grouping search-engine returned citations for person-name queries. In *Proceedings of the 6th annual ACM international workshop on Web information and data management* (pp. 96–103).

Amigó, E., Artiles, J., Gonzalo, J., Spina, D., Liu, B., & Corujo, A. (2010). WePS-3 evaluation campaign: Overview of the online reputation management task. In *CLEF 2010 labs and workshops notebook papers*.

Amigó, E., Corujo, A., Gonzalo, J., Meij, E., & de Rijke, M. (2012). Overview of RepLab 2012: Evaluating online reputation management systems. In *CLEF 2012 labs and workshop notebook papers*.

Amigó, E., Gonzalo, J., & Verdejo, F. (2012). *Reliability and sensitivity: Generic evaluation measures for document organization tasks*. Tech. rep., UNED.

Artiles, J. (2009). *Web people search*. Ph.D. thesis, UNED University.

Artiles, J., Borthwick, A., Gonzalo, J., Sekine, S., & Amigó, E. (2010). *Weps-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks*.

Artiles, J., Gonzalo, J., & Amigó, E. (2009a). The impact of query refinement in the web people search task. In *Proceedings of the ACL-IJCNLP 2009 conference short papers. Association for Computational Linguistics* (pp. 361–364).

Artiles, J., Gonzalo, J., & Sekine, S. (2007). The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of Semeval*.

Artiles, J., Gonzalo, J., & Sekine, S. (2009b). Weps 2 evaluation campaign: Overview of the web people search clustering task. In *18th WWW conference 2nd web people search evaluation workshop (WePS 2009)*.

Bagga, A., & Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics* (Vol. 1, pp. 79–85).

Boyd, D., Golder, S., & Lotan, G. (2010). Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of the 2010 43rd Hawaii international conference on system sciences, HICSS'10* (pp. 1–10). Washington, DC, USA: IEEE Computer Society.

Bunescu, R., & Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*. (Vol. 6, pp. 9–16).

Cha, M., Haddadi, H., Benevenuto, F., & Gummadi, K. P. (2010). Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the 4th international AAAI conference on weblogs and social media (ICWSM)*, Washington DC, USA.

Cheng, Z., Caverlee, J., & Lee, K. (2010). You are where you tweet: A content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 759–768).

Cilibrasi, R. L., & Vitanyi, P. M. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*.

Comm, J., & Robbins, A. (2009). *Twitter power: How to dominate your market one tweet at a time*. John Wiley & Sons Inc.

Cormack, G., & Lynam, T. (2005). Trec 2005 spam track overview. In *Proceedings of the 14th text retrieval conference (TREC 2005)*.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL* (Vol. 2007, pp. 708–716).

Dellarocas, C., Awad, N., & Zhang, X. (2004). Exploring the value of online reviews to organizations: Implications for revenue forecasting and planning. In *Proceedings of the international conference on information systems*.

Dredze, M., McNamee, P., Rao, D., Gerber, A., & Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 277–285).

Edmonds, P., & Cotton, S. (2001). Senseval-2: Overview. In *Proceedings of the 2nd international workshop on evaluating word sense disambiguation systems SENSEVAL-2* (pp. 1–6).

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Ferragina, P., & Scaiella, U. (2010). Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 1625–1628).

Frank, E., Paynter, G., Witten, I., Gutwin, C., & Nevill-Manning, C. (1999). Domain-specific keyphrase extraction. In *International joint conference on artificial intelligence* (Vol. 16, pp. 668–673).

García-Cumbreras, M.A., García-Vega, M., Martínez-Santiago, F., & Peréa-Ortega, J. M. (2010). SINAI at WePS-3: Online reputation management. In *CLEF 2010 labs and workshops notebook papers*.

Gentile, A., Zhang, Z., Xia, L., & Iria, J. (2010). Semantic relatedness approach for named entity disambiguation. In *Digital libraries* (pp. 137–148).

Gooi, C., & Allan, J. (2004). Cross-document coreference on a large scale corpus. In *Proceedings of HLT/NAACL* (Vol. 4).

Gouws, S., Metzler, D., Cai, C., Hovy, E., & Marina del Rey, C. (2011). *Contextual bearing on linguistic variation in social media*.

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 16th conference on computational linguistics* (Vol. 1, pp. 466–471).

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. (2006). *Feature extraction: Foundations and applications*. Springer Verlag, pp. 67–78, [Ch. 2.2].

Hoffman, T. (2008). Online reputation management is hot?but is it ethical?. *Computerworld*, February.

Hull, D., & Robertson, S. (1999). The trec-8 filtering track final report. In *Proceeding of eighth text retrieval conference (TREC-8)*.

Hull, D. et al. (1998). The trec-7 filtering track: Description and analysis. *NIST Special Publication SP* (pp. 45–68).

Hurlock, J., & Wilson, M.L. (2011). Searching twitter: Separating the tweet from the chaff. In *Proceedings of ICWSM 2011, the 5th international AAAI conference on weblogs and social media*, AAAI.

Jansen, B., Zhang, M., Sobel, K., & 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.

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on web mining and social network analysis* (pp. 56–65).

Ji, H., & Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (Vol. 1, pp. 1148–1158).

Ji, H., Grishman, R., Dang, H., Griffitt, K., & Ellis, J. (2010). Overview of the tac 2010 knowledge base population track. In *TAC (Text analysis co nference) 2010 workshop*.

Kalashnikov, D., Mehrotra, S., Chen, Z., Nuray-Turan, R., & Ashish, N. (2007). Disambiguation algorithm for people search on the web. In *IEEE 23rd international conference on data engineering, ICDE 2007* (pp. 1258–1260).

Kalmar, P. (2010). Bootstrapping websites for classification of organization names on twitter. In *CLEF 2010 Labs and Workshops Notebook Papers*.

Kaufmann, M., & Kalita, J. (2010). Syntactic normalization of twitter messages. In *International conference on natural language processing*, Kharagpur, India.

Kim, S., Medelyan, O., Kan, M., & Baldwin, T. (2010). Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 21–26).

Krishnamurthy, B., Gill, P., & Arlitt, M. (2008). A few chirps about twitter. In *Proceedings of the first workshop on online social networks WOSP'08* (pp. 19–24). New York, NY, USA: ACM.

Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009). Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 457–466).

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web WWW '10* (pp. 591–600). New York, NY, USA: ACM.

Laboreiro, G., Sarmento, L., Teixeira, J., & Oliveira, E. (2010). Tokenizing micro-blogging messages using a text classification approach. In *Proceedings of the fourth workshop on analytics for noisy unstructured text data* (pp. 81–88).

Liu, Z., Huang, W., Zheng, Y., & Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 366–376).

Mann, G. (2006). *Multi-document statistical fact extraction and fusion*. Ph.D. thesis, The Johns Hopkins University.

Mann, G., & Yarowsky, D. (2003). Unsupervised personal name disambiguation. In *Proceedings of the HLT-NAACL'03* (pp. 33–40).

Mann, H., & Whitney, D. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics, 18*(1), 50–60.

Martinez-Romo, J., & Araujo, L. (2009). Web people search disambiguation using language model techniques.

McNamee, P., & Dang, H. (2009). Overview of the TAC 2009 knowledge base population track. In *Text analysis conference (TAC)*.

Meij, E., Weerkamp, W., & de Rijke, M. (2012). Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on web search and data mining*.

Michelson, M., & Macskassy, S. (2010). Discovering users' topics of interest on twitter: A first look. In *Proceedings of the fourth workshop on analytics for noisy unstructured text data* (pp. 73–80).

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). YALE: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD*.

Mihalcea, R., & Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *CIKM* (Vol. 7, pp. 233–242).

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of EMNLP* (Vol. 4, pp. 404–411).

Milios, E., Zhang, Y., He, B., & Dong, L. (2003). Automatic term extraction and document similarity in special text corpora. In *Proceedings of the sixth conference of the pacific association for computational linguistics* (pp. 275–284).

Milne, D., & Witten, I. (2008). Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management* (pp. 509–518).

Milstein, S., Chowdhury, A., Hochmuth, G., Lorica, B., & Magoulas, R. (2008). *Twitter and the micro-messaging revolution: Communication, connections, and immediacy–140 characters at a time*. O'Reilly Media, Inc.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3–26.

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC 2010*.

Pollach, I. (2006). Electronic word of mouth: A genre analysis of product reviews on consumer opinion web sites. In *Proceedings of the 39th annual Hawaii international conference on system sciences* (Vol. 3).

Quinlan, J. (1993). *C4. 5: Programs for machine learning*. Morgan Kaufmann.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th international conference on world wide web WWW'10* (pp. 851–860).

Soboroff, I., McCullough, D., Lin, J., Macdonald, C., Ounis, I., & McCreadie, R. (2012). *Evaluating real-time search over tweets*.

Spina, D., Amigó, E., & Gonzalo, J. (2011). Filter keywords and majority class strategies for company name disambiguation in Twitter. In *Conference on multilingual and multimodal information access evaluation, CLEF 2011* (pp. 50–61). Berlin/Heidelberg: Springer.

Spina, D., Meij, E., Oghina, A., Bui, M.T., Breuss, M., & de Rijke, M. (2012). A corpus for entity profiling in microblog posts. In *LREC workshop on language engineering for online reputation management*.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceeding of the 33rd international ACM SIGIR conference on research and development in information retrieval, SIGIR'10* (pp. 841–842). New York, NY, USA: ACM.

Strube, M., & Ponzetto, S. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the national conference on artificial intelligence* (Vol. 21, p. 1419).

Tsagkias, M., & Balog, K. (2010). The university of Amsterdam at WePS3. In *CLEF 2010 labs and workshops notebook papers*.

Tsagkias, M., de Rijke, M., & Weerkamp, W. (2011). Linking online news and social media. In *Proceedings of the fourth ACM international conference on web search and data mining*.

Turney, P. (1999). Learning to extract keyphrases from text, national research council. Institute for Information Technology, *Technical Report ERB-1057*.

Wan, X., Gao, J., Li, M., & Ding, B. (2005). Person resolution in person search results: Webhawk. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 163–170).

Wilson, R. (2003). Keeping a watch on corporate reputation. *Strategic Communications Management, 7*(2).

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., & Nevill-Manning, C.G. (1999). Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on digital libraries, DL'99* (pp. 254–255).

Wu, S., Hofman, J. M., Mason, W. A., & Watts, D. J. (2011). Who says what to whom on twitter. In *Proceedings of the 20th International Conference on World Wide Web, WWW'11* (pp. 705–714). New York, NY, USA: ACM.

Yang, J., & Leskovec, J. (2010). Modeling information diffusion in implicit networks. *IEEE International Conference on Data Mining, 0*, 599–608.

Yang, J., & Leskovec, J. (2011). Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on web search and data mining, WSDM'11* (pp. 177–186).

Yerva, S.R., Miklós, Z., & Aberer, K. (2010). It was easy when apples and blackberries were only fruits. In *CLEF 2010 labs and workshops notebook papers*.

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

Yoshida, M., Matsushima, S., Ono, S., Sato, I., & Nakagawa, H. (2010). ITC-UT: Tweet categorization by query categorization for on-line reputation management. In *CLEF 2010 labs and workshops notebook papers*.

Zhang, S., Wu, J., Zheng, D., Meng, Y., Xia, Y., & Yu, H. (2012). Two stages based organization name disambiguity. *Computational Linguistics and Intelligent Text Processing*, 249–257.

Zhang, Y., Milios, E., & Zincir-Heywood, N. (2007). A comparative study on key phrase extraction methods in automatic web site summarization. *Journal of Digital Information Management, 5*(5), 323.

Zhang, Y., Zincir-Heywood, N., & Milios, E. (2004a). Term-based clustering and summarization of web page collections. *Advances in Artificial Intelligence*, 60–74.

Zhang, Y., Zincir-Heywood, N., & Milios, E. (2004b). World wide web site summarization. *Web Intelligence and Agent Sstems, 2*, 39–54.

Zhang, Y., Zincir-Heywood, N., & Milios, E. (2005). Narrative text classification for automatic key phrase extraction in web document corpora. In *Proceedings of the 7th annual ACM international workshop on web information and data management* (pp. 51–58).

Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., & Li, X. (2011a). Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd european conference on advances in information retrieval*.

Zhao, X., Jiang, J., He, J., Song, Y., Achananuparp, P., LIM, E., & Li, X. (2011b). *Topical keyphrase extraction from twitter*.