Extracting Unambiguous Keywords from Microposts Using Web and Query Logs Data

Davi de Castro Reis
Google Engineering
Belo Horizonte, Brazil
davi@google.com

Felipe Goldstein
Google Engineering
Paris, France
felipeg@google.com

Frederico Quintao
Google Engineering
Belo Horizonte, Brazil
quintao@google.com

If a lion could talk, we could not understand him.
(Ludwig Wittgenstein)

ABSTRACT

In the recent years, a new form of content type has become ubiquitous in the web. These are small and noisy text snippets, created by users of social networks such as Twitter and Facebook. The full interpretation of those microposts by machines impose tremendous challenges, since they strongly rely on context. In this paper we propose a task which is much simpler than full interpretation of microposts: we aim to build classification systems to detect keywords that unambiguously refer to a single dominant concept, even when taken out of context. For example, in the context of this task, *apple* would be classified as ambiguous whereas *microsoft* would not. The contribution of this work is twofold. First, we formalize this novel classification task that can be directly applied for extracting information from microposts. Second, we show how high precision classifiers for this problem can be built out of Web data and search engine logs, combining traditional information retrieval metrics, such as inverted document frequency, and new ones derived from search query logs. Finally, we have proposed and evaluated relevant applications for these classifiers, which were able to meet precision $\geq$ 72% and recall $\geq$ 56% on unambiguous keyword extraction from microposts. We also compare those results with closely related systems, none of which could outperform those numbers.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic Processing*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*

General Terms

Algorithms, Experimentation

Published as part of the #MSM2012 Workshop proceedings, available online as CEUR Vol-838, at: http://ceur-ws.org/Vol-838
#MSM2012, April 16, 2012, Lyon, France.

Keywords

unambiguity, word sense disambiguation, query logs, web, advertising

1. INTRODUCTION

The availability of huge Web based corpora has spawned a series of important advancements in the Natural Language Processing (NLP) field recently [2, 10]. Moreover, the increase of computational power has made it possible to run simpler algorithms over much more data [7]. However, computer interpretation of natural language is still an unsolved challenge. On the other hand, a very successful set of free text controlled applications have blossomed in the last decade: the Web search engines. The natural language processing techniques employed by these systems are not enough to give users a natural speaking experience, but regardless of that, users type queries in sites like Yahoo!, Bing and Google on a daily basis. Instead of teaching the machine how to speak like us, we ended up learning a simple yet effective way of expressing our information needs.

In this work we start from the premise that full understanding of natural language cannot be achieved with the current technology. In fact, not even a human being is able to fully understand all communication due to either lack of cultural context or to inherent ambiguity in the language. This is especially true in the context of microposts, where both the media culture and technical constraints impose a limit on the size of the messages. Given these limitations, we have aimed to detect parts of natural language that can be unambiguously understood in the lack of any context. The key observation that allowed us to identify those parts of natural language is that they tend to be used as search engine queries on the Web [17].

For example, when using a search engine, if the user wants to know about Michael Jackson, the American singer, dancer, and entertainer, she expects to type these two words in the search box and get relevant results. On the other side, if she is looking for Michael Jackson, the social anthropologist from New Zealand, she knows that she will need to further qualify the query.

In this work we show that search engine query logs are a valuable source of data to build classifiers that can identify unambiguous concepts in a language. In our work, the features for such classifiers are calculated over a crawl of the Web and all queries issued in evenly spread days of a large commercial search engine. Classifiers like these seem especially suited to process short, noisy, conversational texts, which since recently have become widely available on the

Web. We show experimental results from shares and microposts from both Facebook[1] and Twitter[2]. We also propose two different applications to the classifiers built in this paper.

The rest of this paper is organized as follows. Section 2 discusses existing work that is relevant to our system. In Section 3 we discuss in detail how the classifiers proposed in this paper are built and in Section 4 we present numbers assessing their effectiveness. A discussion of potential applications for the system is shown in Section 5. Finally, Section 6 contains our conclusions about this work.

2. RELATED WORK

Krovetz and Croft observed that 75% of early information retrieval systems queries are unambiguous [8]. This observation has been later corroborated by a survey from Sanderson [17], where the impact of word sense disambiguation in information retrieval systems is studied. Although we do not rely only on that observation, one of the core hypothesis of this paper is that, to a lesser extent, this continues to be true for modern search engine queries, as long as the query does not show often in the query logs with further qualification. For example, the query *Washington* often needs to be refined as *George Washington* or *Washington (state)* or even *Washington, D.C.* while the query *Canada* often does not. This hypothesis is the central idea behind the metric described in Session 3.4.2, which has shown to be one of the strongest signals described in this paper.

The usage of the Web as an implicit training set for Natural Language Processing problems and ambiguity resolution in particular is presented in [15], where the author shows that the usage of simple algorithms and features extracted from large amounts of data yield competitive results with sophisticated unsupervised approaches and close results to that of supervised state of the art approaches.

Wacholder et al. studied the problem of disambiguation of proper names in [20]. Nadeu and Sekine conducted a survey [14] on the related field of Named Entity Recognition and Classification (NERC). Finally, the broader area of Word Sense Disambiguation is discussed in depth by Navigli [16].

The problem of extracting information from microposts has gained significant attention recently. In [4], Choudhury and Breslin have presented a classifier for Twitter posts able to detect players and associated micro-events in a sports match, achieving a f-measure of 87%. Using knowledge of the domain, such as sports jargon and names of players, they are able to disambiguate the Tweets. Li et al. proposed using a keyword extraction system for targeting ads to Facebook updates in [12], one of the applications we discuss in Section 4. Signals based on capitalization and document frequency are present in their work, but they did not explore any of the query log derived metrics.

Although the problems presented by the works discussed above share similarities with ours, none of their techniques can be directly applied. Word Sense Disambiguation is focused on finding senses of ambiguous terms in the local context, and does not discuss the properties of a given keyword outside its context. Also, traditional keyword extraction systems extract a set of keywords that characterize or summarize a given text, even if each of the individually extracted keywords might be ambiguous outside that set. Similarly, Named Entity Recognition systems look for entities that may or may not be unambiguous outside their context, such as *Ford* However, in our problem definition, only the keywords *Ford Motors*, *Henry Ford* or *Gerard Ford* should be extracted. Finally, we have no knowledge of the microposts being analyzed, preventing us from using domain specific features.

[1]www.facebook.com
[2]www.twitter.com

3. DETECTING UNAMBIGUITY

The ultimate goal of this work is to develop classifiers that detect unambiguous keywords in a language. As far as the knowledge of the authors goes, this is the first work proposing such classifiers. In order to formally present the problem, we will first introduce a few relevant concepts.

A common step previous to the processing of any corpus is the *segmentation* of the text in the documents. This is a complex problem which is beyond the scope of this paper and we assume that there is a state-of-the-art segmenter available[3]. The output of the segmentation process is a set of *keywords*. One keyword can be composed by one word or by a sequence of words – in the latter case we also refer to it as an n-gram or compound.

The Merriam-Webster dictionary[4] defines ambiguity as (1) *doubtful or uncertain especially from obscurity or indistinctness* or (2) *capable of being understood in two or more possible senses or ways*. However, there is a shortcoming in this definition, since it relies on human interpretation. One person can say that a given word is ambiguous while another could disagree. It turns that both could be right since the interpretation of the senses of a word can be done at different granularities.

Lapata and Keller [11] define ambiguity, or its complement, unambiguity, as function of a frequency of the senses that a given word or compound shows in a large corpus. We will instead use the terminology of the semiotics model by Ferdinand de Saussure [18], which yields a more intuitive definition for the scope of our work.

DEFINITION 1. *Let a pair (f, c) be a sign, being f the form of the sign and c the concept it represents.*[5] *Let L be a language and S the set of all signs used in that language. Let the document frequency of a sign $df(f, c)$ in S be the number of documents the sign appear in a large corpus representative of the language L. We say that f is unambiguous if and only if $df(f, c)/\sum df(f, c') > \alpha$.*

In other words, we say that f is unambiguous if one of the concepts it may represent is α times more frequent in documents of the corpus than all the others combined. For our purposes, f is always a word or a compound word, and given that restriction, we will use Definition 1 as the basis for the problem being discussed through the rest of the paper.

3.1 Keyword Evaluation Methodology

Given a keyword q, an human evaluator can use Definition 1 to rate it as ambiguous or unambiguous. From this

[3]In this paper we use a segmenter developed internally at Google, Inc.
[4]www.merriam-webster.com
[5]In the original, form and concept are called signifier and significant, respectively.

definition, the evaluator should look at all the web documents containing q to understand the sense of the keyword in each of them. In practice, we do not need to go through all the documents in the corpus to find whether a keyword is unambiguous. Instead, we select, from the web, 16 random documents that contain q and manually check if all occurrences refer to the same concept, i.e., all positive occurrences. If yes, we say that the keyword q is unambiguous.

The choice of 16 random documents is legitimate. Since the evaluation of each random document that contains q is an experiment that has only two possible answers, we can assume it is a random sampling over a binomial distribution. Then, by using the Wilson method [21], we can calculate the binomial proportion confidence interval for a sample size of 16 with all of them positive occurrences, i.e. $\hat{p} = 1.0$, which result is $[0.8, 1.0]$ with center at 0.9. This interval gives us the α of Definition 1, which in this case will have a lower bound of 0.8 and an expected value of 0.9 with a 95% confidence. In other words: Given a keyword that was human-evaluated as unambiguous, we have 95% of chance that this keyword will refer to the same dominant concept in 80% of the corpus, but more likely it will be 90% of the corpus. This is the threshold we decided to use to assume a keyword is unambiguous in our human evaluations.

Using this methodology we have built a reference-set with 2634 ambiguous and 426 unambiguous keywords to be used in the analaysis of the metrics presented in the next sections and as training-set input to the Machine Learning approach at Section 3.6.

3.2 Model Generation

There are two main source of signals for the unambiguous keywords classifiers presented here. The first is a sample with billions of documents of Google's search engine web collection. The second is as sample with billions of query entries from Google's query log corpus collected in evenly spread days.

The model generation is composed by a chain of off-line calculations of statistical properties of all signs in these two corpora and takes a few thousands of cpu-hours to complete. These properties will serve as the basis for the classification algorithms. This is an one-off work that only needs to be redone whenever there is an opportunity and/or the need of improving the performance of the system.

3.3 Web Collection Metrics

For every keyword resulting from the segmentation, we compute several properties using a large MapReduce-like system [5] visiting all the documents of the Web corpus. In the next sections we explain each property and give a histogram of its distribution among the keywords. Additionally to the plain histograms, we present two additional complementary histograms, one for the probability density of the metric among the ambiguous and another one for the unambiguous keywords of the reference-set defined in Section 3.1.

3.3.1 Inverse Document Frequency

The first metric, the Inverse Document Frequency (IDF), is the same found in the traditional information retrieval literature. It is computed over the Web document collection and is a proxy of the popularity of the keyword. It also serves as a good confidence indicator for all remaining metrics. The more popular a keyword is, the better the signal-to-noise ratio we have on it for all metrics. Figure 1 shows the IDF distribution for unigrams and keywords found on the Web collection.

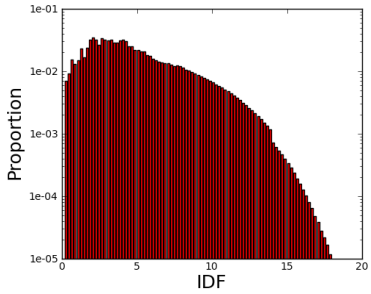

Figure 1: IDF distribution for the web collection.

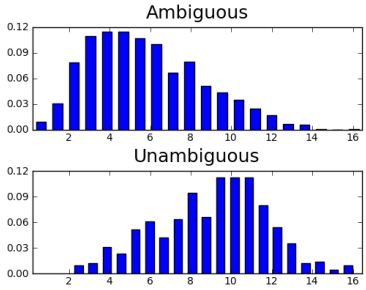

Figure 2: IDF distribution for the reference-set.

The histograms for IDF distribution among the reference-set is plotted in Figure 2. The top chart shows the histogram for ambiguous keywords while the bottom shows the unambiguous. This histogram shows that ambiguous keywords tends to have lower IDF values. The lowest ones are language constructions such as *just* and *this*. The misspellings and uncommon person names lies in the very high IDF range. While unambiguous keywords tends to have higher IDF values, there is a big overlap with lots of unambiguous ones in the mid-lower IDF range, such as *White House* and *Disney*. This overlap makes it hard for the IDF metric to be used to separate both sets. However, we can apply it for filtering language constructions and misspellings.

3.3.2 Caps First Ratio

Caps First Ratio (CFR) is the ratio that a given keyword shows up on the Web collection with the first letter capitalized and we interpret it as strong indicator of names. We implemented the techniques from [13] to detect capitalized keywords.

The CFR metric has the obvious property of detecting nouns, but it has another subtle interesting characteristic. Several noun compounds include, as an extra qualifier, sub-compounds or unigrams that are unambiguous by themselves, for example *Justin Bieber* and *Bieber* are both unambiguous. In this case, we consider the occurrence of every capitalized word not only in the CFR calculation of the compound it belongs to – *Justin Bieber* – , but also in the CFR score of the sub-compounds and unigrams of that compound

– *Bieber*. This helps increasing the CFR score of nouns that act as unambiguous qualifiers. For example, for the *Bieber* unigram, using only the initials for legibility, we calculate:

$$CFR(B) = \frac{count(JB) + count(B)}{count(jb) + count(b) + count(JB) + count(B)}$$

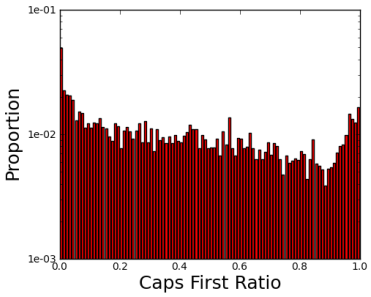

Figure 3: CFR distribution for the web collection.

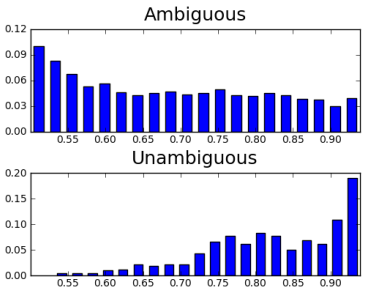

Figure 4: CFR distribution for the reference-set.

Figure 3 shows, the CFR distribution seen on Web documents. The reference-set histograms at Figure 4, are more heterogeneous than the IDF counterpart. The mid-low range of CFR values includes mostly only ambiguous keywords, while the unambiguous histogram has a sharp growth in the high values.

3.4 Query Log Metrics

Query logs have proved to be a valuable source of information for several fields of computer science [19]. In our work we collected data from three evenly spread days worth of queries in the logs of a large search engine. As with the Web corpus, we compute the following metrics for each keyword generated by the segmentation of the query log corpus.

3.4.1 *Sessions Inverse Document Frequency*

The Sessions Inverse Document Frequency (SIDF) is analogous to the Web metric of the same name, but it is calculated over the search engine query stream. Each session [19] is considered as a document. Figures 5 and 6 presents the distribution of this metric for the query stream and for the reference-set respectively. This signal has similar properties to its Web counterpart, but with a bias towards concepts and against intrinsic language characteristics. By comparing Figures 1 and 5, one can draw an interesting conclusion: stopwords and auxiliary language constructions appear much less often in the query stream. Because of that we can say it is safe to discard anything that is not popular in the query stream.

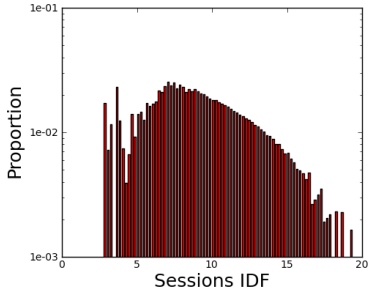

Figure 5: SIDF distribution for an infinite stream of web text.

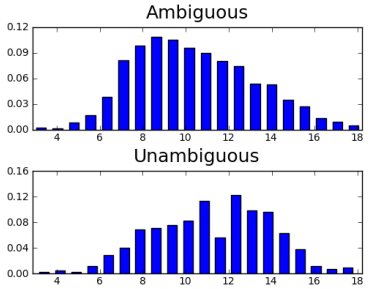

Figure 6: SIDF distribution for the reference-set.

3.4.2 *Sessions Exact Ratio*

A key metric that comes from the query stream analysis is the Sessions Exact Ratio (SER). It tells how often a given keyword shows up by itself in the search box. This is the strongest indicator that this keyword is unambiguous when taken out of context. Figures 7 and 8 shows the histogram for this metric on the Web collection and the reference-set respectively. As can be seen, the ambiguous and unambiguous reference-set is mostly separable. Some examples of unambiguous keywords in the very high range of the histogram are: *Tom Hicks*, *Madison Square Garden* and *Groupon*.

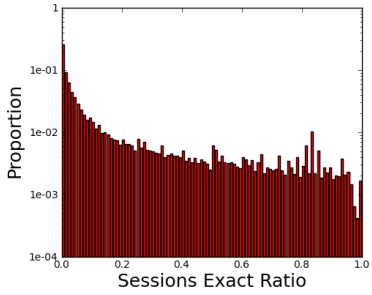

Figure 7: SER distribution for an infinite stream of web text.

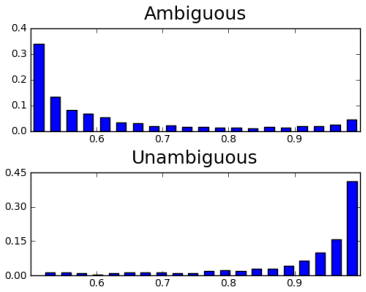

Figure 8: SER distribution for the reference-set.

3.4.3 *Search Bias*

The last metric, Search Bias (SB), is not directly derived from the query stream, but rather obtained through a combination of sessions and Web signals. Search Bias can be thought as the ratio of appearance of a keyword in the query logs corpus divided by the ratio of appearance of the same keyword on the Web corpus.

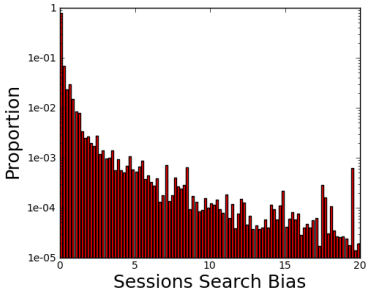

Figure 9: SB distribution for an infinite stream of web text.

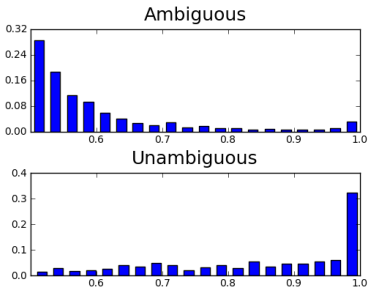

Figure 10: SB distribution for the reference-set.

The naive calculation of this number leads to a value with bad properties due to very common words on the Web corpus and the high frequency of compounds in the query log corpus. To avoid those issues, search bias is calculated taking into account only "noble" occurrences of a keyword in Web and query logs corpora. For the Web, we consider that only capitalized occurrences are noble, while from query logs we only consider those occurrences where the keyword appear by itself in the query. The distribution of this metric can be seen on Figures 9 and 10.

The histograms shown here are not just a tool to help visualize how each metric may be used to dissociate both sets, but more than that, it is an evidence that the metrics used here can succeed in building an effective classifier.

3.5 A hand-crafted classifier

In this section we present a simple hand-crafted algorithm. It was developed upon the discussions and histogram observations of above metrics, regarding the reference-set separation. We use this algorithm to expose the ideas without adding the complexity that inherently comes with traditional machine learning techniques, as well as to avoid hiding the interesting properties of the data under analysis. Later in this paper we present a Support Vector Machines (SVM) approach for the same classification task. Refer to Section 3.6 for more details.

Algorithm 1: The *IsUnambiguous* Algorithm.

```
1  begin
2  |  if sidf > 15 then return false;
3  |  if uni ∧ idf > 12 ∧ sidf > 12 then return false;
4  |  if cfr < 0.4 then return false;
5  |  if ser < 0.35 then return false;
6  |  if sb < 0.01 then return false;
7  |  if cfr + ser + sb < 1 then return false;
8  |  if charcount < 3 then return false;
9  |  if blacklisted then return false;
10 end
```

Algorithm 1 presents our hand-crafted approach for the unambiguity classification problem. Each line is a filter of ambiguous keywords. In Figure 11 one can see how many keywords are being discarded as the classifier is applied on top of the Web corpus. In the end, only 3.8% of all keywords occurrences are considered unambiguous.

The *Sessions IDF* funnel component corresponds to Line 2 of the algorithm. Its goal is to remove everything that is too rare, such as misspells. Usually, plain IDF is used for this type of cutting, but bigrams and larger keywords have a very high IDF on the Web corpus. In the query logs corpus, however, large keywords appear much more often and anything that is not too rare will not be filtered by this rule.

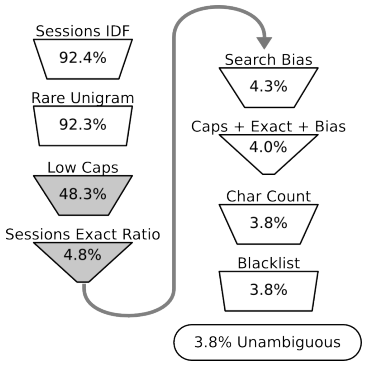

Figure 11: The percentage of the input text keywords each line of the algorithm filters. The most important filters are highlighted in gray.

In Line 3, the *Rare Unigrams* filter unigrams typos that

are not rare enough to be discarded by Sessions IDF and also come with all types of capitalization. Since unigrams are more frequent than compounds, we can apply a more restrictive threshold.

The *Low Caps* filter comes from Line 4 of the algorithm. Not only it is responsible for restricting the classifier to nouns, it also rejects all the general nouns. For example, the noun *ball* has a very low caps first ratio, but *Wilson NCAA Reaction Basketball* is almost always typed all caps.

The most powerful feature for our classifier, the *Sessions Exact Ratio* (SER) filter, is used in Line 5. It reflects the key intuition that our work builds upon: users know that search engines have little context to understand their communication and because of that they formulate unambiguous queries.

The derived metric *Search Bias* is used in Line 6. Some keywords, like *Unfortunately*, tend to have both high CFR – because they are used to start phrases – and high SER – because they have low query volume. This filter detects those language constructions that are way more common in the Web corpus than in the search corpus and discards them.

The combined *Caps+Exact+Bias* filter in Line 7 is the most complex part of the algorithm. Its goal is allow us to reduce the thresholds of the individual filters applied before without incurring in a loss of precision. This filter will let keywords that score very high in any of the metrics combined pass, but will discard those that have a low average all around.

The *Character Count* is a simplistic filter as can be seen in Line 8. When dealing with two characters keywords, all the metrics have bad properties, and we simply discard all of them. In fact, a perfect English classifier limited to two character unigrams can be manually built by inspecting all the 626 possible combinations.

Finally, the last step of the algorithm is the *Blacklist* filter, in Line 9. Some keywords have very extreme metrics and tend to pass through all the filters, and we simply blacklist them. For English, we currently blacklist 40 greetings expressions, such as *Happy birthday* and *Hello* and some very common words like *Indeed*. In fact, the metrics for those keywords are so extreme that by looking at the top values for each metric one can easily spot them. We also blacklisted the Web sites names *Google*, *Facebook*, *Twitter* and *Gmail* because, although unambiguous, they are so common in our evaluation data that they would positively benefit our results with no interesting characteristics.

3.6 A Machine Learning approach

To challenge the hand-crafted algorithm presented in the previous section and test if its intuitions were correct, we employ a Support Vector Machines (SVM) algorithm to the same classification task. It is a well-known technique and there is a ready to use state-of-the-art implementation, namely libSVM [3]. The down-side of the machine learning approach is that labeled data acquisition for this novel classification task is challenging. The training set used was the reference-set explained before, with the 2634 ambiguous and 426 unambiguous manually classified keywords. Each metric shown in sections 3.3 and 3.4 were rescaled to the 0-1 range and used as SVM input features. We used only the Radial Basis Function kernel present in libSVM: $K(x_i, x_j) = \exp(-\gamma \times \|x_i - x_j\|^2), \gamma > 0$. By tuning the γ and C parameters we can control the trade-off between false-positive and false-negative errors. After doing a simple grid-search of both parameters using cross-validation, we picked a value around 0.05 for γ and 0.1 for C.

4. EXPERIMENTAL RESULTS

In this section we present experimental results obtained with the implementation of the classifier described in Section 3.5 and the SVM described in Section 3.6.

4.1 Test set definition for manual evaluation

The input of each classifier is a chunk of free form text, from now on referred to as a *micropost*, and the output is a list of keywords assumed by the classifier to represent unambiguous concepts from the input. We decided to use a micropost as a unit of evaluation, in opposition to a single keyword, because we believe the results from this analysis represent better the real world applications. In order to not favor our classifier, the rater is instructed to mark a micropost as False Positive if she finds one ambiguous keyword classified as unambiguous, even if the classifier also correctly extracted another unambiguous keyword from the same micropost.

We selected two different sources of microposts to feed the classifier: Twitter and Facebook. This data set and the reference-set that was used to train the SVM classifier are disjoint to avoid over-fit. Twitter is a social Web site where users can send and read text-messages with up to 140 characters, called tweets. We expect this kind of social Web site to have mostly conversational text and noise, which carry little information by themselves. To feed the classifiers, each tweet is considered independent from each other and its text is used as input to the classifier. Facebook is a social network site where users can create virtual connections with their friends. One Facebook feature is the status message updates. The status message is much like a tweet, but the 140 characters limit is not imposed. Since users can follow-up on their friends updates, the entire conversation is used as input for the classifiers.

4.2 Methodology for manual evaluation

The methodology used for manual evaluation consists of collecting a random set of microposts from each data source and feeding each one into the classifiers. The original micropost and the output of the classifier are then shown to three different raters. The output might be empty, in the case the classifier did not find any unambiguous keyword in the micropost. Otherwise it contains at least one keyword, which was classified as unambiguous. In case of discordance, it is discussed until consensus is reached. Regardless of the classifier output, raters must investigate the keywords present in the micropost. They use the methodology presented in Section 3.1 to rate each keyword q. Based on the output of the classifier and the inspection of the micropost content carried out by the rater, each micropost is rated as below:

True Positive (TP): There are unambiguous keywords in the micropost and the system has extracted at least one.

True Negative (TN): There are no unambiguous keywords and the system extracted nothing.

False Positive (FP): The system has extracted an ambiguous keyword.

False Negative (FN): There are unambiguous keywords in the micropost but the system has extracted none.

We use the output of the rating phase to compute the standard evaluation metrics in the Information Retrieval field, such as precision, recall, accuracy and F-score [1].

Both models – the hand-crafted and the SVM classifier – were built using the context available at the English Web, but it must be considered that people have an incomplete cultural context and sometimes it may not be obvious that a given keyword is unambiguous. For example, during the evaluation one rater could not recognize upfront the keyword *Doug Flutie*, which was extracted by the system. Even though this rater did not recognize this keyword, *Doug Flutie* is indeed an unambiguous keyword because every person with culture about American Football will recognize him as Douglas Richard "Doug" Flutie, a famous football quarterback who played professionally in the United States Football League and, more importantly, the name *Doug Flutie* is not used as an identifier in any other significant context besides that. Our precise definition of unambiguity prevents this problem, since the rater will learn the sense(s) of the keyword when looking at the 16 randomly sampled documents, as it was the case in this example.

4.3 Numerical results

Table 1 presents the output of the raters for Facebook and Twitter microposts.

	Hand Algorithm		**SVM**	
	Twitter	Facebook	Twitter	Facebook
TP	99	106	85	85
TN	494	480	511	510
FP	38	34	22	21
FN	74	64	87	68

Table 1: Break-down of metrics for Twitter and Facebook.

Twitter

Following the experimental methodology we analyzed a set of 705 tweets, which were randomly selected from a set of 170k tweets that were crawled by a fairly random walk in the Twitter graph. We used these tweets as input of both classifiers and presented the output to the raters. The hand-crafted classifier was able to reach precision of 72.26%, and sensitivity (recall) of 56.22%. The True Negative Rate (TNR, or specificity) is high (92.86%), upon what one can conclude that most tweets do not contain unambiguous keywords. For Twitter the system reached an accuracy of 84% with an F-Score of 0.64. The SVM model reached a precision of 79.43%, i.e., a performance almost 10% better than the achieved by the hand-crafted algorithm. The achieved recall is 49.42%, considerably worse than the recall reached by the hand-crafted algorithm. The TNR of the SVM model is 95.87%, and the system reached an accuracy of 85% with an F-Score of 0.61.

Facebook

Following a random selection strategy similar to Twitter, we collected 684 conversations that took place around Facebook status message updates. For this data set, the hand-crafted system reached a precision of 75.71% and recall of 62.36%. The TNR was of 93.39%, whereas the accuracy reached 85% with an F-score of 0.68. The SVM model got slightly better results. The precision is 80.19% (around 6% better), whereas the recall is 55.55%. Again, the True Negative Rate is really high, 96.05%. The classifier has an accuracy of 87% with an F-Score of 0.66. The high value for the True Negative Rate is a sign that most conversations in Social Networks like Facebook and Twitter are not proper for context-extraction systems such as content-targeted advertisement if used without any pre-processing.

To compare the results of the two classifiers presented above, we also evaluated two known systems: Yahoo! Term Extractor API – aimed to Keyword Extraction tasks – reached 18.98% of precision and 57.69% of recall for Facebook data, and 14.89% of precision and 77.7% of recall for Twitter data; and the Stanford Named Entity Recognizer [6] – aimed to Named Entity Recognition and Classification tasks – reached 35% of precision and 69.99% of recall for Facebook data, and 39.65% of precision and 74.19% of recall for Twitter data.

Both systems reached a higher recall, but for the real-world applications discussed in section 5 we cannot afford extracting a wrong keyword from a noisy text. In these applications precision is more important, and for both systems it is much lower than the two filters developed in this work. The high recall and low precision result is expected for these systems, since they were engineered for different tasks and do not perform well for the unambiguity detection task defined here.

5. APPLICATIONS

Given the properties of the classifiers presented in this paper, we believe they are suited for a set of different applications that are becoming more important given last developments on the Web industry.

5.1 Ad targeting in Social Networks

In Social Networks users keep updating their status messages (or tweets) with what they have in mind. The number of daily updates in the most prominent networks is huge[6], turning this channel into a potential candidate for input of content-targeted advertisement systems [12]. For instance, it is just fine to deliver an advertisement piece like *Buy tickets to the coming Jonas Brothers show!*, right next to a micropost where a user claims to be the biggest fan of this music group. However, the conversational text brings even more complexity for the already tough task [9] of delivering content-targeted ads. Feeding these systems with noisy text may lead them to return non-relevant ads. One can use the classifiers proposed in this paper as a filtering layer on top of current content-targeted advertisement systems. The filter would delegate calls to the ads systems only when it is possible to retrieve relevant content from the microposts being targeted.

5.2 Automatic reference in Social Networks

User profiles in Social Networks could also be classified as unambiguous by using the profile name for example. Whenever a micropost has the unambiguous keywords that matches a profile name, a link to that profile could be added, instead of just pure text. This could be done for celebrity profiles, for example, when a user posts *"I just watched the last Quentin Tarantino movie."*, a link to the *Quentin Tarantino* profile could be added.

[6]http://news.cnet.com/8301-13577_3-10378353-36.html

6. CONCLUSIONS

In this paper we presented a novel classification problem aimed at identifying the unambiguous keywords in a language, and formally defined it together with an evaluation methodology. We also have presented two different algorithms for the classification problem and the corresponding numerical results achieved by both of them. The proposed algorithms are built on top of traditional information retrieval metrics and novel metrics based on the query log corpus. The introduction of these metrics, Sessions IDF, Sessions Exact Ratio and Search Bias, is by itself an important contribution. We believe these metrics will be useful in other problem domains as well.

Our evaluation have shown that our classifiers were able to meet precision $\geq$ 72%, recall $\geq$ 49%, accuracy $\geq$ 84% and F-Score $\geq$ 0.61, even when the input is composed by the noisy microposts from Facebook and Twitter, two of the biggest sites in the world nowadays, outperforming two known systems from the traditional keyword extraction and Named Entity Recognition fields.

Another interesting aspect of the presented work is that it diverges from the bag-of-words analyses that dominate the research in the area. Instead, we have focused on directly finding the specific keyword that define a concept, avoiding the shortcomings that come from having a representation that cannot be understood by a human or does not meet the expectations of other systems. This leads immediatelly to our future work proposal of using the extracted keywords as beacons for further qualification of other keywords in the text. For example, the extracted keyword *Lionel Messi* can be used to anchor the word *goal* to the concept of scoring in the soccer sport, instead rather the more general idea of an objective to be achieved. We expect this inside-out approach for extracting semantics in microposts to perform better than traditional word collection approches.

More and more researchers have access to query logs and many may directly benefit from the metrics proposed here either to tackle the same classification problem or to innovate in their own domains. For the industry, we have shown a solution for extracting information from microposts, a type of content that has experienced tremendous growth on the Web in the recent past.

7. REFERENCES

[1] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[2] M. Banko and E. Brill. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 26–33, Morristown, NJ, USA, 2001.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: a Library for Support Vector Machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[4] S. Choudhury and J. Breslin. Extracting semantic entities and events from sports tweets. In *Making Sense of Microposts (#MSM2011)*, pages 22–32, 2011.

[5] J. Dean and S. Ghemawat. Mapreduce: Simplified Data Processing on Large Clusters. In *Sixth Symposium on Operating System Design and Implementation*, pages 137–150, 2004.

[6] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[7] A. Halevy, P. Norvig, and F. Pereira. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24:8–12, 2009.

[8] R. Krovetz and W. B. Croft. Lexical Ambiguity and Information Retrieval. *ACM Transactions on Information Systems*, 10:115–141, April 1992.

[9] A. Lacerda, M. Cristo, M. A. Gonçalves, W. Fan, N. Ziviani, and B. Ribeiro-Neto. Learning to Advertise. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 549–556, New York, NY, USA, 2006.

[10] M. Lapata and F. Keller. Web-based Models for Natural Language Processing. *ACM Transactions on Speech and Language Processing*, 2(1):3, 2005.

[11] M. Lapata and F. Keller. An Information Retrieval Approach to Sense Ranking. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 348–355, 2007.

[12] Z. Li, D. Zhou, Y.-F. Juan, and J. Han. Keyword Extraction for Social Snippets. In *Proceedings of the 19th ACM International Conference on World wide web*, pages 1143–1144, New York, NY, USA, 2010.

[13] A. Mikheev. A Knowledge-free Method for Capitalized Word Disambiguation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 159–166, Morristown, NJ, USA, 1999.

[14] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007.

[15] P. Nakov and M. Hearst. Using the Web as an Implicit Training Set: Application to Structural Ambiguity Resolution. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 835–842, Morristown, NJ, USA, 2005.

[16] R. Navigli. Word Sense Disambiguation: A Survey. *ACM Comput. Surv.*, 41(2):1–69, 2009.

[17] M. Sanderson. Retrieving with Good Sense. *Information Retrieval*, 2(1):49–69, 2000.

[18] F. D. Saussure. *Course in General Linguistics*. Open Court Pub Co, 1986.

[19] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum*, 33(1):6–12, 1999.

[20] N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of Proper Names in Text. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 202–208, Morristown, NJ, USA, 1997.

[21] E. B. Wilson. Probable Inference, the Law of Succession, and Statistical Inference. *Journal of the American Statistical Association*, 22(158):pp. 209–212, 1927.