Exploiting New Sentiment-Based Meta-level Features for Effective Sentiment Analysis

Sérgio Canuto
Federal University of Minas Gerais
Computer Science Department
Belo Horizonte, MG, Brazil
sergiodaniel@dcc.ufmg.br

Marcos André Gonçalves
Federal University of Minas Gerais
Computer Science Department
Belo Horizonte, MG, Brazil
mgoncalv@dcc.ufmg.br

Fabrício Benevenuto
Federal University of Minas Gerais
Computer Science Department
Belo Horizonte, MG, Brazil
fabricio@dcc.ufmg.br

ABSTRACT

In this paper we address the problem of automatically learning to classify the sentiment of short messages/reviews by exploiting information derived from meta-level features i.e., features derived primarily from the original bag-of-words representation. We propose new meta-level features especially designed for the sentiment analysis of short messages such as: (i) information derived from the sentiment distribution among the k nearest neighbors of a given short test document x, (ii) the distribution of distances of x to their neighbors and (iii) the document polarity of these neighbors given by unsupervised lexical-based methods. Our approach is also capable of exploiting information from the neighborhood of document x regarding (highly noisy) data obtained from 1.6 million Twitter messages with emoticons. The set of proposed features is capable of transforming the original feature space into a new one, potentially smaller and more informed. Experiments performed with a substantial number of datasets (nineteen) demonstrate that the effectiveness of the proposed sentiment-based meta-level features is not only superior to the traditional bag-of-word representation (by up to 16%) but is also superior in most cases to state-of-art meta-level features previously proposed in the literature for text classification tasks that do not take into account some idiosyncrasies of sentiment analysis. Our proposal is also largely superior to the best lexicon-based methods as well as to supervised combinations of them. In fact, the proposed approach is the only one to produce the best results in all tested datasets in all scenarios.

CCS Concepts

•Information systems → Document representation; •Computing methodologies → Machine learning approaches;

WSDM'16, February 22–25, 2016, San Francisco, CA, USA.
© 2016 ACM. ISBN 978-1-4503-3716-8/16/02 ...$15.00.
DOI: http://dx.doi.org/10.1145/2835776.2835821

Keywords

meta features, sentiment analysis

1. INTRODUCTION

The popularity of online forums, reviews and social networks has led numerous people to share their opinions on a wide range of subjects, including products, events, news and even daily experiences. Dealing with this massive amount of data, generated everyday on online platforms, can bring a number of new opportunities to businesses and markets. In particular, the sentiment analysis of such unstructured data can reveal how people feel about a particular product or service.

In this work, we focus on a supervised learning paradigm to deal with sentiment classification of short messages/(micro-)reviews, since it is one of the most effective and adaptable approaches for this task [18]. Given a set of training messages classified into one or more predefined sentiments/polarities, the task is to automatically learn how to classify new (unclassified) messages, using a combination of features of these messages that associate them with predefined sentiments or polarities. In particular, we focus on the supervised (binary) task of discriminating between positive and negative polarities of the messages. The reasons for this are threefold: (i) in several domains (e.g., reviews and micro-reviews), the basic motivation for people to write such messages is to provide positive or negative feedback on products, experiences and services that can be helpful to others; (ii) even in other domains in which "neutral" opinions can occur more frequently, many applications are interested in knowing only the most "polarized" opinions about certain topics (e..g., politicians, events, etc); and finally (iii), even if identifying neutral positions is important, some works (e.g., [4, 25, 33] have advocated doing this in a prior step (aka, subjectivity extraction) before determining the polarity of the message, which is our focus here.

A recent trend that has emerged in supervised approaches for text classification, that works in the data engineering level instead of in the algorithmic level, is the introduction of meta-level features[1] that can replace or work in conjunction with the the original set of (bag-of-words-based) features [7, 6, 21, 22, 27]. Such meta-level features, which are usually manually designed and extracted from other features, cap-

[1]In this paper, we will use the terms "meta-level features" and "meta-features" interchangeably.

ture intrinsic relationships among a pair $(document, class)$ or a triple $(document, class, algorithm)$. These meta-level features can capture insightful new information about the unknown underlying data distribution that relates the observed patterns to the associated category.

However, despite their potential, meta-level features have not been extensively studied in sentiment classification tasks. Specifically, they were not evaluated with any of the most common sentiment analysis benchmarks for this task (e.g. twitter, youtube, reviews, amazon, etc). In this paper we fulfill this gap. Moreover, inspired by recent work [6, 7, 17, 35] which takes into account the labels of training documents to generate meta-level features, we propose a new set of meta-level features based on the lazy classifier method kNN as well as freely available evidence towards the positivity or negativity of messages.

These new meta-level features exploit sentiment and distances distributions in a possibly noisy neighborhood considering the scarcity of information in short messages. In more details, we make use of BM25 [23] as similarity score in kNN, since it is an useful measure to rank documents with short messages as queries. We also exploit the neighborhood of a test example in both the training set and in a dataset containing 1.6 millions of tweets automatically labeled by its users with emoticons [14]. This methodology allow us to exploit discriminative information from different domains, even in noisy ones, like the large twitter dataset with emoticons. The last additional evidence we exploit is taken from the weighted sentiment polarity of the nearest neighbors by using recent lexicon-based methods to infer the message's polarity towards a sentiment [2, 19, 31]. In our experiments our approach outperformed by large margins all baselines which include: (i) the traditional Bag-of-word representation; (ii) the sets of meta-features proposed previously in the literature for general text classification tasks; (iii) the best lexicon-based methods; and (iv) supervised ensembles of lexicon-based methods. In fact, our approach was the sole one to win in all (nineteen) tested datasets.

In sum, the main contributions of this paper are: (i) the proposition and evaluation of new meta-level features especially designed for sentiment analysis of short messages; (ii) a comprehensive study on the use of ours and previously proposed kNN-based meta-level features in this context; (iii) a simple method to easily extract discriminative information from highly noisy external data.

This paper is organized as follows. In Section 2 we discuss related work. In Section 3 we introduce our proposed meta-level features and the state-of-art kNN-based meta-level features. In Section 4 we present the experimental setup and discuss the experimental results. Finally, in Section 5 we present our conclusions and highlight future work.

2. RELATED WORK

2.1 Sentiment Analysis

Current sentiment extraction tasks differ according to the types of classes predicted (positive or negative, subjective or objective), the classification techniques used, and the considered classification levels, which comprehend sentence, aspect or document level [12]. Each level depends on the granularity of the sentiment class, which may be predicted to whole documents (for document-level), to individual sentences (for sentence-level) or to specific aspects of entities (for aspect-level). In this work, we focus at the sentiment detection of messages composed of one or a few short sentences, such as tweets, comments and (micro-)reviews.

Regarding the document classes, there is an active research on subjectivity classification [5, 34, 29, 3], in which the goal is to discriminate objective messages from subjective ones. Some works separate subjective sentences from objective ones and then classify the polarity (positive or negative) of the opinions expressed in the subjective sentences [4, 25, 33]. These efforts present empirical evidence that performing a simultaneous classification between positive, negative and objective (neutral) messages is worst than a two-step approach that filters the objective messages first.

Regarding the classification techniques, most methods can be categorized into one of two main groups: supervised approaches [16], which use a wide range of features and labeled data for training sentiment classifiers, and lexicon-based approaches [2, 19, 31], which make use of pre-built lexicons of words weighted with their sentiment orientations to determine the overall sentiment of a given document. Specifically, recent methods such as Vader [19] and SentiStrength [31] rely on linguistic clues (e.g., punctuation and words that intensify or invert polarity) and on a dictionary of words provided with their sentiment scores, manually labeled [2]. Their dictionaries differ, since Vader's lexicons include most internet slag terms and SentiStrength rely on the combination of diverse available lexicon dictionaries. They also differ on their scoring scheme, since SentiStrength [31] weights the sentiment of a message by attributing to it the sentiment score of the most positively (or negatively) classified word in the message, and Vader sums the scores in of the words in the message. Instead of relying on a dictionary of lexicons, SentiWordNet [2] explores relationships between unambiguous polarized (or neutral) "concepts" in order to find the average polarity of close concepts associated to a message. Therefore, terms like dog and bite (both representing neutral concepts in SentiWordNet) appearing in the same message could eventually be expanded with a more emotional term like hurt, which holds a negative polarity. We choose to use these three lexicons as the basis to construct our meta-level features as they showed to be effective for sentiment detection in recent benchmark comparison studies [15, 1, 28].

Overall, unlike the above efforts, most supervised approaches are direct applications of the traditional classification techniques. Thus, to the best of our knowledge, our effort is the first to propose specific meta-level features for text classification in the sentiment analysis domain.

2.2 Meta-Level Features for Text Classification

Meta-features have been proposed to improve the effectiveness of text classification. They can be based on ensemble of classifiers [10, 32], derived from clustering methods [21, 27] or from the instance-based kNN method [17, 35].

Meta-features derived from ensembles exploit the probability distribution over all classes generated by each of the individual classifiers composing the ensemble [32]. In [10] other ensemble-based meta-features were also used, includ-

[2]the sentiment of each word was manually adjusted. In Vader, for example, each word was manually scored with 9 degrees of valence (between extremely positive to extremely negative).

ing: the entropies of the class probability distributions and the maximum probability returned by each classifier. This scheme was found to perform better than using only probability distributions. The most recent effort regarding sentiment analysis towards this kind of meta-feature uses only the scores of the first level classifier's output [18, 8].

Clustering techniques may also be used to derive meta-features. In this case, the feature space is augmented using clusters derived from a previous clustering step considering both the labeled and unlabeled data [27, 21]. The idea is that clusters represent higher level "concepts" in the feature space, and the features derived from the clusters indicate the similarity of each example to these concepts. In [27] the largest n clusters are chosen as representatives of the major concepts. Each cluster c contributes with a set o meta-features like, for instance, binary feature indicating if c is the closest of the n clusters to the example, the similarity of the example to the cluster's centroid, among others. In [21] the number of clusters is chosen to be equal to the predefined number of classes and each cluster corresponds to an additional meta-feature.

Recently, several works [6, 7, 17, 35] have been proposed to use kNN as the main tool to generate meta-features. Despite the fact that these meta-features are not created based on an ensemble of classifiers, they differ from the previously presented meta-features derived from clusters because they explicitly capture information from the labeled set. In more details, [17, 35] reported good results by designing meta-features that make a combined use of local information (through kNN-based features) and global information (through category centroids) in the training set. This work is extended in [7] by the use of meta-features derived from the class distribution, the entropy and the within-class cohesion observed in the k nearest neighbors of a given test document x. In order to make the meta-feature generation viable for huge data, [6] proposed a GPU-based kNN implementation for highly dimensional and sparse textual data.

The kNN-based meta features proposed in this work differ from the ones presented in previous works [17, 35] due to the fact that we focus on engineering meta-features for sentiment analysis in short messages. Besides using BM25 as similarity measure, we explore additional information from a large dataset containing 1.6 millions of emoticons [14] and from recently proposed lexicon-based methods [31, 19, 2] to infer the document's polarity.

In next section we describe in details previously proposed meta-level features for general text classification tasks, as well as our newly proposed kNN-based ones for sentiment analysis. As we shall see, our proposed meta-features are more effective then the previous ones in most datasets as they consider intrinsic aspects of the sentiment analysis of short messages.

3. META-LEVEL FEATURES

Let $\mathcal{X}$ and $\mathcal{C}$ denote the input (feature) and output (class) spaces, respectively. Let $\mathbb{D}_{train} = \{(x_i, c_i) \in \mathcal{X} \times \mathcal{C}\}|_{i=1}^{n}$ be the training set. Recall that the main goal of supervised classification is to learn a mapping function $h : \mathcal{X} \mapsto \mathcal{C}$ which is general enough to accurately classify examples $x' \notin \mathbb{D}_{train}$.

3.1 Gopal et al's Meta-Level Features

The kNN-based meta-level features proposed in [17], are designed to replace the original input space $\mathcal{X}$ with a new informative and compact input space $\mathcal{M}$. Therefore, each vector of meta-level features $m_f \in \mathcal{M}$ is expressed as the concatenation of the following sub-vectors, which are defined for each example $x_f \in \mathcal{X}$ and category $c_j \in \mathcal{C}$ for $j = 1, 2, \ldots, |\mathcal{C}|$.

- $\vec{v}_{\vec{x_f}}^{dist} = [dist(\vec{x}_{ij}, \vec{x_f})]$ A k-dimensional vector whose elements $dist(\vec{x}_{ij}, \vec{x}_f)$ denote a distance score between $\vec{x}_f$ and the i^{th} nearest class c_j neighbor of $\vec{x}_f$
- $\vec{v}_{\vec{x_f}}^{cent} = [dist(\vec{x}_j, \vec{x_f})]$ A 1-dimensional vector where $\vec{x}_j$ is the c_j centroid (i.e., vector average of all training examples of the class c_j).

These sub-vectors are composed of distances (Euclidean, Manhattan and Cosine) that make a combined use of local information (through kNN-based features) and global information (through category centroids). More specifically, each test example is directly compared to a set of nearest labeled examples and category centroids, which are assumed to be enough to effectively characterize and discriminate categories. The intuition behind these meta-features consists in the assumption that if the distances between an example to the nearest neighbors belonging to the category c (and its corresponding centroid) are small, then the example is likely to belong to c.

Considering k neighbors and using three distance measures, the number of features in vector x_f is $(3k + 3)$ per category, and the total of $(3k + 3)|\mathcal{C}|$ for all categories. The size of this meta-level feature set is much smaller than that typically found in ATC tasks, while explicitly capturing class discriminative information from the labeled set.

3.2 Canuto et al's Meta-Level Features

Similarly to the meta-level features previously presented, the meta-features proposed in [7] are also based on nearest neighbor search. However, they go beyond by also exploiting discriminative information from the labeled set under several aspects, namely the class distribution in the neighborhood of the test example $\vec{x_f}$, the within-class cohesion and entropy.

In particular, the first key aspect of these meta-features is the fact that they exploit the continuity hypothesis which guarantees the kNN classifier's success: the existence of a mode in the class distribution of the neighborhood of $\vec{x_f}$ usually determines the category of $\vec{x_f}$. They also exploit a summarized version of the Gopal et al's meta-features through category distance quartiles instead of the full distance distribution, which reduces considerably the number of dimensions and prevents overfitting in small datasets.

Another key aspect exploited by these meta-features refers to proximities of the neighbors of $\vec{m_f}$ belonging to some class $c_i \neq c_j$ to the centroid of c_j. This directly evaluates the class cohesion in the neighborhood of $\vec{x_f}$, being an important information regarding the uncertainty level in such region of the input space. Finally, the entropy of the neighborhood and the correlation between neighbors from different classes provide additional evidence about the purity of the top ranked neighbors.

3.3 Proposed Meta-Level Features for Sentiment Analysis

Similarly to the baseline meta-level features, the proposed meta-features for sentiment analysis are based on nearest neighbor search. However, we focus on dealing with specific

aspects of sentiment analysis of short messages. We use BM25 [23] for dealing with short texts as queries and exploit additional information from lexical-based methods as well as the inexpensive (though noisy) tweet messages labeled by Tweeter users with emoticons.

Given the examples in the original input space $\mathcal{X}$, the proposed vector of meta-level features $m_f \in \mathcal{M}$ is expressed as the concatenation of the following sub-vectors, which are defined for each example $x_f \in \mathcal{X}$ and category $c_j \in \mathcal{C}$ for $j = 1, 2, \ldots, |\mathcal{C}|$.

- $\vec{v}_{\vec{x}_f}^{rawsim} = [sim(\vec{x}_{ij}, \vec{x_f})]$ A k-dimensional vector produced by considering the k nearest neighbors of class c_j to the target vector x_f. More specifically, $\vec{x}_{ij}$ is the i^{th} $(i \leq k)$ nearest neighbor to $\vec{x}_f$, and $sim(\vec{x}_{ij}, \vec{x_f})$ is the a similarity score (BM25 or cosine) between them. Thus, k meta-level features are generated to represent x_f.

- $\vec{v}_{\vec{x}_f}^{cagglex} = [\sum sim(\vec{x}_{ij}, \vec{x_f}) * p_{ij}]$ A 1-dimensional vector produced by considering the weighted polarity sum of the k nearest neighbors of x_f that belong to the class c_j. The similarity $sim(\vec{x}_{ij}, \vec{x_f})$ between $\vec{x}_f$ and its i^{th} $(i \leq k)$ nearest neighbor $\vec{x}_{ij}$ is used to weight the polarity p_{ij} of the document x_{ij}. The polarity score p_{ij} (that represents the valence of a positive sentiment, for example) is given by a lexical-based method.

- $\vec{v}_{\vec{x}_f}^{maxminlex} = [max(sim(\vec{x}_{ij}, \vec{x_f}) * p_{ij}), min(sim(\vec{x}_{ij}, \vec{x_f}) * p_{ij})]$ A 2-dimensional vector produced by considering the maximum and minimum document weighted polarity of the k nearest neighbors of the target vector x_f that belong to the class c_j. The similarity $sim(\vec{x}_{ij}, \vec{x_f})$ between $\vec{x}_f$ and its i^{th} $(i \leq k)$ nearest neighbor $\vec{x}_{ij}$ is used to weight the polarity p_{ij} of the document x_{ij}.

- $\vec{v}_{\vec{x}_f}^{agglex} = \left[\sum sim(\vec{t}_i, \vec{x_f}) * p_i)\right]$ A 1-dimensional vector with the weighted polarity sum of the k-nearest neighbor's polarities. The similarity $sim(\vec{t}_i, \vec{x_f})$ between $\vec{x}_f$ and its i^{th} $(i \leq k)$ nearest neighbor $\vec{t}_i$ is used to weight the polarity p_i of the document t_i

- $\vec{v}_{\vec{x}_f}^{rawlex} = [p_f]$ A 1-dimensional vector produced by one of the (possibly many) outputs of a lexical-based method. The output p_f corresponds to the polarity score of the document represented by x_f.

With exception of $\vec{v}_{\vec{x}_f}^{rawlex}$, all remaining described vectors depend on a similarity measure in order to find the nearest neighbors. We generate our meta-features with two similarity scores: Cosine (i.e., $sim(\vec{x}, \vec{q}) = Cosine(\vec{x}, \vec{q})$) and BM25 (i.e., $sim(\vec{x}, \vec{q}) = BM25(\vec{x}, \vec{q})$). Cosine similarity is one of the most popular similarity measures applied to text documents in numerous information retrieval, text classification and clustering applications. Furthermore, it produced very good results when applied together with bag-of-words weighted with TF-IDF in previous meta-features studies [7]. We also compute the vectors using the BM25 similarity score due to its effectiveness on dealing with short queries in information retrieval, which is the case when dealing with short messages as queries for kNN. These two similarity scores are fundamentally different, since BM25 computes the similarity based on terms in a query document appearing in other documents, regardless the additional terms that are not in the query document. On the contrary, cosine similarity considers all terms from both compared documents, since it is symmetrical.

The vectors were grouped into 4 categories, considering whether they use only the polarity information (RAWLEX), only the neighborhood distance information (RAWSIM), the combination of distances with polarities (KNNLEX) and information generated from the external data containing tweet messages labeled with emoticons, instead of the original training data (TWEMOT). Table 1 summarizes the names we give to different groups of meta-features depending on the variations of these factors.

Group	Description
RAWSIM	Vector $\vec{v}_{\vec{x}_f}^{rawsim}$
RAWLEX	Vector $\vec{v}_{\vec{x}_f}^{rawlex}$
KNNLEX	Vectors $\vec{v}_{\vec{x}_f}^{cagglex}$, $\vec{v}_{\vec{x}_f}^{maxminlex}$ and $\vec{v}_{\vec{x}_f}^{agglex}$
TWEMOT	Vectors $\vec{v}_{\vec{x}_f}^{cagglex}$, $\vec{v}_{\vec{x}_f}^{maxminlex}$, $\vec{v}_{\vec{x}_f}^{agglex}$ and $\vec{v}_{\vec{x}_f}^{rawsim}$ computed using only the neighborhood from the tweet dataset with emoticons.

Table 1: Groups of proposed meta-features.

As described in Table 1, the proposed meta-features are able to capture discriminative information from the labeled set using different sources of information. We now further analyze the proposed groups of meta-features, describing which characteristics of the data each of them aim to exploit.

For the RAWSIM features, each test example $\vec{x}_f$ is directly compared to a set of nearest labeled examples. The intuition behind these meta-features consists in the assumption that if the distances between an example to the nearest neighbors belonging to a category c are small, then the example is likely to belong to c.

The RAWLEX features are the raw scores produced by outputs of the used lexicon-based methods. All used methods produce one score for positive polarity and another for negative polarity. Each one of these raw scores correspond to a one-dimensional vector $\vec{v}_{\vec{x}_f}^{rawlex}$.

The KNNLEX features also use the neighborhood distances, but they are only employed for finding and weighting the polarity of the nearest neighbors. More specifically, the meta-features $\vec{v}_{\vec{x}_f}^{cagglex}$ supply the agglomerated polarity of the category c's neighbors. This meta-feature obtains, for each neighbor, the polarity valence p towards a sentiment (e.g., negativity score) using a lexicon-based method (e.g., SentiStrength). The weighted sum of these polarities can be seen as the summarization of the distribution of the neighbor's polarities. If the neighbors of a test document $\vec{x}_f$ have very high values for the positive polarity p and they are also very close to $\vec{x}_f$, the sentiment of $\vec{x}_f$ is likely to be positive. This summarization can be useful, since the lexical-based approaches usually have a small coverage of words which often are not in a particular message. The exploitation of the neighborhood of this message aims at easing this coverage problem by looking at the words of the neighbors.

Another way to analyze the neighborhood polarity distribution is by looking at the maximum and minimum values. The meta-features $\vec{v}_{\vec{x}_f}^{maxminlex}$ extract this additional information to complement the agglomerated polarity previously described. The final meta-feature belonging to the KNNLEX group is the agglomeration $\vec{v}_{\vec{x}_f}^{agglex}$. It differs from

the other meta-features because it does not use the labeled information, but just summarizes the polarity of the k nearest neighbors without looking at their categories. By ignoring the category of the neighbors, it is possible to exploit the existence of a mode in the polarity distribution of the neighborhood.

Most meta-features depend on a training set in which we find the k-nearest neighbors. We generate them using the traditional training set (that follows the same distribution of the test documents). Besides this traditional training set, we use an additional very large dataset with 1.6 million tweet messages with emoticons inserted by the users (which can be considered as noisy labels) as a training set. The group of meta-features TWEMOT comprises all the proposed kNN-based meta-features generated using this twitter training set. There are two factors that make possible the exploration of this highly noisy tweet dataset. First, we exclude all messages from the tweet dataset that are not in the neighborhood of a query document $\vec{x}_f$. By doing so, we ignore a huge volume of messages that are not close to the domain of interest in which $\vec{x}_f$ is inserted. The second factor is related to the fact that the neighborhood's information is summarized with similarity scores. This summarization with fewer features can be easily evaluated regarding its discriminative power.

Considering k_t neighbors from the tweet dataset, k_o neighbors from the original training data, the $|\mathcal{C}|$ sentiment categories, 2 similarity measures (cosine and BM25) and p polarity scores, the number of features generated will be $2|\mathcal{C}|(k_t + k_o) + p + 4(1 + 3|\mathcal{C}|)$. Specifically, k_t meta-features per category are generated by the vector $\vec{v}^{rawsim}_{\vec{x}_f}$ derived from the k_t nearest neighbors from the tweet dataset. Since we use two similarity measures, we will generate one vector for each similarity measure which gives us $2k_t$ meta-features per category, or $|\mathcal{C}|2k_t$. Similarly, we generate $2k_o$ meta-features driven by the original labeled data, which gives us $2|\mathcal{C}|(k_t + k_o)$ meta-features generated by $\vec{v}^{rawsim}_{\vec{x}_f}$. The equation also accounts for p polarity scores given by each single lexical-based method output, as described in $\vec{v}^{rawlex}_{\vec{x}_f}$.

The vectors $\vec{v}^{cagglex}_{\vec{x}_f}$ and $\vec{v}^{maxminlex}_{\vec{x}_f}$ provide three meta-features per category and $\vec{v}^{agglex}_{\vec{x}_f}$ provides only one meta-feature, generating the total of $1+3|\mathcal{C}|$ meta-features. Since they are generated using two similarity measures and two different datasets (original training dataset and tweets dataset), the final number of features driven from these vectors are $4(1+3|\mathcal{C}|)$.

In any case, the size of this meta-level feature set is much smaller than that typically found in bag-of-words representation, while explicitly capturing class discriminative information from the labeled set, polarity scores and external data.

4. EXPERIMENTAL EVALUATION

We start by presenting the experimental setup (Section 4.1) followed by the experimental results using distinct sets of meta-features (Section 4.2).

4.1 Experimental Setup

4.1.1 Textual Datasets

In order to evaluate the meta-feature strategies, we use a recent and publicly available benchmark [28] with nineteen real-world textual datasets gathered from different works. They are named aisopos_tw [13], debate [9], narr_tw [24], pappas_ted [26], pang_movie [25], sanders_tw[3], ss_bbc [30], ss_digg [30], ss_myspace [30], ss_rw [30], ss_twitter [30], ss-youtube [30], stanford_tw [14], semeval_tw[4], vader_amzn [19], vader_movie [19], vader_nyt [19], vader_tw [19] and yelp_review[5]. In addition to the previously described datasets, we also exploit the information of 1.6 millions of tweets automatically labeled by its emoticons [14]. All these datasets contain short texts as documents, which are distinct from documents usually found in text classification (e.g. news and web pages).

dataset	#msgs	#feat	density	#pos	#neg
aisopos_tw	278	1493	13.0	159	119
debate	1979	3360	11.5	730	1249
narr_tw	1227	3508	11.3	739	488
pappas_ted	727	1635	11.7	318	409
pang_movie	10662	12432	13.9	5331	5331
sanders_tw	1091	3102	13.5	519	572
ss_bbc	752	5655	40.3	99	653
ss_digg	782	4015	22.2	210	572
ss_myspace	834	2639	14.7	702	132
ss_rw	705	4595	43.3	484	221
ss_twitter	2289	7777	13.8	1340	949
ss_youtube	2432	6275	12.2	1665	767
stanford_tw	359	1620	12.0	182	177
semeval_tw	3060	9087	16.4	2223	837
vader_amzn	3610	3678	11.9	2128	1482
vader_movie	10568	11980	14.0	5242	5326
vader_nyt	4946	8756	13.0	2204	2742
vader_tw	4196	7346	11.2	2897	1299
yelp_review	5000	19398	71.5	2500	2500

Table 2: Dataset caracteristics

For all datasets, we performed a traditional preprocessing task: we removed stopwords, using the standard SMART list and the standard Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme for our Bag-of-Words (BoW) representation. As previously mentioned, we consider only messages with a clear polarity (positive or negative), since our focus is on this binary classification task.

Table 2 shows some of the characteristics of the datasets used to evaluate our meta-features. The first column indicates the name of the dataset used in our experiments, the second column is the number of messages in the dataset, the third shows the number of features (words) represented in the dataset, the fourth corresponds to the average number of words (density) of a message, and the last two columns show the number of positive and negative messages, respectively.

4.1.2 Evaluation, Algorithms and Procedures

The meta-level features were compared using the MicroF$_1$ score, which is a standard text categorization measure[6]. In order to compute this measure, the system-made decisions on test set $\mathcal{D}$ with respect to a specific category must be divided into three groups: True Positives (TP), False Positives (FP) and False Negatives (FN), respectively. The terms positive and negative refer to the classifier's prediction, and the terms true and false refer to whether that prediction corresponds to the external judgment. The measure

[3]http://www.sananalytics.com/lab/twitter-sentiment
[4]https://www.cs.york.ac.uk/semeval-2013/task2
[5]http://www.yelp.com/dataset_challenge
[6]Results with MacroF$_1$ are very similar and are omitted for space reasons.

is described as:

$$\mathrm{F}_1 = \frac{2TP}{2TP + FP + FN}$$

All experiments were executed using a 5-fold cross-validation procedure. The parameters were set via cross-validation within the training set, and the effectiveness of the algorithms running with distinct types of features were measured in the test partition. In other words, all results correspond to the effectiveness average in the 5 test partitions.

In order to evaluate the performance of different groups of features, we adopted the LIBLINEAR [11] implementation of the SVM classifier. The regularization parameter was chosen among eleven values from 2^{-5} to 2^{15} by using 5-fold cross-validation within each training dataset. The other parameter necessary to execute the experiments is the size of neighborhood used in kNN-based meta-level features. The neighborhood size k was chosen among ten values from 10 to 100 by also using 5-fold cross-validation within each training dataset. The parameters b and k_1 of the BM25 similarity score used in kNN were set to the default values 0.75 and 2 [23], respectively.

We use three recent and freely available lexicon-based classifiers Vader [19], SentiStrength [31] and SentiWordnet [2] to estimate the sentiment value of each message. Specifically, we extract the positive and negative sentiment scores of each message using these methods, along with combined and neutral scores given by Vader.

All experiments were run on a Quad-Core Intel® Xeon® E5620, running at 2.4GHz, with 16Gb RAM. The meta-features were generated using our GPU-based implementation of kNN[7]. It took only a few hours to generate all kNN-based meta-features using the mid-range NVIDIA 740 GPU, with 2Gb RAM. To compare the average results on our 5-fold cross-validation experiments, we assess the statistical significance of our results by means of a paired t-test with 95% confidence and Bonferroni correction to account for multiple tests. This test assures that the best results, marked in **bold**, are statistically superior to others. The obtained results (and their 95% confidence intervals) are described in Section 4.2. We also count the number of datasets in which a particular group of features is among the best (win count). Specifically, if one approach is statistically superior or tied to others in one dataset, it will be considered a winning approach and will receive one additional win count point. If another approach is statistically tied to a winning approach, then it will also receive an additional win count point.

4.2 Experimental Results

In this section we present results for a series of experiments to evaluate the effectiveness of meta-features. Initially, we present results comparing the effectiveness of our solution with the traditional Bag of Words (BoW) representation. Next, we compare the proposed meta-features with the state-of-the-art meta-features for text classification. Then, we compare the results of the best lexicon-based methods and a supervised ensemble of lexicon-based methods with our meta-features. Finally, we present a study of each individual group of proposed meta-features.

[7]Source code is available under GNU Public License (GPL) at http://purl.oclc.org/NET/gtknn/.

4.2.1 Proposed Meta Features versus Bag-of-Words

We start by demonstrating the effectiveness of the proposed meta-features compared to the set of original features (BoW) when both are used to train SVM classifiers and when BoW is used to train the kNN classifier. As Table 3 shows, the proposed meta-features are consistently among the best in all the nineteen datasets, while BoW only achieves the best results on eight and six datasets using, respectively, the SVM and kNN classifiers. Surprisingly, the kNN classifier performs almost as good as the state-of-art SVM classifier, which indicates the importance of the neighborhood in the sentiment analysis context.

The proposed meta-features performed better than the BoW in eleven out of the nineteen datasets. The most significant gains were up to 16%, 14%, 9%, 7%, 6% and 5.7%, respectively obtained in vader_tw, ss_twitter, narr_tw, semeval_tw, ss_youtube and ss_digg. These results justify meta-features as a replacement for the original high dimensional feature space.

dataset	Proposed	BoW (SVM)	BoW (kNN)
aisopos_tw	**89.2 ± 5.4**	**89.6 ± 2.8**	**82.8 ± 7.0**
debate	**80.0 ± 2.9**	77.4 ± 1.3	76.7 ± 1.6
narr_tw	**88.8 ± 2.0**	81.4 ± 2.7	80.3 ± 3.0
pappas_ted	**72.8 ± 3.0**	**76.1 ± 5.8**	**77.8 ± 4.3**
pang_movie	**78.6 ± 1.0**	76.9 ± 1.3	76.9 ± 1.1
sanders_tw	**86.5 ± 2.4**	**84.5 ± 3.1**	**82.3 ± 4.8**
ss_bbc	**88.6 ± 3.8**	**87.4 ± 5.6**	**87.4 ± 6.0**
ss_digg	**82.1 ± 2.6**	77.6 ± 3.7	78.1 ± 2.6
ss_myspace	**88.4 ± 1.2**	**86.2 ± 2.8**	**85.5 ± 2.4**
ss_rw	**79.8 ± 5.0**	**76.0 ± 4.1**	71.8 ± 2.1
ss_twitter	**82.6 ± 1.1**	72.8 ± 2.8	71.0 ± 0.9
ss_youtube	**86.1 ± 1.6**	81.2 ± 2.5	77.5 ± 1.1
stanford_tw	**86.9 ± 3.5**	**84.7 ± 3.0**	**83.0 ± 4.9**
semeval_tw	**85.8 ± 1.9**	80.2 ± 1.6	76.0 ± 1.2
vader_amzn	**78.0 ± 1.0**	74.1 ± 1.4	72.0 ± 2.4
vader_movie	**79.9 ± 0.6**	78.2 ± 0.6	77.8 ± 1.0
vader_nyt	**71.2 ± 2.5**	67.1 ± 1.2	65.1 ± 3.4
vader_tw	**97.2 ± 0.6**	84.0 ± 1.1	81.8 ± 0.8
yelp_review	**93.4 ± 1.1**	**93.3 ± 0.4**	83.9 ± 2.0
win count	19	8	6

Table 3: Average $\mathbf{F}_1$ with the proposed meta-features, and the bag-of-words performing with SVM and kNN classifiers.

4.2.2 Proposed versus State-of-art Meta-level Features

We now compare the relative effectiveness of our proposed meta-features regarding the state-of-art literature meta-level features for general text classification. As shown in Table 4, the set of proposed meta-features for sentiment analysis is consistently among the best in all nineteen compared datasets. The second best meta-feature group, was capable of being among the best in only six datasets. This is a strong evidence towards the robustness of the proposed meta-features for the specific task of sentiment analysis of short messages. Notice particularly, the significant improvements obtained by the proposed meta-features on vader_tw, ss_twitter, vader_nyt, narr_tw and aisopos_tw with gains ranging from 8% to 13% over the best baseline.

The reasons for the low competitiveness of the baseline meta-features are twofold: (i) most sentiment analysis datasets have only a few documents with very short messages and (ii) they do not exploit any specific characteristic of sentiment analysis. The proposed meta-features address these

dataset	Proposed	Gopal et al [17]	Canuto et al [7]
aisopos_tw	**89.2 ± 5.4**	71.2 ± 6.3	82.4 ± 4.6
debate	**80.0 ± 2.9**	**78.2 ± 1.7**	76.6 ± 1.6
narr_tw	**88.8 ± 2.0**	82.2 ± 1.0	81.0 ± 2.9
pappas_ted	**72.8 ± 3.0**	**67.0 ± 10.0**	**76.6 ± 3.8**
pang_movie	**78.6 ± 1.0**	77.3 ± 0.9	77.5 ± 1.4
sanders_tw	**86.5 ± 2.4**	83.2 ± 3.2	83.4 ± 1.7
ss_bbc	**88.6 ± 3.8**	**86.1 ± 5.3**	**86.3 ± 6.3**
ss_digg	**82.1 ± 2.6**	**76.3 ± 5.2**	**78.5 ± 3.8**
ss_myspace	**88.4 ± 1.2**	**86.2 ± 4.0**	**85.7 ± 3.2**
ss_rw	**79.8 ± 5.0**	**75.2 ± 2.3**	70.9 ± 1.0
ss_twitter	**82.6 ± 1.1**	73.4 ± 2.2	73.1 ± 1.4
ss_youtube	**86.1 ± 1.6**	79.8 ± 1.7	80.2 ± 1.9
stanford_tw	**86.9 ± 3.5**	80.5 ± 4.5	**84.4 ± 3.4**
semeval_tw	**85.8 ± 1.9**	79.1 ± 1.2	71.6 ± 1.1
vader_amzn	**78.0 ± 1.0**	74.2 ± 2.2	74.0 ± 1.6
vader_movie	**79.9 ± 0.6**	78.8 ± 1.0	78.8 ± 1.0
vader_nyt	**71.2 ± 2.5**	66.1 ± 1.9	66.4 ± 2.8
vader_tw	**97.2 ± 0.6**	85.6 ± 0.8	84.2 ± 1.1
yelp_review	**93.4 ± 1.1**	87.3 ± 1.5	90.0 ± 1.3
win count	19	6	5

Table 4: Average F_1 with different groups of meta-features. Canuto et al's MF are the state-of-art kNN based meta-features.

dataset	Proposed	Lex Ensem	Best Lex
aisopos_tw	**89.2 ± 5.4**	85.2 ± 5.9	83.4 ± 6.8
debate	**80.0 ± 2.9**	70.1 ± 1.2	67.4 ± 3.3
narr_tw	**88.8 ± 2.0**	85.8 ± 2.4	81.2 ± 1.9
pappas_ted	**72.8 ± 3.0**	**72.6 ± 7.0**	**72.8 ± 3.2**
pang_movie	**78.6 ± 1.0**	68.4 ± 1.3	63.7 ± 1.0
sanders_tw	**86.5 ± 2.4**	75.5 ± 2.2	73.1 ± 3.6
ss_bbc	**88.6 ± 3.8**	**88.6 ± 4.6**	84.7 ± 1.8
ss_digg	**82.1 ± 2.6**	80.5 ± 5.4	77.2 ± 3.4
ss_myspace	**88.4 ± 1.2**	87.3 ± 1.5	82.6 ± 1.8
ss_rw	**79.8 ± 5.0**	**78.2 ± 5.6**	**78.0 ± 3.8**
ss_twitter	**82.6 ± 1.1**	79.5 ± 2.6	75.8 ± 2.9
ss_youtube	**86.1 ± 1.6**	83.0 ± 3.4	78.6 ± 1.0
stanford_tw	**86.9 ± 3.5**	79.4 ± 6.9	78.5 ± 3.2
semeval_tw	**85.8 ± 1.9**	82.9 ± 1.5	77.9 ± 1.1
vader_amzn	**78.0 ± 1.0**	71.4 ± 2.2	68.5 ± 1.6
vader_movie	**79.9 ± 0.6**	68.8 ± 1.5	64.2 ± 1.0
vader_nyt	**71.2 ± 2.5**	67.5 ± 1.6	65.4 ± 2.1
vader_tw	**97.2 ± 0.6**	**97.3 ± 0.6**	95.0 ± 1.6
yelp_review	**93.4 ± 1.1**	88.8 ± 2.0	82.7 ± 0.9
win count	19	4	2

Table 5: Average F_1 with the proposed meta-features, a simplified ensemble of the lexical-based outputs and best result of an individual lexical method.

issues by: (i) using an external tweet dataset with 1.6 millions of documents; (ii) utilizing a specific similarity score for short queries (BM25), and (iii) by exploiting lexicon-based methods which deal with sentiment analysis specificities.

4.2.3 Proposed versus Lexical Approaches

We now compare the relative effectiveness of our proposed meta-features in comparison to a simplified ensemble using lexical-based methods [2, 19, 31] as first level classifiers, in which their outputs are combined with the SVM classifier. We also use the individual classification of the best unsupervised lexical method as baseline. As show in Table 5, our approach outperformed by large margins both, the lexical ensemble and the best unsupervised method, in various situations, with gains over the best baseline ranging from 9% to 16% in six datasets, namely vader_movie, pang_movie, debate, sanders_tw, stanford_tw and vader_amzn. The most significant gains were on our two biggest datasets vader_movie and pang_movie, with 16% and 15% respectively, which is an evidence that the proposed meta-features can extract more discriminative information from large training datasets.

The results of the best lexical unsupervised method (last column of Table 5) are tied to the supervised methods in only two datasets: pappas_ted and ss_rw, some of the smallest ones. However, the best unsupervised approach could achieve results superior to 75% of effectiveness in twelve of the nineteen datasets, a significant fact, as it does not use any domain-specific labeled information to classify messages. This demonstrates why and how this type of source can produce useful and supplementary information to be exploited in other approaches. Finally, exploiting the lexical approaches in a supervised manner (the Lexical ensemble) helped to improve results substantially in the most datasets, achieving gains about 7% on pang_movie, vader_movie, semeval_tw and yelp_review. This further justifies the use of such information in our meta-features.

4.2.4 Similarity Measure for the Proposed Meta-level Features

In this experiment, we investigate the role and the impact of the two used similarity functions – cosine and BM25 – in our results. Table 6 shows that using BM25 to deal with the similarity of short messages is in fact better than using the cosine similarity, since it is consistently better or tied with the cosine results. In all datasets, BM25 never performs worse than cosine, with small but statistically significant gains up to 2% in six datasets. It is important to point out that the these statistically significant gains are present only in datasets with relatively small density.

dataset	COSINE	BM25
aisopos_tw	**88.2 ± 6.5**	**90.0 ± 6.9**
debate	79.0 ± 2.8	**80.6 ± 2.6**
narr_tw	87.5 ± 2.7	**89.4 ± 2.1**
pappas_ted	**73.6 ± 5.7**	**74.8 ± 5.3**
pang_movie	**78.2 ± 1.1**	**78.1 ± 0.6**
sanders_tw	**86.4 ± 1.7**	**86.1 ± 3.3**
ss_bbc	**88.3 ± 4.7**	**87.4 ± 5.9**
ss_digg	**82.0 ± 2.5**	**81.0 ± 5.1**
ss_myspace	**88.7 ± 1.2**	**86.3 ± 4.3**
ss_rw	**80.4 ± 5.4**	**79.0 ± 3.5**
ss_twitter	82.1 ± 1.3	**83.0 ± 0.8**
ss_youtube	85.1 ± 1.6	**86.7 ± 1.3**
stanford_tw	**88.3 ± 6.5**	**87.5 ± 3.7**
semeval_tw	**85.8 ± 1.0**	**85.9 ± 2.4**
vader_amzn	77.3 ± 1.3	**78.2 ± 1.1**
vader_movie	**79.4 ± 1.1**	**79.5 ± 0.6**
vader_nyt	**71.6 ± 1.8**	**71.8 ± 1.7**
vader_tw	96.9 ± 0.7	**97.4 ± 0.6**
yelp_review	**92.6 ± 0.9**	**92.9 ± 0.9**
win count	13	19

Table 6: Average F_1 of the proposed meta-features generated using only cosine or BM25 as similarity scores.

4.2.5 Analysis of the Proposed Groups of Meta Features

In this section we investigate the quality of each proposed group of meta-level features in isolation as well as the supplementary information they bring to other groups. Table 7 shows the effectiveness of the SVM classifier trained with each group of the proposed meta-features in isolation. As can be seen, all groups achieved relatively good results in iso-

dataset	RAWLEX	RAWSIM	KNNLEX	TWEMOT
aisopos_tw	**85.2 ± 5.9**	**86.4 ± 2.4**	**83.9 ± 8.3**	**83.8 ± 5.7**
debate	70.1 ± 1.2	**79.0 ± 3.0**	**78.7 ± 2.0**	69.0 ± 1.4
narr_tw	**85.8 ± 2.4**	**82.2 ± 3.4**	**81.2 ± 2.8**	**85.3 ± 2.1**
pappas_ted	**72.6 ± 7.0**	**76.5 ± 2.5**	**65.5 ± 4.3**	**74.8 ± 6.0**
pang_movie	68.4 ± 1.3	**77.5 ± 0.7**	**77.1 ± 0.4**	65.5 ± 1.2
sanders_tw	75.5 ± 2.2	**82.7 ± 3.1**	**83.5 ± 3.5**	74.6 ± 2.8
ss_bbc	**88.6 ± 4.6**	**86.5 ± 6.1**	**86.7 ± 5.8**	**87.7 ± 5.9**
ss_digg	**80.5 ± 5.4**	**78.3 ± 2.4**	**74.9 ± 2.7**	**77.1 ± 1.8**
ss_myspace	**87.3 ± 1.5**	**85.6 ± 2.1**	**85.4 ± 3.5**	**87.2 ± 3.2**
ss_rw	**78.2 ± 5.6**	**72.6 ± 2.8**	**72.8 ± 2.4**	**71.4 ± 4.1**
ss_twitter	**79.5 ± 2.6**	73.7 ± 2.7	72.6 ± 2.0	**78.2 ± 2.6**
ss_youtube	**83.0 ± 3.4**	**78.8 ± 1.3**	**78.1 ± 2.0**	**81.2 ± 1.3**
stanford_tw	**79.4 ± 6.9**	**81.1 ± 3.9**	**83.3 ± 6.4**	**80.5 ± 8.0**
semeval_tw	**82.9 ± 1.5**	78.0 ± 1.3	77.3 ± 2.0	**80.8 ± 2.5**
vader_amzn	**71.4 ± 2.2**	**73.4 ± 1.3**	**75.0 ± 2.4**	66.3 ± 1.1
vader_movie	68.8 ± 1.5	**78.3 ± 0.6**	**78.6 ± 0.9**	66.7 ± 1.3
vader_nyt	**67.5 ± 1.6**	**66.9 ± 2.4**	**66.4 ± 2.8**	**66.0 ± 1.4**
vader_tw	**97.3 ± 0.6**	84.7 ± 1.5	84.1 ± 1.2	89.1 ± 0.7
yelp_review	88.8 ± 2.0	**91.1 ± 0.6**	87.9 ± 0.9	79.8 ± 1.3
win count	15	16	15	12

Table 7: Average F_1 of each proposed group of meta-features in isolation.

dataset	ALL	no RAWLEX	no RAWSIM	no KNNLEX	no TWEMOT
aisopos_tw	89.2 ± 5.4	86.4 ± 6.3	86.0 ± 6.4 ↓	88.9 ± 6.7	90.7 ± 3.6
debate	80.0 ± 2.9	78.8 ± 2.4 ↓	79.9 ± 2.9	79.5 ± 2.7	80.2 ± 2.9
narr_tw	88.8 ± 2.0	86.1 ± 1.3 ↓	88.8 ± 2.4	89.1 ± 2.2	88.0 ± 1.5
pappas_ted	72.8 ± 3.0	72.5 ± 1.6	76.6 ± 5.5	69.6 ± 6.8	72.9 ± 3.3
pang_movie	78.6 ± 1.0	78.0 ± 1.0 ↓	78.2 ± 0.8	78.3 ± 0.9	78.5 ± 1.0
sanders_tw	86.5 ± 2.4	84.7 ± 2.4 ↓	85.8 ± 2.3 ↓	85.6 ± 2.6	86.2 ± 3.1
ss_bbc	88.6 ± 3.8	88.3 ± 5.3	87.9 ± 4.1	88.6 ± 4.7	86.7 ± 4.7 ↓
ss_digg	82.1 ± 2.6	80.1 ± 1.3 ↓	81.5 ± 2.8	81.8 ± 3.0	81.6 ± 3.3
ss_myspace	88.4 ± 1.2	86.2 ± 1.4 ↓	88.7 ± 1.5	89.0 ± 1.5	87.9 ± 2.7
ss_rw	79.8 ± 5.0	74.8 ± 5.1 ↓	79.7 ± 5.6	80.2 ± 4.8	80.4 ± 4.4
ss_twitter	82.6 ± 1.1	77.8 ± 3.1 ↓	83.0 ± 1.5	82.7 ± 1.0	80.9 ± 2.2 ↓
ss_youtube	86.1 ± 1.6	83.0 ± 0.9 ↓	85.8 ± 2.0 ↓	86.4 ± 1.8	84.7 ± 1.9 ↓
stanford_tw	86.9 ± 3.5	83.9 ± 5.6	86.3 ± 4.0	86.1 ± 3.0 ↓	84.7 ± 5.1 ↓
semeval_tw	85.8 ± 1.9	82.0 ± 1.2 ↓	85.8 ± 1.7	85.6 ± 1.7	85.3 ± 0.5
vader_amzn	78.0 ± 1.0	75.3 ± 2.0 ↓	76.9 ± 1.4 ↓	77.3 ± 1.3	77.5 ± 2.8
vader_movie	79.9 ± 0.6	79.3 ± 0.9 ↓	79.1 ± 1.1 ↓	79.1 ± 1.0 ↓	79.8 ± 0.6
vader_nyt	71.2 ± 2.5	69.5 ± 1.3 ↓	71.4 ± 2.0	71.2 ± 2.1	70.9 ± 2.6
vader_tw	97.2 ± 0.6	89.6 ± 1.3 ↓	97.2 ± 0.7	97.2 ± 0.7	97.3 ± 0.5
yelp_review	93.4 ± 1.1	91.1 ± 0.7 ↓	93.0 ± 1.1 ↓	92.3 ± 1.0 ↓	93.0 ± 1.1

Table 8: Average F_1 removing one group of meta-features from the full set. ↓ indicates statistically significant losses due to the removal of a meta-feature group from the full set ALL.

lation. In fact, there are ten datasets in which all the groups are statistically tied. Despite this, there is no isolated group that is consistently among the best in all datasets, and there are only two datasets in which one specific group is better than all remaining ones.

TWEMOT achieves the worst performance among the four groups, being only twelve times among the best individual groups. This is expected, since TWEMOT exploits primarily the evidence from an external dataset of tweets, labeled with emoticons. It is interesting to notice how TWEMOT and RAWLEX have similar results in most datasets. Similarly, KNNLEX and RAWSIM also share some close results. This may be due the fact that both TWEMOT and RAWLEX try to exploit external sources of information to classify messages, but KNNLEX and RAWSIM are focused primarily on exploiting the neighborhood inside the training data.

In order to further analyze the effects of possibly noisy meta-features and the supplementary information provided by the groups, we performed a new set of experiments in which we removed one group of meta-features from the full set ALL before training the SVM classifier. Table 8 shows the effects of such procedure.

The first observation is that the removal of the meta-feature group RAWLEX produced significant losses with regards to the set with ALL meta-features in most datasets. This is an evidence supporting the supplementary discriminative evidence provided by the output scores of lexical methods. This was somewhat expected, since these methods provide specific additional information about lexical clues on the message.

The second most impacting group is RAWSIM, which provides the similarity information between a message and their neighborhood. The removal of this group causes statistically significant losses in six datasets, demonstrating the supplementary nature of pure similarity scores. In three datasets, KNNLEX is capable of providing supplementary information by combining similarity scores with lexical evidence in a message's neighborhood. Although it does not produce high improvements, since RAWLEX, RAWSIM and TWEMOT already provide some of this information, the statistical significant losses demonstrate that there is additional information that can be extracted from this group in some datasets.

The removal of the TWEMOT group also produces significant losses in four datasets. This is an evidence towards the importance of the additional information exploited by

the large (and noisy) tweet dataset. As we can see, the removal of this group does not improve the effectiveness in any dataset. This is surprising – TWEMOT is capable of capturing useful classification evidence from a highly noisy dataset for some datasets, without harming the remaining ones.

As we can see, the removal of any single group does have positive impacts on the effectiveness in several datasets. This is evidence that the proposed meta-features are not inserting noise and can be included in the pool of available evidence in order to exploit additional information from the data in the hard task of sentiment analysis of short messages.

4.2.6 Importance of Groups with using $2^k r$ Factorial Design

The best way to evaluate the interactions among our four groups of meta-features is by doing an analysis of all 2^k possible combinations of groups for each dataset[8]. By replicating the experiments with each possible combination (using 5-fold cross validation), we can also evaluate the effects of uncontrollable external factors. We follow the standard quantitative approach called $2^k r$ factorial design [20] to analyze the effects of the individual groups of meta-features, as well as the effectiveness improvements produced by their interactions.

The first step to perform a $2^k r$ factorial design is to define the binary factors that may affect a response variable (e.g., MicroF$_1$ score). In our case, each factor corresponds to the presence or absence of one group of meta-features. We name the presence of each group of features as follows:

- A – Presence of KNNLEX
- B – Presence of RAWSIM
- C – Presence of TWEMOT
- D – Presence of RAWLEX

In order to summarize the analysis of the nineteen datasets with the sixteen possible combinations of our four factors, we clustered our datasets into two groups, according to the importance and interaction of meta-features in each dataset. We hope to keep similar datasets in the same group in order to summarize the results of the group. We use a simple k-means algorithm to group close datasets according to their individual factorial design results. The found groups are:

- **Group 1 datasets:** debate, pang_movie, sanders_tw, vader_amzn, vader_movie and yelp_review.
- **Group 2 datasets:** aisopos_tw, narr_tw, pappas_ted, ss_bbc, ss_digg, ss_myspace, ss_rw, ss_twitter, ss_youtube, stanford_tw, semeval_tw, vader_nyt and vader_tw.

For each dataset, we compute the percentage of variation in the results that can be explained by each individual group of meta-features and by the interaction between groups of meta-features considering each possible combination. We summarize the effects of each combination[9] on different datasets by showing its average. Table 9 shows this summarization for the two groups of datasets. In addition to the mean value, we also show the lowest and the highest value among the datasets in each group.

[8]We consider the case without any group using a "random" classifier, which returns positive with a 50% chance.

[9]The 95% confidence interval on the effects of all combinations of the tested datasets are always inferior to 1%.

Factor/ Interaction	Group 1 datasets		Group 2 datasets	
	Mean	Range	Mean	Range
A	19.5	[14.4, 24.3]	6.1	[3.8, 10.1]
B	19.8	[10.9, 26.3]	5.7	[1.1, 9.3]
C	3.3	[2.2, 5.5]	9.5	[3.2, 16.2]
D	9.6	[4.6, 17.2]	20.7	[7.3, 39.5]
A:B	17.3	[10.3, 22.4]	5.6	[3.6, 9.1]
A:C	2.6	[1.6, 3.8]	4.8	[2.8, 7.4]
B:C	2.6	[1.8, 4.2]	4.0	[2.2, 6.2]
A:D	3.9	[2.4, 8.1]	4.5	[3.2, 8.0]
B:D	3.8	[2.8, 6.3]	3.6	[1.6, 5.3]
C:D	2.2	[1.4, 3.2]	6.8	[3.8, 9.4]
A:B:C	2.9	[1.9, 4.2]	4.6	[2.6, 10.0]
A:B:D	3.9	[2.6, 6.3]	3.8	[2.4, 5.6]
A:C:D	1.9	[1.4, 3.6]	4.0	[1.4, 6.0]
B:C:D	1.8	[1.2, 2.8]	3.3	[0.4, 6.2]
A:B:C:D	1.9	[1.3, 2.9]	3.0	[0.6, 5.6]
Residuals	2.8	[0.7, 5.7]	9.9	[0.6, 30.4]

Table 9: Explained percentage of result variation by individual meta-feature groups and interactions between them.

As we can see, the variation on results in the datasets of group 1 can mostly be explained by the presence of A (KNNLEX) and B (RAWSIM) in isolation. The inclusion of each one of these groups account for about 20% of all the variation in the observed results. The interaction between A:B is the third most important observed factor. In fact, the effects of A, B and A:B account for 60% of all the MicroF$_1$ variation. The fourth most important factor is the inclusion of D, which accounts for about 10% of the variation, showing the importance of the RAWLEX meta-features in isolation. The remaining effects, though small, account together for about 30% of the total variation in the results, which highlights the supplementarity among the proposed groups of meta-features.

Regarding the datasets in group 2, the presence of D (RAW-LEX) in isolation accounts for 20% of all the MicroF$_1$ variations in the experiments. The presence of C (TWEMOT) in isolation and the interaction C:D are the most important effects to explain the results. This means that the use of additional information from the lexical-based approaches and from the large tweet corpus are the most important factors for these datasets. It is important to notice that A, B and most of the interactions have a considerable participation to explain the results. The residuals (inexplicable fraction of the variation in the results) are quite high (about 10%) on the second group. This may be due the fact that most of the datasets on the second group are small, which leads to higher variability when classifying samples from them. Since they are more likely to suffer from overfitting due to shortage of training information, their dependence on external data (provided by RAWLEX and TWEMOT) is expected.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed new meta-level features for sentiment analysis, especially for the case of short messages. We experimentally compared them with (i) the original features (bag-of-words); (ii) previously proposed state-of-the-art meta-features found in the literature for automatic text classification; (iii) state-of-the-art lexicon-based sentiment analysis methods; and (iV) supervised ensembles of lexicon-based methods.

Our controlled experiments with these new meta features on nineteen benchmark datasets showed significant effectiveness improvements in most datasets over all baselines. In fact, our proposed approach was the best one in all tested datasets and scenarios, with no other approach getting close to this result. This provides strong evidence towards the benefits of using the more compact and informative space of features proposed in this paper to replace highly dimensional bag-of-words representation. By introducing meta-level features that effectively and efficiently capture important information from highly noisy external domains while keeping the standard formulation of the automated classification problem, we enable our solution to be applied in a number of different sentiment analysis scenarios.

We hope this study provides useful insights into how to enhance the performance of sentiment analysis by improving the representation schemes for instances, categories and their relationships. A line of future research would be to explore our meta features with other classification algorithms and feature selection techniques in different sentiment analysis tasks, such as scoring movies or products according to their related reviews.

6. ACKNOWLEDGMENTS

This work was partially supported by the Brazilian government through CNPq, CAPES and FAPEMIG.

7. REFERENCES

[1] M. Araújo, P. Gonçalves, F. Benevenuto, and M. Cha. ifeel: A system that compares and combines sentiment analysis methods. In *WWW'14*, 2014.

[2] S. Baccianella, A. Esuli, and F. Sebastiani. Senti wordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC'10*, 2010.

[3] C. Banea, R. Mihalcea, J. Wiebe, and S. Hassan. Multilingual subjectivity analysis using machine translation. In *EMNLP'08*, pages 127–135, Stroudsburg, PA, USA, 2008.

[4] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *COLING '10*, pages 36–44, Stroudsburg, PA, USA, 2010.

[5] P. Biyani, S. Bhatia, and C. Caragea. Using non-lexical features for identifying factual and opinionative threads in online forums. *Knowledge-Based Systems*, 69(0):170 – 178, 2014.

[6] S. Canuto, G. Marcos, W. Santos, T. Rosa, and M. Wellington. Efficient and scalable metafeature-based document classification using massively parallel computing. In *Proc. SIGIR*, 2015.

[7] S. Canuto, T. Salles, M. A. Gonçalves, L. Rocha, G. Ramos, L. Gonçalves, T. Rosa, and W. Martins. On efficient meta-level features for effective text classification. In *CIKM'14*, pages 1709–1718, 2014.

[8] N. F. da Silva, E. R. Hruschka, and E. R. H. Jr. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, 66(0):170 – 179, 2014.

[9] N. A. Diakopoulos and D. A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *SIGCHI'10*, pages 1195–1198. ACM, 2010.

[10] S. Dzeroski and B. Zenko. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273, 2004.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[12] R. Feldman. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89, Apr. 2013.

[13] Fotis Aisopos. Manually annotated sentiment analysis twitter dataset ntua. www.grid.ece.ntua.gr, 2014.

[14] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Technical report, Stanford, December 2009.

[15] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha. Comparing and combining sentiment analysis methods. In *COSN'13*, 2013.

[16] P. Gonçalves, D. H. Dalip, H. Costa, M. A. Gonçalves, and F. Benevenuto. On the combination of "off-the-shelf" sentiment analysis methods. In *SAC'16*. ACM, 2016.

[17] S. Gopal and Y. Yang. Multilabel classification with meta-level features. In *SIGIR'10*, 2010.

[18] A. Hassan, A. Abbasi, and D. Zeng. Twitter sentiment analysis: A bootstrap ensemble framework. In *SocialCom'13*, pages 357–364. IEEE, 2013.

[19] C. J. Hutto and E. Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM'14*, 2014.

[20] R. Jain. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.* Wiley professional computing. Wiley, 1991.

[21] A. Kyriakopoulou and T. Kalamboukis. Using clustering to enhance text classification. In *SIGIR'07*, pages 805–806, 2007.

[22] A. Kyriakopoulou and T. Kalamboukis. Combining Clustering with Classification for Spam Detection in Social Bookmarking Systems. RSDC '08, 2008.

[23] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge Press, NY, USA, 2008.

[24] S. Narr, M. Hulfenhaus, and S. Albayrak. Language-independent twitter sentiment analysis. *Knowledge Discovery and Machine Learning (KDML)*, pages 12–14, 2012.

[25] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL'04*, Stroudsburg, PA, USA, 2004.

[26] N. Pappas and A. Popescu-Belis. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In *SIGIR'13*, pages 773–776. ACM, 2013.

[27] B. Raskutti, H. L. Ferrá, and A. Kowalczyk. Using unlabelled data for text classification through addition of cluster parameters. In *ICML'02*, 2002.

[28] F. Ribeiro, M. Araújo, P. Gonçalves, F. Benevenuto, and M. A. Gonçalves. A benchmark comparison of state-of-the-practice sentiment analysis methods. *arXiv preprint arXiv:1512.01818*, 2015.

[29] F. Su and K. Markert. From words to senses: A case study of subjectivity recognition. In *COLING '08*, pages 825–832, Stroudsburg, PA, USA, 2008.

[30] M. Thelwall. Heart and soul: Sentiment strength detection in the social web with sentistrength. http://sentistrength.wlv.ac.uk/documentation/SentiStrengthChapter.pdf, 2013.

[31] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, Dec. 2010.

[32] K. M. Ting and I. H. Witten. Issues in stacked generalization. *J. Artif. Int. Res.*, 10(1):271–289, 1999.

[33] X. Wang, C. Zhang, and M. Wu. Sentiment classification analysis of chinese microblog network. In G. Mangioni, F. Simini, S. M. Uzzo, and D. Wang, editors, *Complex Networks VI*, volume 597 of *Studies in Computational Intelligence*, pages 123–129. Springer, 2015.

[34] J. Wiebe and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *CICLing'05*, pages 486–497, Berlin, Heidelberg, 2005. Springer-Verlag.

[35] Y. Yang and S. Gopal. Multilabel classification with meta-level features in a learning-to-rank framework. *Mach. Learn.*, 88:47–68, 2012.