Content vs. Context for Sentiment Analysis: a Comparative Analysis over Microblogs

Fotis Aisopos[$], George Papadakis[$,◇], Konstantinos Tserpes[$], Theodora Varvarigou[$]
[◇] L3S Research Center, Germany papadakis@L3S.de
[$] ICCS, National Technical University of Athens, Greece {fotais, gpapadis, tserpes, dora}@mail.ntua.gr

ABSTRACT

Microblog content poses serious challenges to the applicability of traditional sentiment analysis and classification methods, due to its inherent characteristics. To tackle them, we introduce a method that relies on two orthogonal, but complementary sources of evidence: content-based features captured by n-gram graphs and context-based ones captured by polarity ratio. Both are language-neutral and noise-tolerant, guaranteeing high effectiveness and robustness in the settings we are considering. To ensure our approach can be integrated into practical applications with large volumes of data, we also aim at enhancing its time efficiency: we propose alternative sets of features with low extraction cost, explore dimensionality reduction and discretization techniques and experiment with multiple classification algorithms. We then evaluate our methods over a large, real-world data set extracted from Twitter, with the outcomes indicating significant improvements over the traditional techniques.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—Information filtering

Keywords

Sentiment Analysis, N-gram Graphs, Social Context

1. INTRODUCTION

The advent of the Web 2.0 and Social Media platforms led to an unprecedented increase in the volume of the user-generated content that is available on the Web [24]. One of the most popular services is microblogging, with Twitter[1] constituting the most successful application of this kind: it encompasses around 180 million users that post more than 1 billion messages per week[2]. A large portion of this content - if not its majority - is subjective, containing opinions and sentiments on various topics of interest [21]. Thus it includes valuable information for a number of tasks that range from product marketing to politics and policy making. To leverage this bulk of subjective information, automatic techniques are required for processing it; this need recently gave rise to *Sentiment Analysis* (**SA**), also known as *Opinion Mining* in the IR community [30]. The popularity of this field is reflected in the high number of on-line services offering sentiment extraction from Twitter messages, such as Twendz[3] and TweetFeel[4].

Existing SA systems typically aim at extracting sentiment-expressive textual patterns from unstructured documents. To this end, they employ either discriminative (series of) words [5] or dictionaries that assess the meaning and the lexical category of specific words and phrases (e.g., SentiWordNet[5]) [14, 21, 31]. Although these approaches are sufficiently effective in the context of specific settings (e.g., large curated documents), they are built on the assumption that the input documents are written in the particular language their methods are crafted for, not including noisy content and misspelled words. However, these fundamental assumptions are broken by the inherent characteristics of microblog content, which call for a *language-agnostic* SA approach that is tolerant to high levels of noise:

(i) *Sparsity.* Microblog posts solely comprise free-form text that is rather short in length (e.g., maximum 140 characters in Twitter). Due to their limited size, they typically consist of a few words, thus involving little extra information that can be used as evidence for identifying their polarity.

(ii) *Non-standard vocabulary.* Microblog posts are informal, as they are mainly exchanged between fellows who indulge in using slang words and non-standard expressions (e.g., "koo" instead of "cool") [6]. Also, the limited size of messages urges authors to shorten words into new forms that bear little similarity to the original one. For instance, "great" is replaced by "gr8" and "congratulations" by "congratz".

(iii) *Noise.* The real-time nature of microblogging encourages users to post their messages without verifying their correctness with respect to grammar or syntactic rules. In case a message (or part of it) is incomprehensible, the author can simply replace it with a new one. As a result, the user-generated, microblog content abounds in misspelled words and incorrect phrases, thus entailing high levels of noise.

(iv) *Multilinguality.* Although the majority of users stems from English-speaking countries, microblogging platforms are popular world-wide [13]; their user base encompasses

[1] http://twitter.com
[2] http://blog.kissmetrics.com/twitter-statistics/
[3] http://twendz.waggeneredstrom.com
[4] http://www.tweetfeel.com
[5] http://sentiwordnet.isti.cnr.it

HT'12, June 25–28, 2012, Milwaukee, Wisconsin, USA.

people talking in various languages and dialects, thus rendering inapplicable the language-specific SA methods.

In this paper, we introduce a novel approach for SA that relies on two *orthogonal*, yet *complementary* sources of evidence, both being language-neutral and robust to noise. The first one extracts reliable content-based features using the n-gram graphs document representation model. The second one considers the contextual information of individual messages in order to infer their sentiment without considering their content. It relies on social graph connections to capture the general mood expressed in the *social context* of each message: its author, her friends as well as the users closer to her (i.e., those friends that share higher levels of homophily with her). It also considers the general mood related to the resources contained in each message: its topic(s), the users it mentions as well as the media items it points to, information that is excluded from the content features. We thus compare between two distinct sources of evidence and analytically examine how they perform in conjunction.

In addition to effectiveness, we also pay attention to improving efficiency. We actually aim at identifying those classification settings that offer the best balance between effectiveness and efficiency. To this end, we investigate four possibilities: first, we propose alternative features with low extraction cost for both sources of evidence. Second, we experiment with attribute filtering approaches in order to reduce the feature space to its minimal subset that maintains the original levels of accuracy at a significantly lower processing time. Third, we propose discretization techniques that turn our numeric features into nominal ones, which involve higher classification efficiency [16]. Last but not least, we examine several classification algorithms of varying time complexity. We analytically examine the actual performance of all these classification settings, applying them on a large-scale, real-world data set of Twitter data.

On the whole, the main contributions of our paper can be summarized as follows:

- We distinguish between two orthogonal, yet complementary categories of SA features: the content-based and context-based ones. The former detects novel textual patterns in microblog messages, while the latter encapsulate the aggregate polarity of their social context.
- We examine the n-gram graphs performance in the context of SA over microblog content. We explain how their features can be discretized and compare their numeric and their nominal form with the traditional representation models. We also compare it with the alternative of exclusively considering specific punctuation features.
- We introduce Polarity Ratio as a novel metric encapsulating the aggregate sentiment of a document collection and explain how it can form the basis for context-based features. We apply it on several aspects of a document's social context and present an approach for discretizing its value. Also, we compare it with the efficient alternative of considering several direct contextual features.
- We apply our features to several state-of-the-art classification algorithms and evaluate their performance over a large, real-world data collection comprising 3 million Twitter messages.

The rest of the paper is structured as follows: Section 2 formally defines the problem we are tackling, while Section 3 presents the main characteristics of Twitter. We present our approach in Section 4 and in Section 5 we evaluate its performance over a thorough experimental study. Related work is discussed in Section 6, followed by our conclusions and future directions in Section 7.

2. PROBLEM FORMULATION

Sentiment Analysis is distinguished in three tasks [18]:

(i) *Document-level SA* assumes each document to express a single opinion about a particular topic or object,

(ii) *Sentence-level SA* splits each document into sentences, hypothesizing that they express individual opinions, and

(iii) *Feature-level SA* splits each document and sentence into polarized phrases that correspond to a particular feature of the discussed object or topic.

In this work, we exclusively focus on document-level SA in the context of microblog posts. In particular, we aim at detecting the polarity of individual Twitter messages, which typically consist of few sentences. Given their limited length, though, the problem we are tackling is very close to the Sentence-level SA, as well.

In practice, the task of document-level SA is typically cast as a binary classification problem, where the goal is to identify whether a document expresses a negative or a positive opinion [18]. Formally, it is defined as follows:

Problem 1 (Binary Polarity Classification). *Given a collection of documents $\mathcal{D}$ and the set of binary polarization classes $\mathcal{P}_\mathcal{B} = \{negative, positive\}$, the goal is to* approximate *the unknown target function $\Phi_\mathcal{B} : \mathcal{D} \rightarrow \mathcal{P}_\mathcal{B}$, which describes the documents' polarization according to a golden standard, by means of a function $\Phi'_\mathcal{B} : \mathcal{T} \rightarrow \mathcal{P}_\mathcal{B}$ that is called the* ***binary polarity classifier****.*

This formulation simplifies the task of SA, as it is based on the assumption that each document is subjective (i.e., it expresses a single, polarized opinion). In practice, however, a document can be neutral, as well, containing objective (i.e., factual) information. For this reason, we additionally consider the following, more general problem of document-level SA:

Problem 2 (General Polarity Classification). *Given a collection of documents $\mathcal{D}$ and the set of all polarization classes $\mathcal{P}_\mathcal{G} = \{negative, neutral, positive\}$, the goal is to approximate the unknown target function $\Phi_\mathcal{G} : \mathcal{D} \rightarrow \mathcal{P}_\mathcal{G}$, which describes the polarization of documents according to a golden standard, by means of a function $\Phi'_\mathcal{G} : \mathcal{T} \rightarrow \mathcal{P}_\mathcal{G}$ that is called the* ***general polarity classifier****.*

Note that both problems are modeled as *single-label classification* tasks (i.e., each document belongs to a single polarity class). Note also that some works address them in a slightly different manner [18]: given a set of documents, its elements are first categorized into a binary scale of objective (i.e., neutral) and subjective (i.e., polarized) ones; in a second stage, they further categorize the subjective documents into negative and positive ones. In this work, we consider the multiclass version of Problem 2 so as to compare its performance with that of Problem 1 on an equal basis (i.e., applying both of them on the same data). In this way, we provide a holistic overview of SA in real settings, and analytically examine the effect of extending Problem 1 with the class of objective documents.

3. PRELIMINARIES

Among all microblogging platforms, we selected Twitter for developing and testing our approach, due to the following advantages it conveys:

(i) *Strict interaction.* Twitter defines a single, strict way of interaction, allowing users to post only short messages of up to 140 characters - called *tweets*. To draw the attention of other users, tweets typically contain original, self-contained content that requires the minimum attention from readers. Thus, user sentiments are exclusively encapsulated in tweets, unlike other platforms that offer diverse ways for expressing them (e.g., the "Like" and the "+1" buttons in Facebook[6] and Google+[7], respectively).

(ii) *Social graph.* The morphology of Twitter's social graph captures the relationships between its users in an unequivocal way. More specifically, users can register to any account that is of particular interest to them in order to receive notification for its latest posts; the subscriber is called *follower*, the content provider is the *followee*, and their connection is modeled by a directed edge that points from the former to the latter. This allows for a particular category of interpersonal connections, namely the **reciprocal friends**; these are followers that are followed back by their followees, thus indicating a particularly close relationship between them (i.e., each one finds the other of particular interest). Their strong connection is typically interpreted as a sign of high levels of homophily, in the sense that they share a highly similar background, such as common age, sex or education [32].

(iii) *Public content.* The vast majority of Twitter's content is public and accessible, enabling us to harvest an adequate volume of content for experimentation.

(iv) *Timed activity.* Tweets are timestamped, thus indicating their sequence of appearance. As we explain below, this is critical for deriving past contextual evidence from the activity relevant to individual tweets.

In the following, we analyze the intrinsic characteristics of Twitter that lie at the core of our methods:

(i) *Hashtags.* Users typically categorize their tweets in topics that can be freely defined by any user. This is simply done by adding a topic tag - called *hashtag* - usually at the end of the tweet. To distinguish it from the rest of the message, a hashtag starts with the symbol #, which is then followed by one or more concatenated words or alphanumerics (e.g., #fb). This notation enables the efficient and effective identification of tweets pertaining to a specific topic (i.e., **topic tweets**). Note, though, that a single tweet can be associated with multiple hashtags.

(ii) *Mentions.* Twitter often serves as a platform for discussions among its members (i.e., chat). To facilitate this functionality, a user can address another person simply by adding a *mention* to her username. This is a special notation formed by concatenating the symbol @ at the beginning of the corresponding username (e.g., @*erwtokritos*). In this way, it is easy to identify and aggregate all the tweets pertaining to a particular user (i.e., **mention tweets**).

(iii) *External Pointers.* A common practice in Twitter is to inform one's followers about interesting Web resources (e.g., on-line videos), by posting the corresponding link. Given that URLs spread unchanged even among users that speak different languages, it is easy to track the messages pertaining to a specific resource (i.e., **URL tweets**).

(iv) *Emoticons.* Subjective tweets usually denote the opinion of their author with the help of standard "smileys": positive sentiments are usually marked with one of the following annotations: ":)", "(:", ":-)", "(-:", ":)", "(:", ":D" or "=)", whereas the negative ones are typically annotated by the following emoticons: ":(", "):", ":-(", ")-:", ": (" or ") :" [1, 5, 22]. We call **positive tweets** the messages annotated with at least one from the former group of emoticons, and **negative tweets** those annotated with at least one from the latter [1, 5, 22]; messages that belong to either of these categories are collectively called **polarized tweets**. **Neutral tweets**, on the other hand, are those lacking any polarity indicators. Tweets containing both positive and negative emoticons are entirely excluded from our analysis; the reason is that they are not suitable for the task of single-label, document-level SA we are considering, but rather for the feature-level one, which is out of the scope of this work.

(v) *Retweets.* Users typically share with their followers interesting tweets that have been posted by other users. To distinguish these messages from their own tweets, they mark them as *retweets*, adding the special annotation "*RT* @X" usually at their beginning. In this way, they give credit to the original author (i.e., the user X) and enable us to distinguish genuine tweets from the reproduced ones.

Of the above features, the first four offer valuable contextual information for individual tweets. Retweets are excluded from our analysis, as they do not provide any novel information.

4. APPROACH

In this section, we elaborate on the main techniques that provide the textual and contextual features of our approach. We accompany them with discretization techniques for enhancing their efficiency and consider alternative sets of features with a lower extraction cost, as well.

4.1 Content-based Models

Textual patterns are typically captured through language-specific representation models that detect frequent sequences of words (i.e., word n-grams) [5]. The settings we are considering in this work, however, pose significant obstacles to the applicability of term-based techniques, urging us to consider character-based models instead. Several reasons advocate this choice: first, character n-grams have been verified to outperform word n-grams in various applications, ranging from spam filtering [15] to authorship attribution [7] and utterance recognition [33]. Second, there is no standard tokenization approach for multilingual documents; words are typically identified through the whitespace that delimit them, but there are languages, such as Chinese, where different words can be concatenated in a single token.

Most importantly, though, term-based models depend on dictionary-based and language-specific techniques, such as *stemming* and *lemmatization*, to tackle *synonymy*; that is, words with the same meaning, but different syntactic form (e.g., quickly and rapidly), which are considered as distinct features, unless sophisticated methods for matching them are employed. Such techniques are inapplicable to the user-generated, multilingual microblog content, whose inherent noise (i.e., spelling mistakes) and neologisms further aggravate synonymy. As a result, both the effectiveness and the efficiency of term-based models is significantly degraded in our settings; the former is restricted due to missed patterns, stemming from semantically equivalent features that are treated as different, whereas the latter suffers from the *curse of dimensionality*; the diversity of the vocabulary leads

[6] `http://www.facebook.com/`
[7] `http://plus.google.com`

to a feature space with excessively high complexity and high computational cost.

For these reasons, we focus in the following on character-based representation models, namely the character n-grams and the character n-gram graphs. We also consider the term vector model - free from any optimizations - in order to illustrate the shortcomings of language-dependent methods in our settings. For each model, we explain how it represents individual tweets as well as a collection of tweets sharing the same polarity.

4.1.1 Term Vector Model

Given a collection of tweets T, this model aggregates the set of distinct words (i.e., tokens) $\mathcal{W}$ that are contained in it. Each tweet $t_i \in T$ is then represented as a vector $\vec{v_{t_i}} = (v_1, v_2, \ldots, v_{|\mathcal{W}|})$ of size $|\mathcal{W}|$, whose j-th dimension v_j corresponds to the TF-IDF weight of the j-th token $w_j \in \mathcal{W}$; that is, its value is defined as the product of its *Term Frequency* TF_i (i.e., the number of times w_j occurs in t_i) and its *Inverse Document Frequency* IDF_i (i.e., the cross-document frequency of w_j) [19]. The latter is defined as $DF_i = \log |T|/|\{t : w_i \in t \wedge t \in T\}|$, where the numerator stands for the size of the input tweets collection, and the denominator expresses the number of tweets that contain the word w_i. Similar to individual tweets, each polarity class T_p is modeled as a term vector $\vec{v_{T_p}}$ that comprises all tokens of the tweets t_i corresponding to it (i.e., $t_i \in T_p$).

4.1.2 Character N-grams Model

The set of *character n-grams* of a tweet comprises all substrings of length n of its text. The most common sizes for n are 2 (*bigrams*), 3 (*trigrams*) and 4 (*four-grams*). For example, the trigrams representation of the phrase "home_phone" is the following: { hom, ome, me_, _ph, pho, hon, one }. According to this model, each tweet t_i is represented by a vector v_{t_i} whose i-th dimension corresponds to the Term Frequency of the i-th n-gram [19]. Similarly, a polarity class T_p is modeled as a vector v_{T_p} that comprises all the n-grams contained in its tweets.

The main advantages of this model over the previous one are its language neutrality and its tolerance to noise and spelling mistakes: by considering substrings instead of entire words, their impact on the identified patterns is significantly reduced.

4.1.3 Character N-gram Graphs Model

The main drawback of the previous model is that it converts a tweet into a bag of n-grams, thus disregarding the valuable information that is encapsulated in the sequence of the n-grams of the original text. To overcome this problem, the *character n-gram graphs* method associates all neighboring character n-grams with edges that denote their (average) co-occurrence rate inside an individual tweet or a collection of tweets [9]. That is, it forms a graph whose nodes correspond to distinct n-grams, while its edges are weighted proportionally to the average distance - in terms of n-grams - between the adjacent nodes. To illustrate this structure, Figure 1 depicts the n-gram graph derived from the phrase "home_phone". Apparently, it conveys more information than the trigram representation of the same phrase in Section 4.1.2.

Formally, a character n-gram graph is defined as [9]:

Definition 1 (N-Gram graph). *An n-gram graph is a graph $G = \{V^G, E^G, W\}$, where V^G is the set of vertices (labeled by the corresponding character n-gram), E^G is the set of undirected edges (labeled by the concatenation of the labels of their adjacent vertices in alphabetical order), and W is a function assigning a weight to every edge.*

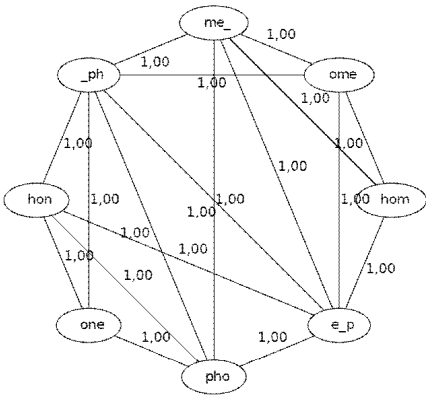

Figure 1: An example of a tri-gram graph that represents the phrase "home_phone".

According to this model, each tweet t_i is represented by a character n-gram graph G_{t_i} - called **tweet graph** - that is constructed by running a window of size n over it. During this process, the tweet is analyzed into overlapping character n-grams, recording information about the neighboring ones (i.e., those placed within the same window). Thus, an edge $e^{G_{t_i}} \in E^{G_{t_i}}$ connecting a pair of n-grams indicates proximity of these character sequences in the original text within the predefined window of size n [9]. The actual weight of the edges is estimated by measuring the percentage of co-occurrences of the corresponding vertices n-grams within the specified window.

A polarity class T_p is modeled by a single graph G_{T_p} that uniformly represents the tweets comprising it. This **class graph** is formed with the help of the *update functionality* [10][8]: given a set of tweets of the same polarity T_p, it builds an initially empty graph G_{T_p}. The i-th tweet $t_i \in T_p$ is transformed into the tweet graph G_{t_i} that is then merged with G_{T_p} to form a new graph G_u consisting of the union of the nodes and edges of the individual graphs; their weights are set equal to the average value of the weights of the individual graphs. More formally, G_u has the following properties: $G_u = (E^u, V^u, W^{u_i})$, where $E^u = E^{G_{T_p}} \cup E^{G_{t_i}}$, $V^u = V^{G_{T_p}} \cup V^{G_{t_i}}$ and $W^{u_i}(e) = W^{G_{T_p}}(e) + (W^{G_{t_i}}(e) - W^{G_{T_p}}(e)) \times 1/i$. Note that the division by i ensures that the aggregated weight converges to the mean value of the corresponding edge among all individual tweet graphs, thus turning the update functionality independent of the order by which tweets are merged [10]. After merging all individual tweet graphs into the class graph G_{T_p}, its edges $E^{G_{T_p}}$ encapsulate the most characteristic patterns contained in the class' content, such as recurring and neighboring character sequences, special characters, and digits.

To estimate the similarity between a tweet graph G_{t_i} and a class graph G_{T_p}, we employ one of the established n-gram graph similarity metrics [9]:

[8]An alternative approach would simply extract the class graph from the single tweet formed by the concatenation of all individual tweets. This practice, though, inserts noise in the form of edges between the last and the first n-gram of two consecutive, but actually independent tweets. In this way, it also depends on the order of concatenation.

(i) *Containment Similarity* (**CS**), which expresses the proportion of edges of graph G_{t_i} that are shared with graph G_{T_p}. Assuming that G is an n-gram graph, e is an n-gram graph edge and that for the function $\mu(e, G)$ it stands that $\mu(e, G) = 1$, if and only if $e \in G$, and 0 otherwise, then:

$$\text{CS}(G_{t_i}, G_{T_p}) = \sum_{e \in G_{t_i}} \mu(e, G_{T_p}) / \min(|G_{t_i}|, |G_{T_p}|),$$

where $|G|$ denotes the *size of the n-gram graph* G (i.e., the number of edges it contains).

(ii) *Value Similarity* (**VS**), which indicates how many of the edges contained in graph G_{t_i} are shared with graph G_{T_p}, considering also their weights. In more detail, every common edge e having weights $w^{t_i}(e)$ and $w^{T_p}(e)$ in G_{t_i} and G_{T_p}, respectively, contributes $\text{VR}(e)/\max(|G_{t_i}|, |G_{T_p}|)$ to the VS, where the *ValueRatio (VR)* is a symmetric, scaling factor that is defined as: $\text{VR}(e) = \frac{\min(w^{t_i}(e), w^{T_p}(e))}{\max(w^{t_i}(e), w^{T_p}(e))}$. It takes values in the interval $[0, 1]$, with non-matching edges having no contribution to it (i.e., for an edge $e \notin G_{t_i}$ we have $\text{VR}(e) = 0$). The full equation for *VS* now is:

$$\text{VS}(G_{t_i}, G_{T_p}) = \frac{\sum_{e \in G_{t_i}} \frac{\min(w^{t_i}(e), w^{T_p}(e))}{\max(w^{t_i}(e), w^{T_p}(e))}}{\max(|G_{t_i}|, |G_{T_p}|)}.$$

This measure converges to 1 for graphs that share both the edges and weights, with a value of VS = 1 indicating perfect match between the compared graphs.

(iii) *Normalized Value Similarity* (**NVS**), which decouples value similarity from the effect of the largest graph's size. Its value is derived from the combination of VS with SS (i.e., the *size similarity* of two graphs) as follows:

$$\text{NVS}(G_{t_i}, G_{T_p}) = \text{VS}(G_{t_i}, G_{T_p}) / \text{SS}(G_{t_i}, G_{T_p}),$$

where $\text{SS}(G_{t_i}, G_{T_p}) = \min(|G_{t_i}|, |G_{T_p}|)/\max(|G_{t_i}|, |G_{T_p}|)$.

On the whole, the n-gram graphs representation model captures the common textual patterns between a tweet t_i and a polarity class T_p through their CS, VS and the NVS similarities. These measures are substantially different from the cosine similarity of the n-grams model: the latter operates on the level of individual n-grams, whereas CS considers pairs of neighboring n-grams. VS goes one step further considering pairs of neighboring n-grams that have the same co-occurrence rate (i.e., edge weight), while NVS further enhances this approach by removing the effect of the relative size of the compared graphs.

The exact process for classifying a tweet t_i with the help of the n-gram graphs in the case of Problem 2 is presented in Figure 2: the tweet graph G_{t_i} is compared with each class graph (i.e., $G_{T_{neg}}$, $G_{T_{neu}}$, $G_{T_{pos}}$) to estimate its closeness to the corresponding polarity class. This is encapsulated in the values of three similarity metrics per class (i.e., CS, NVS and VS), which collectively form the feature vector that is given as input to a trained classifier. Based on the 9 - in total - features, the classifier decides for the most likely class label of tweet t_i. The same process is followed in case of Problem 1 with the only difference that there is no comparison with the neutral class graph $G_{T_{neu}}$ (i.e., the feature vector comprises just 6 features). This makes clear that the feature space of the n-gram graphs model depends on the number of considered classes and does not suffer from the dimensionality curse of the aforementioned representation models; the number of features the latter entail depends on the diversity of the vocabulary of the input document collection, typically amounting to several thousands of them (see Section 5 for more details).

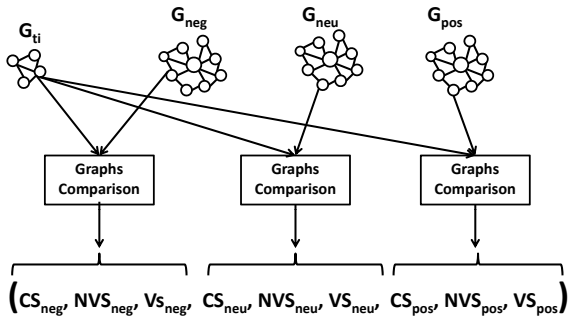

Figure 2: Deriving the feature vector from the n-gram graphs model for Problem 2.

Discretized N-Gram Graph Model. To enhance the classification efficiency of the n-gram graphs model, we propose an intuitive method for discretizing its similarity values. It employs pair-wise comparisons between the values of the same metric for different polarity classes, producing a nominal label according to the following discretization function:

$$dsim(sim_{pol_1}, sim_{pol_2}) = \begin{cases} pol_2, & \text{if } sim_{pol_1} < sim_{pol_2} \\ equal, & \text{if } sim_{pol_1} = sim_{pol_2} \\ pol_1, & \text{if } sim_{pol_1} < sim_{pol_2}, \end{cases}$$

where *sim* is the similarity metric (i.e., $sim \in \{CS, VS,$ or $NVS\}$) and pol_1 and pol_2 are the involved polarity classes (i.e., $pol_1, pol_2 \in \{neg, neu, pos\}$).

Thus, a tweet is classified in the Binary Polarity Problem according to 3 nominal features: $dsim(CS_{neg}, CS_{pos})$, $dsim(NVS_{neg}, NVS_{pos})$, and $dsim(VS_{neg}, VS_{pos})$. In the case of Problem 2, the following 6 additional features are derived from the comparisons of the neutral class similarities with the corresponding ones of the negative and the positive class: $dsim(CS_{neg}, CS_{neu})$, $dsim(NVS_{neg}, NVS_{neu})$, $dsim(VS_{neg}, VS_{neu})$, $dsim(CS_{neu}, CS_{pos})$, $dsim(NVS_{neu}, NVS_{pos})$, and $dsim(VS_{neu}, VS_{pos})$.

4.1.4 Punctuation Model

An alternative, language-agnostic method for detecting textual patterns has been proposed in [5]. It exclusively takes into account the punctuation and special characters that are contained in a tweet, thus being robust to spelling mistakes and neologisms. Its main advantage, though, is its minimal cost for extracting its features: they can be derived from a simple inspection of the characters of individual messages. In the following, we present its features, illustrating the rationale behind them through their average value for each polarity class, as it was estimated over the data set of 3 million tweets (1 million of randomly selected tweets per class) that is presented in Section 5.

(i) *Number of Special Characters.* It denotes the number of characters in a tweet that are neither alphanumerics nor white space. The higher their number is, the more likely is the corresponding message to be subjective. For example, it is common to add punctuation characters to stress a feeling and to replace abusive words with a series of incomprehensible symbols. Indeed, neutral tweets contain - on average - just 6.05 characters of this kind, whereas the positive and negative ones contain 6.44 and 7.76 characters, respectively.

(ii) *Number of "!".* Exclamation marks constitute a typical annotation for positive sentiments; the higher their number is, the more intense the positive feeling of a message is. Thus, positive messages contain 0.65 such characters on average, whereas the negative and the neutral ones contain 0.40 and 0.45 exclamation marks, respectively.

(iii) *Number of Quotes.* Quoted sentences are more likely to be found in objective tweets, whose authors cite other people's statements. Indeed, neutral messages contain 0.15 quotes on average, whereas subjective ones contain almost half as much: 0.08 for negative and 0.09 for positive tweets.

(iv) *Number of "?".* The higher the number of question marks in a message is, the more likely it is to be subjective, usually expressing a negative feeling. On average, tweets of this polarity contain 0.19 question marks in comparison to 0.16 and 0.14 for positive and neutral ones, respectively.

(v) *Number of Capitalized Tokens.* With the exception of abbreviations, capitalized tokens offer a strong indication for subjectivity; the higher their number is, the more intense the expressed feeling is. On average, negative and positive tweets contain 2.31 and 2.17 capitalized tokens, respectively, whereas objective tweets involve just 1.58 tokens of this kind.

(vi) *Tweet Length in Characters.* Negative tweets were found to consist - on average - of 95.05 characters, thus being larger than those of the other two polarity classes. They are followed by the the neutral ones that comprise 90.64 characters and the positive ones with just 88.92 characters.

4.2 Context-based Models

In addition to the textual patterns, another reliable source of evidence for detecting a tweet's sentiment is its **social context**. As such, we define any indication that associates it - directly or indirectly - with other messages (i.e., hashtags and URLs) or with members of the underlying social network (i.e., the author of the message, her friends as well as the users mentioned in it). In a similar vein to the spread of happiness that was suggested in [4], we argue that the overall polarity of the associated entities is critical for the polarity of individual messages; for example, the more positive tweets a user's friends have posted in the past, the more likely is her next tweet to be positive, as well. Note that this idea lies at the core of [27], as well, but it is employed in the context of *user-level sentiment analysis* (i.e., identifying the sentiment of a specific user with respect to a particular topic).

To quantify the effect of social context, we introduce a metric that estimates the aggregate sentiment of a set of tweets: the Polarity Ratio. We explain how it can be applied to the social context of individual tweets and present an intuitive way of discretizing its values. To verify its utility, we also examine an alternative approach of minimal extraction cost that relies on the raw form of the same contextual features (i.e., without taking the Polarity Ratio into account).

4.2.1 Social Polarity Model

The aggregate sentiment of a set of tweets is determined by the dominant polarity class: if the positive messages significantly outnumber the negative ones, the overall sentiment is considered positive and vice versa. This notion can be quantified through the following measure:

Definition 2 (Polarity Ratio). *Given a collection of tweets T, their **polarity ratio** $r_p(T)$ is defined as follows:*

$$r_p(T) = \begin{cases} \frac{|PT|+1}{|NT|+1} - 1, & \text{if } |NT| < |PT| \\ -\frac{|NT|+1}{|PT|+1} + 1, & \text{if } |PT| \leq |NT| \end{cases}$$

where $NT \subseteq T$ and $PT \subseteq T$ stand for the sets of negative tweets and positive tweets, respectively, with $|NT|$ and $|PT|$ representing their cardinality.

Polarity Ratio (**PR**) is defined in the interval $(-\infty, +\infty)$, with positive values suggesting the prevalence of positive tweets, and vice versa. More specifically, a positive value n suggests that the positive tweets are $n+1$ times more than the negative ones. Values very close to 0 corresponds to neutral aggregate polarity, denoting the absence of polarized tweets or the relatively equal portion of positive and negative tweets (i.e., $NT \approx PT$).

PR can be applied to all components of a tweet's social context, provided that they are represented by the set of messages pertaining to them. For example, the polarity ratio of the author's friends is calculated from the entire set of polarized messages they have already posted. Note that a critical point in this procedure is the temporal aspect of the tweets: we can only consider all the messages posted before the message in question. This is because we can only employ evidence from a tweet's past in order to predict its polarity.

In this work, we consider the following features:

(i) *Author Polarity Ratio.* It denotes the aggregate polarity of all messages posted by the same author prior to the given tweet t. Its value expresses her overall mood in the past, which is decisive for the sentiment of the subsequently published tweets. The more positive (negative) tweets she has already published, the more likely it is for t to be positive (negative), as well.

(ii) *Author's Followees Polarity Ratio.* Users pay particular attention to the messages posted by the users they subscribe to. They are expected, therefore, to be influenced by their opinions and sentiments. The higher (lower) the polarity ratio of their posts is, the more probable it is for t to be positive (negative) as well. To quantify this notion, this feature captures the aggregate sentiment of all messages posted by the author's followees prior to tweet t.

(iii) *Author's Reciprocal Friends Polarity Ratio.* It expresses the aggregate sentiment of the tweets posted by the author's reciprocal friends before she posted tweet t. These friends are expected to share a higher degree of homophily with the author [32] and, thus, the higher (lower) the polarity ratio of their posts is, the more probable it is for t to be positive (negative), as well.

(iv) *Topic(s) Polarity Ratio.* This feature is only valid for tweets containing at least one hashtag. It denotes the overall sentiment of all tweets that - regardless of their author - pertain to the same topic and have been posted prior to the given tweet t. In case a tweet contains more than one hashtag, this feature considers the entire set of tweets that is derived from the union of the individual sets of topic tweets. The higher the portion of positive (negative) tweets in the resulting set, the more likely it is for t to be positive (negative), as well.

(v) *Mention(s) Polarity Ratio.* This feature applies only to tweets containing at least one mention to a Twitter user. It represents the overall sentiment of all tweets that - regardless of their author - mention the same user and have been posted prior to the given tweet t. In case of multiple mentions, this feature considers the union of the individual sets of mention tweets. The more positive (negative) tweets mention a particular user, the more likely it is for t to be positive (negative), as well.

(vi) *URL(s) Polarity Ratio.* This feature is only applicable to tweets that contain at least one URL. It expresses the aggregate polarity of all tweets with the same URL that have been posted prior to the given tweet t, regardless of their author. In case a single tweet contains multiple URLs, this

feature considers the union of the individual URL tweets. The more positive (negative) tweets are associated with the referenced URL(s), the more likely it is for t to be positive (negative), as well.

Note that the first half of these features are based on social graph information, while the second half is exclusively derived from the related resources.

Discretized Social Polarity Model. Polarity Ratio produces numeric values, but their actual magnitude might be less significant than their polarity sign (i.e., positive or negative). If this is true, the processing of the corresponding nominal attributes will be significantly more efficient [16]. To validate these premises, we developed a novel method for discretizing the polarity ratio that depends on the polarity classification problem at hand. For its general version (i.e., Problem 2), the discretized polarity ratio $dr_p^{\mathcal{G}}(T)$ over a collection of tweets T takes as value one of the three polarity classes, based on the numeric value of $r_p(T)$, as follows:

$$dr_p^{\mathcal{G}}(T) = \begin{cases} negative, & \text{if } r_p(T) \leq -1 \\ neutral, & \text{if } -1 < r_p(T) < 1 \\ positive, & \text{if } 1 \leq r_p(T). \end{cases}$$

For its binary version (i.e., Problem 1), the discretized polarity ratio $dr_p^{\mathcal{B}}(T)$ is defined as follows:

$$dr_p^{\mathcal{B}}(T) = \begin{cases} negative, & \text{if } r_p(T) < 0 \\ equal, & \text{if } r_p(T) = 0 \\ positive, & \text{if } r_p(T) > 0. \end{cases}$$

4.2.2 Social Context Model

To reduce the feature extraction cost of the above model, we also consider an alternative set of context-based features that can be directly derived from a user's account and the characteristics of her messages. To illustrate their functionality, we present their average value for each polarity class, as it was derived from the data set of 3 million tweets (1 million tweets per class) that is presented in Section 5.

(i) *Number of Author's Tweets.* It represents the number of tweets the author of the given tweet t had published, prior to posting t. The authors of neutral tweets are more prolific, posting 387 messages on average, whereas the authors of negative and positive ones post 356 and 298 tweets, respectively.

(ii) *Number of Author's Followees.* It denotes the number of users the author of the input tweet t had subscribed to, prior to publishing t. Authors of neutral tweets were found to have the most subscriptions (351 followees on average), followed by the authors of the positive (281 followees) and the negative ones (271 followees).

(iii) *Number of Author's Reciprocal Friends.* It stands for the number of reciprocal friends the author of the given tweet had, before publishing it. Authors of neutral tweets were found to have substantially more reciprocal friends (244 on average), followed by the authors of positive (195) and negative ones (181).

(iv) *Number of Topics.* It denotes the number of hashtags contained in the given tweet. Objective messages are typically related to a larger number of topics (0.14 hashtags on average), while subjective tweets contain almost half as much (0.08 hashtags), independently of their polarity.

(v) *Number of Mentions.* It expresses the number of users mentioned in the input tweet. Positive tweets were found to contain the highest amount of mentions (0.75 on average), whereas negative and neutral ones merely refer to 0.51 and 0.54 users, respectively.

(vi) *Number of URLs.* It denotes the number of URLs that the given tweet contains. The higher their number is, the more likely the author is to provide her subscribers with objective information; indeed, neutral tweets contain the highest number of links (0.43 on average), whereas the positive ones contain half as much (0.21). Negative ones lie in the middle of these two extremes, with 0.36 URLs on average.

Basically, these features rely on the same evidence with the Polarity Ratio model, but do not take into account the aggregate polarity of the underlying instances. Thus, they are directly comparable with it, illustrating the contribution of Polarity Ratio to the accuracy of context-based models.

5. EVALUATION

In this section, we analytically present our thorough experimental study that aims at identifying the optimal *classification settings* for Sentiment Analysis over microblogs; that is, the combination of a classification algorithm and a set of features that offers the best balance between effectiveness and efficiency.

Data Set. To examine the performance of our models in practical settings, we conducted a thorough experimental study on a large-scale multilingual collection of real Twitter messages. It is the same data set that was employed in [35], comprising 476 million tweets posted in a period of 7 months - from June 2009 until December 2009. Among them, we identified 6.12 million negative and 14.12 million positive tweets following the common practice in the literature, which employs emoticons as a golden standard [5, 1, 22][9]. We randomly selected 1 million tweets from both polarity classes to form the data set for Problem 1, called D_{binary}. We additionally selected a random sample of neutral tweets to create the data set for Problem 2, called $D_{general}$. Both of them are among the largest data sets employed so far in the context of Sentiment Analysis over microblogs. Note also that our sampling did not restrict the selected tweets to specific language, so as to ensure the multilinguality of our data sets.

To derive the social context of individual tweets, we employed the snapshot of the entire Twitter social graph that was used in [17], which dates from August 2009. This time period coincides with that of the recorded messages, but does not depict the actual evolution of the underlying social network during that period. Its static information allows only for a mere approximation of the actual performance of the context-based models. To estimate the actual value of the polarity ratios they involve, we relied again on the positive and negative emoticons.

Metrics. To measure the effectiveness of the classification models, we considered the established metric of **classification accuracy** α. It expresses the portion of the correctly classified tweets and is formally defined as follows: $\alpha = \frac{TP}{TP+FP}$, where TP stands for true positives (i.e., the number of tweets that were assigned to the correct polarity class) and FP denotes false positives (i.e., the number of incorrectly classified tweets).

Evaluation Method. To evaluate the performance of

[9]Assuming that a positive (negative) emoticon always corresponds to a positive (negative) sentiment is a simplification hypothesis. Nevertheless, it is the only method employed in the literature for large-scale experimental studies.

	Prob. 1	Prob. 2
Term Vector	67.38%	50.68%
2-grams	61.99%	50.11%
3-grams	68.72%	53.15%
4-grams	70.62%	53.41%
2-gram Graphs	64.38%	45.86%
3-gram Graphs	79.95%	65.28%
4-gram Graphs	**91.51%**	**83.80%**
Discr. 2-gram Graphs	65.58%	48.01%
Discr. 3-gram Graphs	89.71%	78.52%
Discr. 4-gram Graphs	**97.12%**	**93.43%**

Table 1: Accuracy of Naive Bayes Multinomial.

	Prob. 1	Prob. 2
Term Vector	70.66%	52.65%
2-grams	69.80%	57.36%
3-grams	72.89%	57.86%
4-grams	71.76%	54.63%
2-gram Graphs	68.70%	57.11%
3-gram Graphs	74.02%	63.12%
4-gram Graphs	**86.10%**	**79.18%**
Discr. 2-gram Graphs	64.35%	52.67%
Discr. 3-gram Graphs	71.69%	60.89%
Discr. 4-gram Graphs	**84.57%**	**78.82%**

Table 2: Accuracy of Support Vector Machines.

	Prob. 1	Prob. 2
Term Vector	1,245	1,221
2-grams	1,796	1,848
3-grams	6,255	6,358
4-grams	10,888	11,045
2-gram Graphs	6	9
3-gram Graphs	6	9
4-gram Graphs	6	9
Discr. 2-gram Graphs	3	9
Discr. 3-gram Graphs	3	9
Discr. 4-gram Graphs	3	9

Table 3: Number of features per representation model.

our models, we employ the 10-fold cross-validation approach. For the evaluation of the n-gram graphs model, we followed a special procedure: first, we randomly selected half of the training set of each polarity class to build the corresponding class graph. Then, the tweet graphs of all training instances are compared with all polarity graphs and the classification algorithm is trained over the resulting similarities values. Finally, the tweet graphs of the testing instances are compared with all class graphs and the trained algorithm decides for their label according to the derived similarity features. It should be stressed at this point that the emoticons were removed from all training and testing tweets, when building any of their representation models.

Classification Algorithms. To thoroughly evaluate the performance of our models, we consider several state-of-the-art classification algorithms of varying time and space complexity. For the comparative analysis of the document representation models, we employed the Naive Bayes Multinomial (**NBM**) and the Support Vector Machines (**SVM**), two established algorithms for text-categorization, with the former being substantially more efficient than the latter [34]. For the rest of the models, we employed three of the most popular and established classification algorithms: Naive Bayes (**NB**), C4.5 and the SVM. They comprise a quite representative set of classification methods with respect not only to their internal functionality (i.e., probabilistic learning, decision trees and statistical learning, respectively), but also to their efficiency (they appear in ascending order of time complexity). For a detailed description of these algorithms, see [34].

Setup. All models and experiments were fully implemented in Java, version 1.6. For the functionality of the n-gram graphs, we employed the open source library of JInsect[10]. For the implementation of the classification algorithms, we used the Weka open source library[11], version 3.6 [34]. The only exception was the use of the LIBLINEAR optimization technique [8], which was employed for scaling the SVM to the high dimensionality of the term vector and the n-grams models. Given that LIBLINEAR also employs linear kernels for training the SVM, it is directly comparable with the Weka's default SVM configuration, which was applied to the other models. In every case, we employed the default configuration of the algorithms, without fine-tuning any of the parameters. All experiments were performed on a desktop machine with 8 cores of Intel i7, 16GB of RAM memory, running Linux (kernel version 2.6.38).

Comparison of Content-based Models. To identify the most appropriate representation model for the sparse, noisy, multilingual, user-generated content of microblogs, we applied all models of Sections 4.1.1 to 4.1.3 to the data sets D_{binary} and $D_{general}$. For the term vector and character n-grams model, we did not employ any preprocessing technique, due to the multilingual content we are considering. To limit the feature space, we merely employed a threshold of minimum frequency, setting it equal to 0.01% of the size of the input tweets collection. Thus, words or n-grams that appear in less than 200 (300) of the tweets of D_{binary} ($D_{general}$) were not taken into account for Problem 1 (Problem 2). The outcomes of our experiments are presented in Tables 1 to 3.

Table 1 reveals the following interesting pattern in the performance of NBM over both polarity classification problems: the accuracy of the n-grams model increases with the increase in n, exceeding that of the term vector model for $n > 2$ (i.e., for trigrams and four-grams). As expected, the n-gram graphs follow the same pattern: the higher the value of n, the higher the classification accuracy. Most importantly, though, they outperform both the term vector and the corresponding n-grams model in all cases, but for $n > 2$. It is remarkable, though, that the discretization of the graph similarities conveys a significant increase to the performance of the n-gram graphs, which exceed 10% in most cases. On the whole, the highest accuracy is achieved by the four-gram graphs with discretized similarity values.

Similar patterns are depicted in the performance of SVM: the term vector model exhibits the lowest effectiveness, followed by the n-grams model, whose accuracy increases with the increase of n. The n-gram graphs outperform all other models, with their accuracy increasing proportionally to n. Note, though, that their discretized values induce no improvement in accuracy, probably because SVM are crafted for numeric attributes. It is also worth noting that, in several cases, the SVM exhibits a lower performance than NBM; the reason is that we skipped the time-consuming process of configuring SVM parameters (e.g., kernel functions) to their optimal values.

The low efficiency of the traditional representation models is reflected in Table 3: the n-grams involve the most complex feature space among all representation models, with their dimensionality increasing significantly with the increase of n. They are followed by the term vector model, which employs around 30% less features than bigrams; this is because n-grams are more frequent than entire tokens, thus resulting in a higher number of features that exceed the frequency threshold. In complete contrast, the n-gram graphs involve three orders of magnitude less features, as their dimensionality depends on the number of classes rather than the diversity of the vocabulary of the given tweets.

On the whole, the four-gram graphs achieve the highest accuracy across all representation models and classification algorithms - especially after discretizing their values - even when they are combined with a highly efficient, but simple

[10] http://sourceforge.net/projects/jinsect

[11] http://www.cs.waikato.ac.nz/ml/weka

	Problem 1						Problem 2					
	4-gram Graphs	Discr. Graphs	Punct.	Polarity	Discr. Polarity	Social Context	4-gram Graphs	Discr. Graphs	Punct.	Polarity	Discr. Polarity	Social Context
NB	91.51%	96.36%	56.64%	53.40%	74.61%	51.05%	75.82%	93.43%	44.69%	37.40%	60.02%	34.33%
C4.5	**98.76%**	97.17%	60.98%	80.08%	72.89%	60.44%	**96.85%**	94.98%	46.00%	66.55%	61.47%	46.38%
SVM	86.10%	84.57%	50.12%	73.19%	72.89%	56.93%	79.18%	78.82%	39.02%	52.86%	57.27%	36.68%

Table 4: Accuracy of all combinations between models and classification algorithms over both polarity problems.

algorithm like NBM. This means that they are more suitable for tackling the inherent characteristics of microblog content (cf. Section 1) and, thus, we exclusively employ this content-based model in the following.

Content-based vs. Context-based Models The performance of all models of Sections 4.1.3 to 4.2.2 is illustrated in Table 4. Note that the lowest meaningful accuracy (i.e., the performance of the random classifier) is 50% for Problem 1 and 33.33% for Problem 2.

We can notice the following patterns: first, the four-gram graphs model outperforms Social Polarity by (around) 20% in the case of Problem 1 and 30% for Problem 2. Given, though, that the latter does not consider textual patterns at all, its performance is remarkable, as its accuracy is comparable or even better than that of the traditional document representation models (i.e., term vector and n-grams model).

Second, the features with low extraction cost (i.e., Punctuation and Social Context Model) have a performance very close to that of the random classifier, unless they are combined with C4.5. Even in that case, though, their accuracy is significantly lower than that of the n-gram graphs and the Social Polarity model, respectively. For context-based models, this apparently means that plain context features capture rather poor information, thus turning PR indispensable for high effectiveness.

Third, the discretization methods have a rather small impact on the effectiveness of all models across both polarity problems; they degrade accuracy just by 6% for Social Polarity and less than 3% for 4-gram Graphs. For NB, though, they consistently boost accuracy by more than 10%.

Fourth, the additional, third polarity class in Problem 2 has a significant impact on the less effective approaches, reducing their accuracy by 10% in most cases. However, considering the highest performance in each problem, we can see that the effect of the additional class is minimal, lowering accuracy just by 2%.

Fifth, and most important, the C4.5 algorithm achieves the highest performance across most models and problems, taking values very close to absolute correctness. However, there is no clear winner between NB and SVM, probably because of the absence of configuration for the parameters of the latter. Nevertheless, the performance of NB is very close to that of C4.5 when applied on discretized features. In fact, the combination of NB with discretized features provides the best balance between effectiveness and efficiency: it is by far the most efficient combination, while its effectiveness is just 2% lower than the maximum one in both problems.

It is worth noting at this point that we also examined the combination of context-based models with content-based ones, but do not present the exact outcomes due to lack of space. It suffices to say that it was significantly lower than that of the four-gram graphs across all algorithms and problems. For instance, the accuracy of C4.5 was 93.09% and 89.75% for Problem 1 and Problem 2, respectively; for the other algorithms, though, the performance was around the average accuracy of the individual values. This means that their combination inserts noise in the classification procedure, thus indicating that the four-gram graphs model alone provides the optimal approach to SA over microblog content.

Model	Problem 1	Problem 2
4-gram Graphs	CS_{neg}, CS_{pos}	$CS_{neg}, CS_{neu}, CS_{pos}$
Discr. Graphs	$dsim(CS_{neg}, CS_{pos})$	$dsim(CS_{neg}, CS_{neu})$ $dsim(CS_{neg}, CS_{pos})$ $dsim(CS_{neu}, CS_{pos})$

Table 5: The selected features of the 4-gram graphs model for both polarity classification problems.

	Problem 1		Problem 2	
	4-gram Graphs	Discr. Graphs	4-gram Graphs	Discr. Graphs
NB	82.87%	97.12%	76.41%	94.92%
C4.5	97.42%	97.12%	95.40%	94.92%
SVM	84.53%	97.12%	80.43%	94.82%

Table 6: Accuracy of the *filtered* features of the four-gram graphs model.

Attribute Filtering over 4-gram Graphs. To further enhance the efficiency of the four-gram graphs model, we applied on its features the correlation-based feature subset selection method [12] combined with the best-first search algorithm. The selected features for both problems are depicted in Table 5. Basically, the features that rely on the containment similarity were chosen, indicating that CS captures the most reliable textual patterns. Their performance over D_{binary} and $D_{general}$ is presented in Table 6.

We can easily notice that there is a negligible decrease in accuracy by around 1% for the numeric features. In the case of the nominal features, however, the reduction is even lower, being - in fact - statistically insignificant. Most importantly, though, the discretized features exhibit exactly the same accuracy across all algorithms. This means that all useful information is encapsulated in the selected features, enabling the use of simple, highly efficient classification algorithms without any impact on effectiveness.

6. RELATED WORK

Several surveys have recently reviewed the most prominent works on Sentiment Analysis [23, 28, 30]. Among them, though, only [30] discusses the new trend of mining sentiments in the streaming, user-generated content of microblogs. Similar to our work, the majority of relevant papers examines SA on the level of individual documents. Depending on the specific sub-problem they are tackling, they can be grouped in the following categories:

Predictive Sentiment Analysis. The aim of these works is to discover strong correlations between the aggregate sentiment of a collection of tweets and the traditional measures for polling public opinion (e.g., political elections). For example, [21] employed a large corpus of Twitter messages and verified that its aggregate mood provides a good estimation of the *evolution* of consumer confidence and the approval of presidential work in the USA. In a similar vein, [31] analyzed a large collection of tweets and found out that the relative frequency of mentions to political parties was strongly correlated with the actual outcomes of Germany's

presidential elections in 2009. Equally strong is also the correlation of Twitter's aggregate mood with the evolution of stock markets and the value of the Dow Jones Industrial Average, in particular [2].

Fine-grained Sentiment Analysis. The goal of these works is to identify the correct feeling among a larger set of possible sentiments. For example, [3] considers the six distinct emotional states, collectively called POMS (i.e., Tension, Depression, Anger, Vigour, Fatigue and Confusion), whereas [29] considers the eight primary emotions (i.e., acceptance, fear, anger, joy, anticipation, sadness, disgust and surprise). At a finer sentiment granularity, [5] defines 51 different sentiments extracted from hashtags along with 16 ones extracted from smileys, introducing a classification scheme that applies k-NN on top of context-based features.

Target-dependent Sentiment Analysis. The works of this category apply SA techniques on the results of keyword queries, categorizing them into positive, negative and (rarely) neutral. This task has already been explored in the context of Web pages and news articles [20] as well as for customer reviews [25]. In the field of microblogs, it is primarily explored by on-line services that offer SA over Twitter, such as Twendz and TweetFeel. Furthermore, works in [11] and [26] attempted binary Sentiment Classification in Twitter using content as well as context features in the second case, reaching accuracies up to 84.7% in the best case. The main drawback of those solutions is that they are language-dependent, with the exception of [11], and they are also based on target-independent algorithms. [14] improves on them with a novel, three-step approach that is target-dependent and context-aware.

7. CONCLUSIONS

In this paper, we examined several content-based techniques for capturing textual patterns for SA over microblog content and verified that traditional models are inadequate for tackling the intricacies it involves. For higher effectiveness, we proposed contextual features as well as the n-gram graphs model and several techniques for enhancing their efficiency: discretization, attribute filtering and naive classification algorithms. Our experimental evaluation validated that high levels of accuracy and efficiency can be achieved simply by assigning each tweet to the polarity class that shares the maximum number of neighboring pairs of 4-grams with it.

In the future, we intend to further improve the performance of contextual features, as they are particularly useful in real-world integrated SA applications: they involve a minimal extraction cost (merely requiring some counters) and are capable of handling the intricacies of microblog content. Our plan is to enhance PR by taking time into account, so that it considers only the latest posts of a message's context. We also plan to examine how the update functionality of n-gram graphs adapts to new patterns, supporting the evolution of sentiments over time.

Acknowledgement

This work has been partly funded by the FP7 EU Project SocIoS (Contract No. 257774).

References

[1] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *COLING*, pages 36–44, 2010.
[2] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2011.
[3] J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, 2011.
[4] N. Christakis and J. Fowler. *Connected: The surprising power of our social networks and how they shape our lives*. Little, Brown and Company, 2009.
[5] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *COLING*, 2010.
[6] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *EMNLP*, pages 1277–1287, 2010.
[7] H. Escalante, T. Solorio, and M. Montes-y Gómez. Local histograms of character n-grams for authorship attribution. In *ACL*, pages 288–298, 2011.
[8] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.
[9] G. Giannakopoulos, V. Karkaletsis, G. A. Vouros, and P. Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *TSLP*, 5(3), 2008.
[10] G. Giannakopoulos and T. Palpanas. Content and type as orthogonal modeling features. *International Journal of Advances on Networks and Services*, 3(2), 2010.
[11] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6, 2009.
[12] M. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, University of Waikato, 1999.
[13] M. Hurst, M. Siegler, and N. Glance. On estimating the geographic distribution of social media. In *ICWSM*, 2007.
[14] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent Twitter sentiment classification. In *COLING*, 2011.
[15] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos. Words versus character n-grams for anti-spam filtering. *IJAIT*, 16(6):1047, 2007.
[16] L. Kurgan and K. Cios. Caim discretization algorithm. *IEEE TKDE*, pages 145–153, 2004.
[17] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW*, 2010.
[18] B. Liu. *Web data mining*. Springer, 2007.
[19] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.
[20] T. Nasukawa and J. Yi. Sentiment analysis: capturing favorability using natural language processing. In *K-CAP*, 2003.
[21] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *ICWSM*, 2010.
[22] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.
[23] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2008.
[24] G. Shao. Understanding the appeal of user-generated media: a uses and gratification perspective. *Internet Research*, 2009.
[25] G. Somprasertsri, P. Lalitrojwong, and P. Lalitrojwong. Mining feature-opinion in online customer reviews for opinion summarization. *Journal of Univ. Comp. Science*, 2010.
[26] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *EMNLP*, pages 53–63, 2011.
[27] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. User-level sentiment analysis incorporating social networks. In *KDD*, 2011.
[28] H. Tang, S. Tan, and X. Cheng. A survey on sentiment detection of reviews. *Expert Systems with Applications*, 2009.
[29] K. Tsagkalidou, V. Koutsonikola, A. Vakali, and K. Kafetsios. Emotional aware clustering on micro-blogging sources. In *ACII*, 2011.
[30] M. Tsytsarau and T. Palpanas. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery Journal*, 2011.
[31] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM*, 2010.
[32] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.
[33] T. Wilson and S. Raaijmakers. Comparing word, character, and phoneme n-grams for subjective utterance recognition. In *INTERSPEECH*, 2008.
[34] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
[35] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186, 2011.