Informative vs. Non-informative Short Message Detection in Social Networks

Abstract—**We propose a method for classifying tweet messages into two classes: informative and non-informative. We consider informative messages those that contain information that interests the public, trends, events and news. On the other hand, non-informative tweets are personal messages that do not interest the general public. Conversations between friends, feelings and description of mood fall into the latter category. The motivation of our work is cleaning a dataset of tweet messages by filtering out non-informative tweets. Non-informative messages is noise that causes obstacles in retrieving from a dataset of tweets. On the contrary, informative tweets are more likely to contain essential information that is worth it to processing. Hence, by discarding noisy tweets, one can focus merely on processing useful messages. Real applications that can benefit from our work are trend/topic detection applications, recommendation systems and applications that make predictions based on messages that user exchange through social networks.**

Challenges of processing tweet messages is that they are short messages, unstructured with unclear topic. We propose a weighted variation of the binary multinomial naive Bayes' model to identify informative messages. We train our classifier and we evaluate results using 5-fold and 10-fold cross validation. We compare the results with the original binary multinomial naive Bayes' model. We use two independent datasets of tweet messages crawled from the web. We evaluate and present our results using the following metrics: accuracy, recall, specificity, F-messure with its variations (F_2 score and $F_{0.5}$ score).

I. INTRODUCTION

Social media like Twitter and Facebook have been emerging. Facebook has more than one billion active users, and Twitter has about two hundred million active users [1]. Users communicate with each other through these social networks every day, they share their experiences, express their opinions, ideas and feelings. In addition to individual users, social media are used by companies that promote their products and services, journalists that spread news, organizations that inform the public about actions they take. Social media allow users to be connected to each other, e.g. followers, followees, friends, and allow them to publish short messages that are visible either to everyone on the network or only to friends. These short messages are usually called posts or tweets. Moreover, users comment on and reply to other users' posts, or they may express their interest by 'liking' or 'favoriting' other posts. In addition, content of posts varies. It can be text messages, emoticons, website URLs, photos, videos.

Some of the characteristics of posts and tweets are the following: They are short messages, their topic cannot always be clearly specified, sometimes their content includes multiple topics, each user can post unlimited number of messages that spread to everyone that is connected with them. However, not all of the messages are useful for extracting useful information. Challenges of processing tweets are the following:

1) The flow of exchanged tweet messages is high. It is estimated that 6,000 tweets are sent per second [2]. Collecting and processing even a portion of these messages in real time is not trivial. This results in having very large datasets (i.e. volume), that have to be processed instantly (i.e. speed).
2) Messages are short, unstructured with respect to grammar and syntax. They contain informal vocabulary, sometimes even slang words. They have the form of short announcements, titles or they look like conversations. Sometimes they are not even messages. They may be geographic coordinates, urls, links of arcticles and photos, emoticons, or even discription of expressions in quotes. Extracting useful information out of this kind of dataset using tranditional techniques is very hard.
3) Concerning content, their topic may not be clear, or they may refer to a wide range of topics. Every tweet may belong to any, or more that one topics.

Real applications that need to overcome these challenges are: recommendation systems, topic/trend detection applications, geo-social systems and systems that make predictions based on users' tweets.

Data cleaning is the first phase of processing. It includes stopword removal, stemming, building dictionary of terms and collecting their frequencies. Next phase is the information retrieval. Tranditional approaches do not seem to perform well when it comes to detection of informative tweets. In particular, *building dictionary indexes* is affected by slang words, spelling mistakes, words in other languages. *Weighting of terms*, e.g. tf-idf, suffers from the fact that each document, i.e. tweet, consists of few, non-repeated terms. Also, the frequency of terms is not related with their importance. For instance, we observe in our datasets that the distribution of stopwords in informative and non-informative tweets is similar. *Scoring models*, e.g. vector space models, suffer from high dimensionality, i.e. large number of tweets, or vocabulary terms. In addition, algorithms do not always perfom well, because vectors are sparse.

Traditional information retrieval approaches do not fit for discrimination between informative and non-informative tweets. In particular, *classification* like naive Bayes' models calculate the probability of each vocabulary term belonging to a class, without considering its meaning. In addition, it is not necessarily true that terms are always independent. In the case of the *vector space* models, like k-NN, SVMs, one has to predefine the number of clusters, or appropriate set of features

[1] http://www.statista.com

[2] http://www.internetlivestats.com/twitter-statistics/

respectively. The problem with *unsupervised techniques* like k-means and hierarchical clustering, is that they require advanced knowledge or estimation of the number of topics. In *matrix decompositions methods*, e.g. SVD, number of dimensions and sparsity of data cause problems. In *latent topic models* like LDA, it is not easy to define the topics, the distribution of words in each topic, and the mixture of topics in each dataset. Also, the inference techniques can be slow because of the characteristics of the datasets.

In this paper we focus on the first step, data cleaning of messages. Stopword removal and term stemming discard commonly used words or they combine words with the same stem, but they do not evaluate the message as a whole, and they do not consider its content. We are proposing a variation of the binary multinomial naive Bayes' model where the whole tweet message is evaluated and we discriminate 'informative' messages from 'non-informative' messages.

The motivation of our work is the removal of useless tweet messages. First of all, we define when a tweet is informative (useful), or non-informative (useless). 'Informative' tweets are those messages that interest the public, mention public events, produce or reproduce the news, refer to commonly known places or celebrities, quotes of famous people, general beliefs and stereotypes. On the other hand, 'non-informative' tweets express personal opinions and feelings without reference to any useful information. These are tweets that look like conversations between friends on topics that the public is not aware of, automatic follow requests, user statistics relevant with the daily number of followers and unfollowers. We believe that non-informative messages should be considered as noise and should be discarded because they increase the size of the dataset (number of tweets) and the dimensionality of problems (size of vocabulary) without providing useful information. Afterwards, information retrieval techniques can be applied to the informative tweets which is a smaller portion of the original dataset, with better results.

Challenges we meet during this discrimination are:

1) Undefined number of topics are discussed in each corpus of tweet datasets.
2) Users use different vocabulary for the same topic. Some users use informal language.
3) Tweets length is short. This does not allow traditional techniques like LDA to give good results.

We are proposing two solutions to address these challenges:

1) We focus on three textual features of the tweets: hashtags, mentioned users, words. Hashtags and mentioned users in a tweet are more significant than the rest of the words. For instance, tweets that have the hashtag '#news' are very likely to refer to news, or tweets with another hashtag are more likely to follow a trend. Following the same reasoning, when a tweet mentions the username of a journalist, or a politician, it is more likely that that tweet contains useful information. We model these cases by assigning different weights on the three textual features we mention.
2) The distribution of the vocabulary terms is significant and can boost the performance of a model. We incorporate this idea by estimating the prior distribution of the terms.

We incorporate our ideas to the binary multinomial naive Bayes' model by adding different weights to the textual features. In addition, we consider a prior distribution over the terms of the vocabulary. We model the prior distribution by sampling our training dataset and we incorporate it to the binary multinomial naive Bayes' model. Finally, we evaluate our results. We compare our proposed weighted variation of the binary multinomial naive Bayes model with the original binary multinomial naive Bayes model. The evaluation metrics that we use are: accuracy, recall, specificity, F-messure with its variations (F_2-score and $F_{0.5}$-score). A good result is expected to give high recall and high F_2-score, because we are interested in detecting most of the informative messages of our datasets.

The arrangement of the rest of the paper is the following. In Section II we discuss the related work. In Section III we describe our model. Finally, in Section IV we present experiments that evaluate our model.

II. Related Work

Our proposed model is a variation of the binary multinomial naive Bayes' model. An early description of the naive Bayes' classifier can be found in [10], and its feature independence assumption is discussed in [8], [14].

Naive Bayes' text classifiers have been used in the past for categorization of messages in the fields of email spam filtering, email categorization, news article categorization, social content categorization, product classification, sentiment detection. In particular, email categorization techniques are compared in [13]. Email spam filtering methods are described in [17], [21], and event models for naive Bayes' anti-spam email filtering are compared in [4], [22]. An evaluation of naive Bayesian anti-spam filtering techniques is presented in [3]. SMS spam filtering with naive Bayes', SVMs and other classifiers are discussed in [12]. Although detection of spam emails looks similar to the detection of non-informative tweets, these methods cannot be used in tweets, because of the challenges of tweet messages that we have mentioned in Section I.

Moreover, Bayesian analysis techniques for internet traffic classification is discussed in [18]. Also, content-based recommendation systems with naive Bayesian classifiers are presented in [20]. Combination of logistic regression and Naive Bayes' for email spam filtering is proposed in [6]. A comparison of event models for naive Bayes' text classification is described in [16].

Supervized learning methods like support vector machines (SVMs) have also been used for text classification. SVMs are described in [5], [7], [23]. A comparison of naive Bayes' model, linear SVM and other classification techniques in text categorization is presented in [11]. Machine learning techniques for automated text categorization including Naive Bayes, k-NN, SVMs, decision trees are discussed in [25]. Sentiment classification using machine learning techniques in proposed in [19]. SVMs on spam filering are used in [24].

A personalized tweet ranking method is proposed in [26]. This work ranks tweets given a user, so that related tweets for each user can be found. This work also ranks users given a tweet, so that the most interested user can be found for this tweet. However, in our work, we are not interested in tweet recommendation to users. The model presented in [9] combines content relevance of a tweet with the account authority and tweet-specific features such the existence of URL. The goal of this work is to choose the most relevant tweets given a query. However, in our work we do not aim at finding similar tweets to a query.

III. Approach

We begin with the modifications that we make to the traditional binary multinomial naive Bayes' model. Firstly, we enchance the vocabulary of the binary multinomial naive Bayes' model, so that it consists of hashtags, mentioned users and words that appear in tweets. Secondly, the binary multinomial naive Bayes' model treats all the vocabulary terms the same way. However, hashtags and mentioned users are more likely to indicate whether a tweet contains important information. Thus, in our proposed variation we assign different weights on them. Thirdly, the binary multinomial naive Bayes' model uses Laplace smoothing. This is equivalent to Uniform prior distribution of the vocabulary terms. Thus, in our proposed variation we use the training data in order to estimate a prior distribution that fits our dataset, and we replace Laplace smoothing with this prior.

For testing our model we use two independent, completely different datasets crawled from the web. These datasets were crawled at different time periods, so there is no overlap in their timestamps. Also, they refer to different topics and trends. There are no any common messages, users, hashtags, and events or trends between the two datasets. We parse tweets by using the *Twitter4j* [2] Java library for Twitter API. Firstly, we label them manually as informative and non-informative. Then, we perform a preprocessing step in which we remove stopwords. During this step we tokenize each tweet message and we process each token by taking its stem using the *Snowball* [1] stemmer. Finally, we build vocabulary of tokens. The vocabulary includes all three textual features (hashtags, mentioned users, words).

A. The binary multinomial naive Bayes' model

This model is fully described in [15]. It generates one term from the vocabulary in each position of the document. It is a probabilistic learning method. The probability of a tweet t belonging to a class c (informative or non-informative) is computed as:

$$P(c \mid t) \propto P(c) \prod_{1 \leq k \leq n_t} P(term_k \mid c) \tag{1}$$

where n_t is the total number of tweet messages. Or, in detail:

$$P(c \mid t) \propto P(c) P(t \mid c) \tag{2}$$

where

$$\begin{aligned} P(t \mid c) &= P(term_1, ..., term_{n_t} \mid c) \\ &= \prod_{1 \leq k \leq n_t} P(term_k \mid c) \end{aligned} \tag{3}$$

$P(term_k \mid c)$ is the conditional probability of term $term_k$ occuring in a tweet of class c. $P(c)$ is the prior probability of a tweet occuring in class c.

In order to estimate the parameters $\hat{P}(c)$ and $\hat{P}(term_k \mid c)$, we use the maximum likelihood estimate (MLE), which is the relative frequency in the data. It corresponds to the most likely value of each parameter given the training data. Hence, for the priors:

$$\hat{P}(c) = \frac{N_c}{N} \tag{4}$$

where N_c is the number of tweet messages in class c and N the total number of tweet messages. For the conditional probability $\hat{P}(term_k \mid c)$, we estimate the relative frequency of term $term$ in tweets belonging to class c. In other words, it is the fraction of how many times word $term_k$ appears amongst all the words of tweets that belong to class c.

$$\hat{P}(term_k \mid c) = \frac{T_{ck} + 1}{\sum_{k' \in V} T_{ck'} + 1} \tag{5}$$

where T_{ck} is the number of occurences of term k in training tweet messages from class c. It is a count of occurences in all positions k in the tweets in the training set. Zero probabilities cannot be conditioned away, thus *Laplace smoothing* is applied for smoothing of the Naive Bayes estimates.

B. Textual Features

We categorize the tokens of each message into three groups: hashtags, users mentioned, and words. The vocabulary of the training data includes all of them. In the vocabulary, hashtags start with the '#' symbol, so that they will not be confused with words. The users mentioned are usernames of others users that are mentioned in a tweet and they include the '@' character in the begining.

We assign different weights to each textual feature. Hence, there are three types of weights: $w_h^{(f)}$ for hashtags, $w_u^{(f)}$ for mentioned users, $w_w^{(f)}$ for words. All the terms of each feature have the same weight. This means that all hashtags are assigned with the same weight: $w^{(f)}h$ and so on.

C. Variation of the binary multinomial naive Bayes' model

We incorporate the weights to the binary multinomial naive Bayes' model. We make two modifications to this model. The first one is that we apply the above mentioned weights on the frequency occurencies for each group of terms. The second one is that we change the Laplace smoothing parameter. The latter modification is related with the prior distribution of the vocabulary terms.

1) Frequencies: The parameter for each vocabulary term in the multinomial model is defined by Equation 5. We apply the weights on the frequency occurence for each term.

So, each term parameter can be estimated as follows:

$$\hat{P}(term_k \mid c) = \frac{w_k^{(f)} T_{ck} + 1}{\sum_{k' \in V} w_{k'}^{(f)} T_{ck'} + 1} \tag{6}$$

where $w_k^{(f)}$ is the weight for each token. If the token is a hashtag it equals to $w_h^{(f)}$, if the token is a mentioned user, it equals to $w_u^{(f)}$, in the case that it is a word it equals to $w_w^{(f)}$.

These weights are integer values. We have manually tried different values for the parameters and we present the best and most interesting results.

2) Prior distribution: By definition, the multinomial naive Bayes' model uses Laplace smoothing (Equation 5). This is associated with Dirichlet prior distribution of the vocabulary terms. All the parameters of the Dirichlet distribution equal to 1.

$Dir(a_i)$, $a_i = 1$ for $i \in [1, |v|]$

where $|v|$ is the length of the vocabulary. This is also equivalent to a Uniform prior distribution. We show that we achieve better result, if we choose another prior distribution on the terms of the vocabulary.

We calculate the prior distribution by randomly collecting 10% of the training dataset during cross validation. In particular, we collect the frequencies of the terms that appear in the randomly collected tweets. We add 1 to all the vocabulary terms, in order to cover the cases of the terms that do not appear in the randomly chosen (collected 10%) training tweets. So, each term k is associated with a frequency r_k. r_k is the number of times that term k occurs in the randomly collected 10% of the training tweets. Then we normallize the value of each term r_k by dividing it with the sum of the values for all terms: $R = \sum_k r_k$. R is the sum of all terms frequency occurencies in the randomly collected 10% of the training tweets, in both classes: informative or non-informative. It is how many times all of the terms occur in the 10% of the training dataset that we randomly choose. We do not discriminate how many times each term appears to each class.

So, Equation 6 now is modified as follows:

$$\hat{P}(term_k \mid c) = \frac{w_k^{(f)} T_{ck} + f_k}{\sum_{k' \in V} w_{k'}^{(f)} T_{ck'} + f_{k'}} \tag{7}$$

where f_k is the normalized frequency of term k in the 10% of the training dataset that we randomly collect, i.e. $f_k = \frac{r_k}{R}$.

IV. Evaluation

A. Characteristics of datasets and sets of experiments

Both datasets that we use were crawled from the web in different time periods. There is no overlap in their timestamps. Between the two datasets there are no common messages, users, hashtags, and topics/events/trends. We label the tweets of each dataset manually.

In particular, dataset (A), consists of 20,020 tweets: 7,397 informative and 12,623 non-informative and dataset (B), consists of 10,003 tweets: 4,734 informative and 5,269 non-informative. Dataset (B) is more balanced than dataset (A).

Initially we apply the original binary multinomial naive Bayes' model to our datasets using the Equation 5 from Section III-A, and then we perform four sets of experiments:

1) In Table II, we use the prior distribution of vocabulary terms, as we described in Section III-C2 and Equation 7. In this table, we set all terms weights $w^{(f)}$ equal to 1.
2) In Table III, we use different combination of textual features using dataset (A). We evaluate our method via 10-fold cross validation. The purpose of this set of experiments is to show that the result of our proposed model is affected by the combination of different textual features. We use the Equation 7 from Section III-C2.
3) We use the dataset (A) and we present the best results we have observed for different weights. We evaluate our method via 5-fold cross validation in Table IV and 10-fold cross validation in Table V. We use the Equation 7 from Section III-C2.
4) In the last table, we use dataset (B) and we show the results for the same values of the weights as the above two sets of experiments. We evaluate the result with 10-fold cross validation. We use the Equation 7 from Section III-C2.
5) Finally, in Section IV-F we summarize in figures the most significant results.

We present the results for different weights on the frequencies of the vocabulary terms in the following tables. TP is the number of informative tweets that are correctly detected by the models, TN is the number of non-infomative tweets that are correclty detected. The accuracy cannot be the only evaluation metric, because dataset (A) is not balanced. The non-informative tweets are more than the informative tweets, so accuracy depends more on the non-infomative tweets that are detected. However, we are more interested in the number of informative tweets that is correctly detected, because we want to keep useful information.

Thus, we evaluate our result with the following metrics: recall, specificity, precision, F-measure, F_2-score, $F_{0,5}$-score. There is a tradeoff between the number of informative tweets that are detected (TP) and the number of non-informative tweets that are detected (TN). The F-measure combines precision with recall in a balanced way. F_2 measure weights recall higher, whereas $F_{0,5}$ weights precision higher.

Our goal is to retrieve as many informative messages as possible. Thus, a good result is expected to give high recall, low specificity, and high F_2 score.

B. The original binary multinomial naive Bayes' model

In Table I we present the results of the original binary multinomial naive Bayes' model using the Equation 5 from Section III-A. We compare 5-fold cross validation and 10-fold cross validation on dataset (A) and dataset (B).

In these cases, the original model gives low recall and low F_2 score. This indicates that many informative tweets are not detected.

In Table II we present the original binary multinomial naive Bayes' model with prior distribution, as we described in Section III-C2 and in Equation 7. As we have mentioned in Section III-B, we identify three textual features: hashtags, mentioned users, and the rest of the terms, and we apply the following weights: $w_h^{(f)}$, $w_u^{(f)}$, $w_w^{(f)}$ respectively. At this point we do not use feature combination, thus the weights $w_k^{(f)}$ equal to 1. We randomly draw 10% of the training tweets during cross validation. We observe that the prior distribution of terms helps the model to detect more informative tweets, but still the recall and the F_2 score remain low.

C. Weighted binary multinomial naive Bayes' model, feature combination on dataset (A)

In Table III, we combine the textual features using dataset (A). The purpose of this set of experiments is not to show the best results, but to show that combination of different features affects our model. In all cases we use prior distribution as we described in Section III-C2 and Equation 7, and we evaluate the results with 10-fold cross validation.

In Table III, a weight that equals to 1 means that we ignore this feature. When two features have weight different than 1, it is indicated that these features are combined together.

A general observation is that with proper weights we are able to achieve better results than the original binary multinomial naive Bayes' model without or with prior presented in Tables I and II respectively. The second row of these tables, where we use dataset (A) with 10-fold cross validation, can be compared with Table III.

Also, we see that assignment of weights on mentioned users, $w_u^{(f)}$, gives a higher number of detected informative tweets (TP), which is equivalent with higher recall and higher F_2-score. In addition, we try different values on the rest weights, and in the following sets of experiments we present different combinations of feature weights.

D. Weighted binary multinomial naive Bayes' model on dataset (A)

In Tables IV and V we apply our model to dataset (A). We present the results with 5-fold and 10-fold cross validation.

Our model detects more informative tweets with proper weights on features. As we mentioned above, we manually tested different weights on the frequencies of the vocabulary terms, and we present the cases where the best combination (higher number) of both informative and non-informative tweets is detected in Table IV.

Firstly, we observe that high recall is equivalent with many informative tweets correctly detected, whereas specificity is related with non-informative tweets that are correctly detected.

Secondly, we compare the rows of Table IV. The model returns a high number for TN (10215) for $w_h^{(f)} = 2$, $w_w^{(f)} = 1$ and $w_u^{(f)} = 9$. This is a set of parameters that detects many non-informative tweets. However, this is not considered to be a good result because many informative tweets are not detected. The recall is very low compared to the other rows of the table. We include the first row of this table just to show a set of parameters that gives a high number of TN.

We get the best results when $w_u^{(f)} = 100$ and $w_u^{(f)} = 130$. We explain the results in the rows of the table where $w_u^{(f)} = 130$. The model returns the highest number for TP (6890) for $w_h^{(f)} = 1$ and $w_w^{(f)} = 1$, whereas it returns high number for TN (9749) for $w_h^{(f)} = 1$ and $w_w^{(f)} = 10$. We want to detect as many informative tweets as possible and at the same time we want to ignore as many non-informative as possible. This dataset, dataset (A), is imbalanced, so recall and specificity may be misleading. This is why we are looking both F_2-score and $F_{0.5}$-score, by giving more emphasis on F_2-score. The highest scores are given for the parameters: $w_h^{(f)} = 10$, $w_w^{(f)} = 1$ and $w_u^{(f)} = 130$. We see that mentioned users play a more important role in the detection of informative tweets than hashtags, and hashtags play more important role than the rest of the terms.

This result is better than the original binary multinomial naive Bayes' model without prior in the first row of Table I.

By checking TP and TN in the sixth row of Table IV, we realize that with these parameters we keep 92% of the informative tweets (recall) and that 6,062 tweets are detected correctly as non-informative (TN). Our dataset has 20,020 tweets and we can discard 6,062 tweets without losing much of useful information. That way, we can significanlty decrease the size of the dataset, and the dimensionality of the problem, because the vocabulary of the processed messages will have smaller length.

In the Table IV again, we run into similar conclusions for $w_h^{(f)} = 10$, $w_w^{(f)} = 1$ and $w_u^{(f)} = 100$. The number of TP and TN are very close to the previous results for the same values of the weights as before.

Thirdly, we present the results with 10-fold cross validation in Table V. Again we observe similar behavior for the same values of the parameters.

E. Weighted multinomial naive Bayes' model on dataset (B)

In the last table we present the results of our model in dataset (B). As we mentioned above, this dataset is balanced. We use the same parameters as before and we observe similar behavior as we described in Section IV-D. A main difference is that since this dataset is balanced, the F-score can be more reliable metric. We also look at F_2-score and $F_{0.5}$-score. We have similar results and conclusions for the same set of parameters as we described in Section IV-D where we used dataset (A). Again, in the first row of the table we observe that for the same set of parameters as before: $w_h^{(f)} = 2$, $w_w^{(f)} = 1$, $w_u^{(f)} = 9$ our model detects a high number of TN (4155). However, our model gives the best result when we do not weight words ($w_w^{(f)} = 1$), and we give higher weight to hashtags ($w_h^{(f)} = 10$), and even higher weight to users ($w_u^{(f)} = 100$ or $w_u^{(f)} = 130$). In other words, mentioned users are more important than hashtags, and hashtags are more important than the rest of the terms.

We explain the results in the rows of the table where $w_u^{(f)} = 130$. The model returns the highest number for TP (3944) for $w_h^{(f)} = 10$ and $w_w^{(f)} = 1$. Again, we want to be able to detect as many informative tweets as possible and at the same time we want to ignore as many non-informative as possible. This occurs for $w_h^{(f)} = 10$ and $w_w^{(f)} = 1$.

Results		Evaluation							Dataset	
TP	TN	accuracy	**recall**	specificity	precision	F	$\mathbf{F_2}$	$F_{0.5}$	cv	dataset
4924	11279	0.81	0.67	0.89	0.79	0.72	0.69	0.76	5-fold	(A)
4954	11261	0.81	0.67	0.89	0.79	0.72	0.69	0.76	10-fold	(A)
3427	4343	0.78	0.72	0.82	0.79	0.76	0.74	0.77	5-fold	(B)
3449	4359	0.78	0.73	0.83	0.79	0.76	0.74	0.78	10-fold	(B)

TABLE I: Original binary multinomial naive Bayes' model

$w^{(f)}$			Results		Evaluation							Dataset	
$w_h^{(f)}$	$w_w^{(f)}$	$w_u^{(f)}$	**TP**	TN	accuracy	**recall**	specificity	precision	F	$\mathbf{F_2}$	$F_{0.5}$	cv	dataset
1	1	1	5173	10700	0.79	0.70	0.85	0.73	0.71	0.71	0.72	5-fold	(A)
1	1	1	5222	10686	0.80	0.71	0.85	0.73	0.72	0.71	0.73	10-fold	(A)
1	1	1	3402	4164	0.76	0.72	0.79	0.76	0.74	0.73	0.75	5-fold	(B)
1	1	1	3507	4166	0.77	0.74	0.79	0.76	0.75	0.74	0.76	10-fold	(B)

TABLE II: Original binary multinomial naive Bayes' model with prior distribution of terms.
More informative tweets are correctly detected (TP), but still recall and F_2 is low.

$w^{(f)}$			Results		Evaluation						
$w_h^{(f)}$	$w_w^{(f)}$	$w_u^{(f)}$	TP	TN	accuracy	recall	specificity	precision	F	F_2	$F_{0.5}$
10	1	1	4715	11187	0.79	0.64	0.89	0.77	0.70	0.66	0.74
1	10	1	5211	10707	0.80	0.70	0.85	0.73	0.72	0.71	0.73
10	1	10	5163	10734	0.79	0.70	0.85	0.73	0.72	0.70	0.73
1	10	10	5246	10600	0.79	0.71	0.84	0.72	0.72	0.71	0.72
10	10	1	5119	10779	0.79	0.69	0.85	0.74	0.71	0.70	0.73
1	1	10	5671	10084	0.79	0.77	0.80	0.69	0.73	0.75	0.71
10	10	10	5223	10691	0.80	0.71	0.85	0.73	0.72	0.71	0.73

TABLE III: Feature combination on weighted multinomial naive Bayes' model with prior distribution on dataset (A), 10-fold cv.
Weights on features affect the number of correctly detected informative tweets (TP, recall, F_2).

$w^{(f)}$			Results		Evaluation						
$w_h^{(f)}$	$w_w^{(f)}$	$w_u^{(f)}$	**TP**	TN	accuracy	**recall**	specificity	precision	F	$\mathbf{F_2}$	$F_{0.5}$
2	1	9	5544	**10215**	0.79	0.75	**0.81**	0.7	0.72	0.74	0.71
1	1	100	**6870**	6131	0.65	**0.93**	0.49	0.51	0.66	**0.8**	0.56
10	**1**	**100**	6750	6812	0.68	**0.91**	0.54	0.54	0.68	**0.8**	**0.59**
1	10	100	5708	9995	0.78	0.77	**0.79**	0.68	0.73	0.75	**0.7**
1	1	130	**6890**	5708	0.63	**0.93**	0.45	0.50	0.65	**0.79**	0.55
10	**1**	**130**	6813	6062	0.64	**0.92**	0.48	0.51	0.66	**0.79**	**0.56**
1	10	130	5812	**9749**	0.78	0.79	**0.78**	0.67	0.73	0.76	**0.69**

TABLE IV: Weighted multinomial naive Bayes' model on dataset (A), 5-fold cv.
For weights $w_h^{(f)} = 10$, $w_w^{(f)} = 1$, $w_u^{(f)} = 100$ and $w_h^{(f)} = 10$, $w_w^{(f)} = 1$, $w_u^{(f)} = 130$ we obtain higher recall and F_2 than the original binary multinomial naive Bayes model presented in the first row of Tables I, II.

By checking TP and TN in the sixth row of the last table, we realize that with these parameters we keep 83% of the informative tweets and we can ignore 3,529 tweets without losing much useful information. Similarly to the previous case, we can significantly descrease the size of the dataset and keep most of the useful information of the dataset at the same time.

F. Summarization of the results

Finally, we summarize the most significant results from the experiments we presented above.

We isolate the following metrics: the number of informative messages that are correctly detected (TP), the recall, and the F_2 score. We compare the original binary multinomial naive Bayes' model (Original), the binary multinomial naive Bayes' model with prior and weights $(w_h^{(f)}, w_w^{(f)}, w_u^{(f)}) = (1, 1, 1)$, and binary multinomial naive Bayes' model with prior and weights $(w_h^{(f)}, w_w^{(f)}, w_u^{(f)}) = (10, 1, 130)$. We summarize these results using dataset(A) with 5-fold and 10-fold cross validation, using dataset(B) with 10-fold cross validation.

In Figure 1 we present the correctly detected informative tweets (TP). In our datasets we observe that our variation of the binary multinomial naive Bayes' model with prior and weights $(w_h^{(f)}, w_w^{(f)}, w_u^{(f)}) = (1, 1, 1)$ outperforms the original binary multinomial naive Bayes' model. In addition our variation of the binary multinomial naive Bayes' model with prior and weights $(w_h^{(f)}, w_w^{(f)}, w_u^{(f)}) = (10, 1, 130)$ outperforms the previous two models.

In Figure 2 we present the recall for the same models and

$w^{(f)}$			Results		Evaluation						
$w_h^{(f)}$	$w_w^{(f)}$	$w_u^{(f)}$	**TP**	TN	accuracy	**recall**	specificity	precision	F	$\mathbf{F}_2$	$F_{0.5}$
2	1	9	5578	**10236**	0.79	0.75	**0.81**	0.7	0.73	0.74	0.71
1	1	100	**6870**	6054	0.65	**0.93**	0.48	0.51	0.66	**0.80**	0.56
10	**1**	**100**	6799	6624	0.67	**0.92**	0.53	0.54	0.67	**0.80**	**0.58**
1	10	100	5716	**9982**	0.78	0.77	**0.79**	0.68	0.73	0.75	**0.70**
1	1	130	**6922**	5616	0.63	**0.94**	0.45	0.50	0.65	**0.80**	0.55
10	**1**	**130**	6861	6019	0.64	**0.93**	0.48	0.51	0.66	**0.80**	**0.56**
1	10	130	5858	**9760**	0.78	0.79	**0.77**	0.67	0.73	0.76	**0.69**

TABLE V: Weighted multinomial naive Bayes' model on dataset (A), 10-fold cv.
For weights $w_h^{(f)} = 10$, $w_w^{(f)} = 1$, $w_u^{(f)} = 100$ and $w_h^{(f)} = 10$, $w_w^{(f)} = 1$, $w_u^{(f)} = 130$ we obtain higher recall and F_2 than the original binary multinomial naive Bayes model presented in the second row of Tables I, II.

$w^{(f)}$			Results		Evaluation						
$w_h^{(f)}$	$w_w^{(f)}$	$w_u^{(f)}$	**TP**	TN	accuracy	**recall**	specificity	precision	F	$\mathbf{F}_2$	$F_{0.5}$
2	1	9	3497	**4155**	0.77	0.74	**0.79**	0.76	0.75	0.74	0.75
1	1	100	**3951**	3508	0.75	**0.83**	0.67	0.69	0.76	**0.80**	0.72
10	**1**	**100**	3871	3649	0.75	0.82	0.69	0.71	0.76	**0.79**	**0.73**
1	10	100	3550	**4052**	0.76	0.75	**0.77**	0.75	0.75	0.75	**0.75**
1	1	130	**3941**	3452	0.74	**0.83**	0.66	0.68	0.75	**0.80**	0.71
10	**1**	**130**	3944	3529	0.75	0.83	0.67	0.69	0.76	**0.80**	**0.72**
1	10	130	3574	**4067**	0.76	0.76	**0.77**	0.75	0.75	0.75	**0.75**

TABLE VI: Weighted multinomial naive Bayes' model on dataset (B), 10-fold cv.
For weights $w_h^{(f)} = 10$, $w_w^{(f)} = 1$, $w_u^{(f)} = 100$ and $w_h^{(f)} = 10$, $w_w^{(f)} = 1$, $w_u^{(f)} = 130$ we obtain higher recall and F_2 than the original binary multinomial naive Bayes model presented in the fourth row of Tables I, II.

the same datasets as in the previous figure. The results are similar. Our variation of the binary multinomial naive Bayes' model with prior and weights $(w_h^{(f)}, w_w^{(f)}, w_u^{(f)}) = (10, 1, 130)$ outperforms the previous two models.

In Figure 3 we present the F_2 score. As we mentioned above, the F_2 measure combines precision with recall by weighting recall higher.

V. Conclusion

We separate 'informative' and 'non-informative' tweets from a dataset. We propose a variation of the binary multinomial naive Bayes' model for detecting the informative tweets in a dataset. Our goal is to discard as many non-informative (useless) tweets as possible and at the same time to keep as many informative (useful) tweets as possible. That way the size of the original dataset is decreased but at the same time most of the useful information of the dataset is kept. We recognize three textual features in tweets: hashtags, mentioned users and words. We weight each textual feature and we incorporate weights in the multinomial naive Bayes' model. In addition, we improve the prediction of our proposed model, by incorporating prior distribution of the vocabulary terms. The prior distribution is estimated by tweets randomly drawn from the training dataset. Then, we apply our proposed model to two independent datasets crawled form the web that we have manually labeled. Finally, we evaluate the performance of our model using the following metrics: accuracy, recall, specificity, F-messure with its variations (F_2 score and $F_{0.5}$ score), and we show that our model achieves higher recall and F_2 score than the original binary multinomial naive Bayes' model.

References

[1] http://snowball.tartarus.org.

[2] http://twitter4j.org.

[3] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. *arXiv preprint cs/0006013*, 2000.

[4] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 160–167, New York, NY, USA, 2000. ACM.

[5] A. Ben-Hur and J. Weston. A user's guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.

[6] M.-w. Chang, W.-t. Yih, and C. Meek. Partitioned logistic regression for spam filtering. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 97–105, New York, NY, USA, 2008. ACM.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *Machine Learning*, pages 105–112. Morgan Kaufmann, 1996.

[9] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 295–303, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[10] R. O. Duda, P. E. Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[11] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM, 1998.

[12] J. M. Gómez Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García.

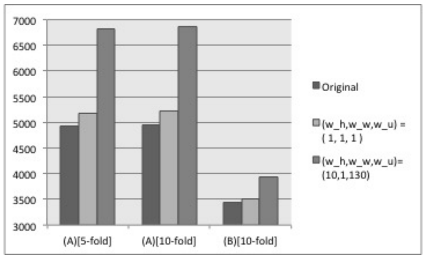

Fig. 1. TP comparison

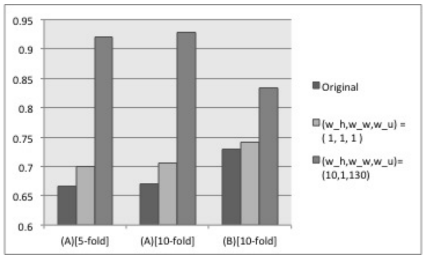

Fig. 2. Recall comparison

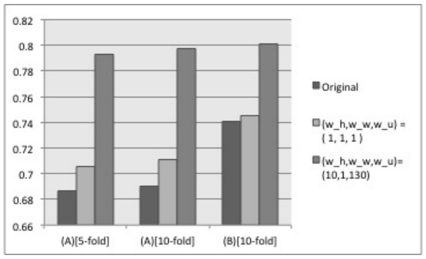

Fig. 3. F_2-score comparison

Content based sms spam filtering. In *Proceedings of the 2006 ACM Symposium on Document Engineering*, DocEng '06, pages 107–114, New York, NY, USA, 2006. ACM.

[13] J. M. G. Hidalgo. Evaluating cost-sensitive unsolicited bulk email categorization. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, SAC '02, pages 615–620, New York, NY, USA, 2002. ACM.

[14] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine learning: ECML-98*, pages 4–15. Springer, 1998.

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

[16] A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[17] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive bayes-which naive bayes? In *CEAS*, pages 27–28, 2006.

[18] A. W. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. *SIGMETRICS Perform. Eval. Rev.*, 33(1):50–60, June 2005.

[19] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[20] M. Pazzani and D. Billsus. Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer Berlin Heidelberg, 2007.

[21] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail, 1998.

[22] K.-M. Schneider. A comparison of event models for naive bayes anti-spam e-mail filtering. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 307–314, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[23] B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[24] D. Sculley and G. M. Wachman. Relaxed online svms for spam filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 415–422, New York, NY, USA, 2007. ACM.

[25] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, Mar. 2002.

[26] I. Uysal and W. B. Croft. User oriented tweet ranking: A filtering approach to microblogs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2261–2264, New York, NY, USA, 2011. ACM.