Short Messages Spam Filtering Using Sentiment Analysis

Enaitz Ezpeleta[1], Urko Zurutuza[1], and José María Gómez Hidalgo[2]

[1] Electronics and Computing Department, Mondragon University
Goiru Kalea, 2, 20500 Arrasate-Mondragón, Spain
{eezpeleta,uzurutuza}@mondragon.edu
[2] Pragsis Technologies
Manuel Tovar, 43-53, Fuencarral - 28034 Madrid, Spain
jmgomez@pragsis.com

Abstract. In the same way that short instant messages are more and more used, spam and non-legitimate campaigns through this type of communication systems are growing up. Those campaigns, besides being an illegal online activity, are a direct threat to the privacy of the users. Previous short messages spam filtering techniques focus on automatic text classification and do not take message polarity into account. Focusing on phone SMS messages, this work demonstrates that it is possible to improve spam filtering in short message services using sentiment analysis techniques. Using a publicly available labelled (spam/legitimate) SMS dataset, we calculate the polarity of each message and aggregate the polarity score to the original dataset, creating new datasets. We compare the results of the best classifiers and filters over the different datasets (with and without polarity) in order to demonstrate the influence of the polarity. Experiments show that polarity score improves the SMS spam classification, on the one hand, reaching to a 98.91% of accuracy. And on the other hand, obtaining a result of 0 false positives with 98.67% of accuracy.

Keywords: SMS, spam, polarity, sentiment analysis, security

1 Introduction

In the era of smartphones and online social networks, instant short message communication tools are growing up very fast. For instance, one of the famous instant messaging applications, WhatsApp, reached 1 billion users on February 2016[3]. Another example is that 6.1 billion people all over the world use an SMS-capable mobile phone on June 2015, so SMS messages can reach more than 6 billion consumers[4]. Such a growth turned those systems in a very attractive objective to malicious companies and groups. Because of this, more and more

[3] https://blog.whatsapp.com/616/One-billion/

[4] http://goo.gl/yqzMDz

illegal activities are being carried out through these communication methods. For example, a gang made at least 5 million Euros over the last decade from a premium-rate SMS messaging scam[5]. Attackers know that there is a huge number of users whose privacy can be threatened in an easy way, sending a direct instant message, such as SMS or WhatsApp messages. Additionally, what makes SMS appropriate for illegitimate activities is the open rate (how many SMSs are opened (or viewed) in a particular SMS campaign) of 98% (for instance, email marketing reports a 22% open rate) [6]. Currently, with a spam proportion of 20-20% of all SMS traffic in China and India, SMS spam is an emerging problem specially in the Middle East and Asia[7].

To deal with this type of problems several tools are designed and developed by researchers all over the word. Those systems are mostly focussed on automatic text classification, but do not take message sentiment into account. It could be considered that a spam message aims at selling products, thus the message should tend to be positive.

The main objective of this paper is to analyze the influence of the polarity in short instant messages spam filtering, testing if sentiment analysis can help on this task. It also aims to provides means to validate the hypothesis that sentiment feature of the short messages can improve the results obtained using common short messages filtering classifiers. Taking into account the publicly available datasets, we focus on SMS messages, which are structurally similar to other instant short messages.

Using a publicly available labelled (spam/legitimate) SMS dataset, we calculate the polarity of each message and aggregate the polarity score to the original dataset, creating new datasets. We compare the results of the best classifiers and filters over the different datasets (with and without polarity).

The remainder of this paper is organized as follows. Section 2 describes the previous work conducted in the area of SMS spam, natural language processing and sentiment analysis. Section 3 describes the process of the aforementioned experiments, regarding sentiment analysis of SMS in terms of polarity and SMS spam filtering. In Section 4, the obtained results are described, presenting the results of the descriptive experiment, and making a comparison between the actual filtering results and the filtering results using the polarity of the messages. Finally, we summarize our findings and give conclusions in Section 6.

2 Related Work

2.1 SMS spam

During the last years attacker have detected that instant message systems are good place to perform malicious activities, specially attracted by the huge amount of users. In this study we are going to focus specially on SMS messages because

[5] http://elpais.com/elpais/2015/04/20/inenglish/1429529298_0 01329.html

[6] http://goo.gl/CaxweY

[7] https://goo.gl/g6R7uW

structurally are similar to other currently more used short messages applications in our area such as Whatsapp or Twitter. Our decision to focus on certain messages is principally based on the public access to a tagged datasets needed to perform our experiments. This give us the possibility to compare our results with previous works. But also we take into account SMS is an emerging problem in the Middle East and Asia, with SMS spam contributing to 20-30% of all SMS traffic in China and India[8].

In [1] authors presented a survey of work on filtering SMS spam and reviewed recent developments in SMS spam filtering. Also a brief discussion about publicly available corpora and availability for future research in the area is shown.

The authors in [2] compare different machine learning methods and indicated that Support Vector Machine was the best. They obtained an accuracy of 97.64% using this method. Furthermore, they offer a public and non-encoded SMS spam collection that can be used by the research community. This study gives us the possibility to test with the same dataset and to compare results.

In other recent studies such as [3] and [4] two-level classifiers are used to obtain better results filtering the spam. But in this study we are going to focus on improving one-level learning-based classifiers.

2.2 Sentiment analysis

Natural Language Processing (NLP) techniques are becoming more and more useful for spam filtering, as it is demonstrated in [5] using sender information and text content based NLP techniques.

Researchers in [6] and [7] confirmed that it is possible to create an application or a system to detect spam in different formats using text mining techniques and semantic language models respectively.

Among all NLP techniques we focus on the use of Sentiment Analysis (SA) to improve the detection of illegitimate short instant messages. This is a different strategy if we compare with the traditional short spam detection techniques which focus on automatic text classification, but do not take SA into account.

During the last years SA has been used in several research areas, although there has been a continued interest for a while. In [8] the most important research opportunities related to SA are described. Based on that, we select document sentiment classification topic as a possible option to short messages filtering.

As it is presented in [9] this area aims at defining if a document is positive or negative based on the its content. In order to improve the classification into positive, negative or neutral, other studies propose supervised learning techniques [10] or unsupervised learning techniques based on opinion words or phrases [11].

Different tools with the objective of helping during the sentiment classification have been proposed in the last years. Lexicon-based methods are interesting tools for our work. Those methods are used to extract the polarity of a certain word or phrase. In [12] a comparison between 8 popular sentiment analysis methods is presented and the author develops a combined method to improve the

[8] https://goo.gl/g6R7uW

results. Centered on short messages, another comparison between lexicon-based approaches is described in [13].

Taking into account those comparisons, we decided to use the publicly available dictionary called *SentiWordNet*. The last version of this tool was presented in [14], which is an improved version of the first dictionary introduced by [15].

3 SMS spam filtering using sentiment analysis

This study has been carried out following the procedure showed in figure 1, where two main experiments are implemented during the work.

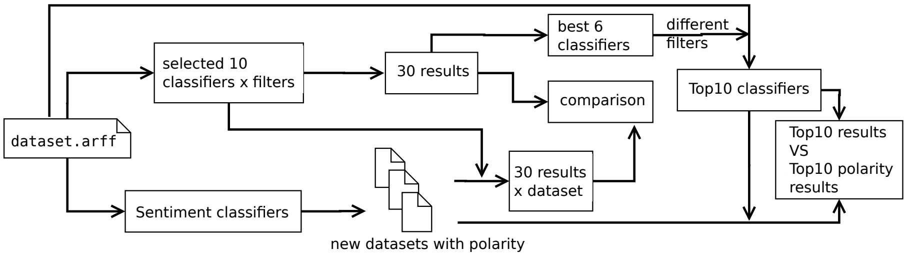

Fig. 1. Full process

The process is divided in two phases. On the one hand, a sentiment analysis of the dataset is done to create new datasets including the measure of polarity.

On the other hand, two experiments are carried out applying different classifiers to the datasets in order to demonstrate the influence of the polarity over the SMS spam filtering.

Those experiments are carried out using the 10-fold cross-validation technique and the results are analyzed in terms of false positive rate and accuracy, being the percentage of testing set examples correctly classified by the classifier.

3.1 Datasets

During this study two different publicly available dataset are used:

- *SemEval-2013*[9]: Introduced in [16]. This dataset contains labelled (positive, negative and neutral) mobile phone messages, and we use it to evaluate the effectiveness of each sentiment classifier during the first phase. Specifically we use positive (492 SMS) and negative (394 SMS) messages.
- *SMS Spam Collection v.1*[10]: Published in [2], it is composed by 5,574 English, real and non-enconded messages, tagged according being legitimate (ham) or spam. Specifically, it contains 747 spam messages and 4,827 ham messages. This dataset is used to carry out the two spam filtering experiments.

[9] https://www.cs.york.ac.uk/semeval-2013/task2/

[10] http://www.dt.fee.unicamp.br/ tiago/smsspamcollection/

3.2 Sentiment analysis

The main objective of this part is to add the polarity of each message to the original dataset *SMS Spam Collection v.1* in order to carry out the experiments. To do that, we analyze different possibilities for sentiment classification of text messages, we evaluate those classifiers using the *SemEval-2013* dataset, we choose the best three classifiers and we apply those to the dataset.

Own sentiment classifier In order to design and implement an own classifier, we decided to use sentiment dictionaries to develop a lexicon based classifier. As it is explained in Section 2.2, among all the possibilities the publicly available *SentiWordNet* is used. It returns to the user the polarity score of a certain word depending on its grammatical properties, and based on this score, the average polarity of the SMS messages is calculated.

Using several settings offered by the dictionary, five different sentiment classifiers have been developed. The first four are called Adjective, Adverb, Verb and Noun. The methodology of each classifier is to consider every word to be a certain part of speech (represented by the name), so we have obtained the polarity of those words that have that grammatical property. For example, in the Noun classifier every word is considered to be a noun, so the polarity of the words is extracted from the dictionary. And the last classifier is called AllPosition where every part of speech per each word is considered to obtain the polarity scores.

TextBlob classifier Another way to do sentiment analysis is to use publicly available resources. In this case, we focus on a simple API for diving into common NLP tasks called TextBlob[11]. More specifically, this tool returns a float value within the range [-1.0,1.0] for the polarity giving a certain string. The system calculates the average score of the sentence using a lexicon based score of each word.

To improve the effectiveness of those classifiers we change settings and select different thresholds (-0.05, -0.1, 0, 0.1, 0.05). The threshold means the point were we consider the polarity score positive or negative, and we use it in the name of the classifier to differentiate from each other.

Comparison between classifiers Once the classifiers have been defined, a tagged (in terms of polarity) publicly available dataset is required to evaluate the effectiveness of the classifiers. Taking into account that SMS messages are the final objective, we decided to choose a dataset composed by SMS messages in order to obtain more reliable results: *SemEval-2013* dataset.

We apply the different sentiment classifiers to the dataset, and we analyze the accuracy of correctly classified messages. In table 1 a comparison between different classifiers and thresholds is shown.

[11] http://textblob.readthedocs.org/

Classifier	**Accuracy**	**Classifier**	**Accuracy**
TextBlob 0.05	0.78	Adjectives	0.66
TextBlob 0.1	0.76	Nouns	0.58
TextBlob -0.05	0.73	TextBlob 0	0.56
AllPositions	0.72	Adverb	0.53
TextBlob -0.1	0.71	Verb	0.52

Table 1. Comparison in terms of accuracy between classifiers

Based on the accuracies offered by the table, the best three classifiers are selected (*TextBlob 0.05*, *TextBlob 0.1* and *TextBlob -0.05*) in order to use those ones to annotate the messages included in *SMS Spam Collection v.1* which has not been annotated for sentiment. As a result, we obtain three new datasets (one per each classifier). The original one and the new three ones are used in the next experiments.

3.3 Spam filtering

To analyze the influence of the polarity over the filtering of SMS messages, the first step is to select 10 representative classifiers and some of the best filter settings for natural language processing. To do that the results presented in [17] are taken into account. Consequently the best five classifiers from the mentioned paper are used in this study. Also the best three settings of the filters are chosen. Moreover, we added more classifiers to the list based on other research studies such as [18]. Final list: Large-scale Bayesian logistic regression for text categorization, discriminative parameter learning for Bayesian networks, complement class Naive Bayes classifier, multi-nominal Naive Bayes classifier, updateable multi-nominal Naive Bayes classifier, decision tree (C4.5), random tree, forest of random trees, Support Vector Machine (SMO) and adaptive boosting meta-algorithm with Naive Bayes.

The next step is to apply those classifiers, combined with the best three filters and settings, to the datasets (with and without polarity) and compare the results.

This first step provides the best classifier for text messages, so in the following phase, the best six classifiers are picked. And the second experiment is carried out applying those classifiers with different combination of filters and settings (56 combinations per classifier) to the datasets. The objective of the combination of this filters and settings is to follow a text mining process in order to compare results and identified the best ones, and those are some of its main details:

- A filter to convert a string to feature vector containing the words. We combine different options: words are converted to lower case, special characters (.,;:$&%=_@()?!+-#[]) are removed using tokenizers; n-gram with min and max grams are created; roots of the words are obtained using a stemmer...
- Attribute Selection: a ranker to evaluate the worth of an attribute by measuring the information gain with respect to the class (spam/ham) is used.

Using those combinations we identify the best ten settings and classifiers for SMS spam classification, and those are applied to the dataset with polarity to compare the results.

4 Experimental Results

In this Section the results obtained during the previously explained experiments are shown. To carry out those experiment the dataset called *SMS Spam Collection v.1* is used.

4.1 Descriptive experiment

Once the dataset is selected, we perform a descriptive experiment of the dataset. The objective of this step is to analyze the polarity of the messages applying our previously selected (Section 3.2.3) sentiment classifiers. This is the point where the polarity extracted during the analysis is inserted in the dataset, creating three new datasets (one per each classifier) and where statistics about the polarity are calculated.

In table 2 the results of the experiment are presented (*Tb 005* means *TextBlob 0.05*, *Tb 01* means *TextBlob 0.1* and *Tb-005* means *TextBlob -0.05*).

	Number of messages						Percentage (%)					
	Tb 005		**Tb 01**		**Tb-005**		**Tb 005**		**Tb 01**		**Tb-005**	
	P	N	P	N	P	N	P	N	P	N	P	N
spam	430	317	408	339	688	53	58	42	55	45	92	7
ham	1,960	2,867	1,859	2,968	4,121	687	41	59	39	61	85	14

Table 2. Sentiment analysis of SMS messages

According to the classifiers it is possible to see, specially in the first two sentiment classifiers, that spam messages are mostly positive while ham messages are more negative. This means that there is a difference between spam and ham messages in terms of polarity, so it can be helpful for improving SMS spam filtering.

Two experiments to see the real influence are carried out.

4.2 First experiment: finding the best SMS spam filtering classifiers

This experiments aims to identify the best SMS spam filtering classifiers in order to use them in the next experiment with more filters and settings. As it is mentioned in Section 4 we choose 10 classifiers and the following filter combinations per each one. Those filters are used to obtain the results presented in the Table 4. The explanation is based on the results obtained in [17].

1. *stwv.go.wtok*: the best result.

2. *i.t.c.stwv.go.ngtok.stemmer.igain*: into the best two algorithms this settings obtained the best result in one, and the second result in the other.
3. *i.t.c.stwv.go.wtok*: appeared in the top ten results and is was the first with NGrams and information gain filter.

The nomenclature used in this list and in the following tables is explained in table 3, where idft means Inverse Document Frequency (IDF) Transformation, tft means Term Frequency score (TF) Transformation and outwc counts the words occurrences.

	Meaning		Meaning
NBMU	Naive Bayes Multinominal Updatable	.stwv	String to Word Vector
NBM	Naive Bayes Multinominal	.go	general options
CNB	Complement Naive	.wtok	Word Tokenizer
BLR	Bayesian Logistic Regression	.ngtok	NGram Tokenizer 1-3
.c	idft False, tft False, outwc True	.stemmer	Stemmer
.i.c	idft True, tft False, outwc True	.igain	Attribute selection using InfoGainAttributeEval
.i.t.c	idft True, tft True, outwc True		

Table 3. Nomenclatures

In table 4 the results of the ten classifiers with the three listed filters are presented. The number in the name represents the type of the filter used.

Analyzing the table we can see that applying classifiers to the original dataset the best one in terms of accuracy is SMO with the third settings. In this case the polarity does not help to improve the result. But applying the Discriminative Parameter Learning for Bayesian Network (DMNBtext) classifier and the first filter to the dataset created using the sentiment analyzer *TextBlob01* the top result is improved. Specifically, an accuracy of 98.76% is obtained. In other two cases the top result is also improved.

Another important information shown in the table is that although it is not the best result in terms of accuracy there is case that must be highlighted. This case is the Bayesian Logistic Regression with the second filter applied to the dataset *TextBlob01*, which obtained an accuracy of 98.67% and 0 false positives.

4.3 Second experiment: SMS spam filtering with polarity score

The second experiment is based on the results obtained in the first one. While the previous aims to search the best algorithms, this one aims to explore most of the possible filter combinations with the best classifiers.

On this way, we identify the best 6 classifiers in table 4 and combined each one with 56 different filter settings. We analyze the achieved results and we get the classifiers that obtained the best ten results in terms of accuracy. The next step is to apply those classifiers to the new datasets that we created using the sentiment classifiers. Those results are shown in table 5.

	Normal		**Tb 005**		**Tb 01**		**Tb -005**	
Spam classifier	**FP**	**Acc**	**FP**	**Acc**	**FP**	**Acc**	**FP**	**Acc**
SMO.3	3	98.73	5	98.71	4	98.67	3	98.73
NaiveBayesMultinomial.3	12	98.69	12	98.62	12	98.69	8	98.69
NaiveBayesMultinomialUpdateable.3	12	98.69	12	98.62	12	98.69	8	98.69
BayesianLogisticRegression.3	5	98.64	6	98.60	5	98.67	4	98.64
DMNBtext.1	10	98.62	7	**98.74**	7	**98.76**	6	98.69
BayesianLogisticRegression.2	2	98.60	3	98.49	**0**	**98.67**	**0**	**98.58**
NaiveBayesMultinomial.1	23	98.53	23	98.51	22	98.53	16	98.64
SMO.2	4	98.53	4	98.58	5	98.58	4	98.53
NaiveBayesMultinomialUpdateable.2	36	98.51	36	98.49	35	98.51	26	98.67
NaiveBayesMultinomialUpdateable.1	29	98.49	25	98.56	26	98.55	16	98.73
ComplementNaiveBayes.1	31	98.44	32	98.40	32	98.39	18	98.64
NaiveBayesMultinomial.2	52	98.37	52	98.33	52	98.31	46	98.48
DMNBtext.2	4	98.28	3	98.31	3	98.37	2	98.31
DMNBtext.3	4	98.28	3	98.31	3	98.37	2	98.31
ComplementNaiveBayes.2	64	98.19	64	98.15	62	98.19	58	98.30
ComplementNaiveBayes.3	56	98.17	48	98.30	48	98.26	19	**98.74**
BayesianLogisticRegression.1	1	97.45	0	96.41	0	96.59	0	96.18
SMO.1	0	97.45	0	97.54	0	97.56	0	97.45
J48.3	54	97.02	58	96.68	56	96.72	54	96.97
J48.2	58	96.90	62	96.56	62	96.54	58	96.86
J48.1	42	96.86	43	96.90	43	96.90	42	96.86
RandomForest.2	0	96.38	0	95.82	2	96.05	0	96.39
RandomForest.3	0	96.21	0	96.27	1	95.91	0	96.29
RandomTree.1	25	95.60	18	95.59	24	95.71	17	95.95
RandomForest.1	0	95.19	0	94.76	0	94.94	0	95.03
RandomTree.3	84	95.16	79	95.43	92	94.90	95	95.25
RandomTree.2	88	95.07	73	95.35	79	95.28	93	95.32
AdaBoostM1.2	167	91.44	166	91.46	166	91.46	167	91.44
AdaBoostM1.3	167	91.44	166	91.46	166	91.46	167	91.44
AdaBoostM1.1	188	91.32	139	91.59	139	91.59	188	91.32

Table 4. Comparison between results

The table shows that a higher accuracy than in the previous experiment is obtained applying new settings of the filters to the original SMS dataset.

Analyzing the data we realize that in half of cases polarity helps to improve the accuracy, and also that by applying the Bayesian Logistic Regression classifier to the dataset created by *TextBlob-005* classifier we improve the best result. While without polarity the best result is 98.85%, using the polarity a 98.91% of accuracy is obtained.

Furthermore, in same cases where better accuracy is not obtained, polarity helps to reduce the number of false positives. Obtaining a percentage of 98.76% and 8 false positives in two cases, reducing from 27 false positives in one case and from 13 in the other.

	Sentiment analyzer							
	None		Tb 005		Tb 01		Tb -005	
Spam classifier	**FP**	**Acc**	**FP**	**Acc**	**FP**	**Acc**	**FP**	**Acc**
NBMU.i.c.stwv.go.ngtok	28	98.85	36	98.73	36	98.73	35	98.74
NBMU.i.t.c.stwv.go.ngtok	27	98.82	17	98.60	16	98.71	**8**	98.76
NBM.i.t.c.stwv.go.ngtok	32	98.78	37	98.74	37	98.74	33	98.78
NBMU.i.t.c.stwv.go.ngtok.stemmer	23	98.78	36	98.71	36	98.71	34	98.74
NBM.c.stwv.go.wtok	13	98.76	33	98.78	32	98.80	28	98.85
NBM.i.t.c.stwv.go.ngtok.stemmer	34	98.76	34	98.74	33	98.74	32	98.76
NBMU.c.stwv.go.wtok	13	98.76	17	98.60	16	98.71	**8**	98.76
CNB.i.t.c.stwv.go.ngtok.stemmer	37	98.73	28	98.85	28	98.85	27	98.82
NBM.i.c.stwv.go.ngtok	37	98.73	26	98.85	25	98.87	22	**98.91**
NBM.i.c.stwv.go.ngtok.stemmer	36	98.73	23	98.80	22	98.82	19	98.82

Table 5. Comparison between Top10 results

5 Conclusions

This work shows that sentiment analysis can help improving short messages spam filtering. We have demonstrated that, adding the polarity obtained during a sentiment analysis of the short text messages, in most of the cases the result is improved in terms of accuracy. Moreover, we have proved our hypothesis obtaining better results with the polarity score than the top result without polarity. (98.91% versus 98.58%). Despite the difference in the percentage does not seem to be relevant, if we take into account the amount of real SMS traffic the improvement is significant.

In addition, a substantial improvement in terms of the number of false positive messages have been achieved in this work. For instance, during the first experiment the best accuracy with 0 false positives is obtained using polarity: 98.67%.

Acknowledgments. This work has been partially funded by the Basque Department of Education, Language policy and Culture under the project SocialSPAM (PI_2014_1_102).

References

1. Delany, S.J., Buckley, M., Greene, D.: Sms spam filtering: methods and data. Expert Systems with Applications **39**(10) (2012) 9899–9908
2. Almeida, T.A., Gómez Hidalgo, J.M., Yamakami, A.: Contributions to the study of sms spam filtering: new collection and results. In: Proceedings of the 11th ACM symposium on Document engineering, ACM (2011) 259–262
3. Narayan, A., Saxena, P.: The curse of 140 characters: evaluating the efficacy of sms spam detection on android. In: Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices, ACM (2013) 33–42

4. Nagwani, N.K., Sharaff, A.: Sms spam filtering and thread identification using bi-level text classification and clustering techniques. Journal of Information Science (2015) 1–13
5. Giyanani, R., Desai, M.: Spam detection using natural language processing. International Journal of Computer Science Research & Technilogy **1** (August 2013) 55–58
6. Echeverria Briones, P.F., Altamirano Valarezo, Z.V., Pinto Astudillo, A.B., Sanchez Guerrero, J.D.C.: Text mining aplicado a la clasificación y distribución automática de correo electrónico y detección de correo spam. (2009)
7. Lau, R.Y.K., Liao, S.Y., Kwok, R.C.W., Xu, K., Xia, Y., Li, Y.: Text mining and probabilistic language modeling for online review spam detection. ACM Trans. Manage. Inf. Syst. **2**(4) (January 2012) 25:1–25:30
8. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. Mining Text Data (2012) 415–463
9. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval **2**(1-2) (2008) 1–135
10. Pang, B., Lee, L., Vaithyanathan, S.: 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, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 79–86
11. Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 417–424
12. Gonçalves, P., Araújo, M., Benevenuto, F., Cha, M.: Comparing and combining sentiment analysis methods. In: Proceedings of the first ACM conference on Online social networks, ACM (2013) 27–38
13. Musto, C., Semeraro, G., Polignano, M.: A comparison of lexicon-based approaches for sentiment analysis of microblog posts. Information Filtering and Retrieval (2014) 59
14. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC. Volume 10. (2010) 2200–2204
15. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. Volume 6., Citeseer (2006) 417–422
16. Nakov, P., Kozareva, Z., Ritter, A., Rosenthal, S., Stoyanov, V., Wilson, T.: Semeval-2013 task 2: Sentiment analysis in twitter. (2013)
17. Ezpeleta, E., Zurutuza, U., Gómez Hidalgo, J.M.: Does sentiment analysis help in bayesian spam filtering? In: Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Sevilla, Spain, April 18-20, 2016, Springer (2016)
18. Kumar, R.K., Poonkuzhali, G., Sudhakar, P.: Comparative study on email spam classifier using data mining techniques. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. Volume 1. (2012) 14–16