Does sentiment analysis help in bayesian spam filtering?

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. Unsolicited email campaigns remain as one of the biggest threats affecting millions of users per day. During the last years several techniques to detect unsolicited emails have been developed. Among all proposed automatic classification techniques, machine learning algorithms have achieved more success, obtaining detection rates up to a 96% [1]. This work provides means to validate the assumption that being spam a commercial communication, the semantics of its contents are usually shaped with a positive meaning. We produce the polarity score of each message using sentiment classifiers, and then we compare spam filtering classifiers with and without the polarity score in terms of accuracy. This work shows that the top 10 results of Bayesian filtering classifiers have been improved, reaching to a 99.21% of accuracy.

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

1 Introduction

The mass mailing of unsolicited e-mails have been one of the biggest threats to Internet security for years. Spam campaigns have been used both for the sale of products such as online fraud. Researchers are investigating many approaches that try to minimize this type of malicious activity that report billionary benefits, being a booming economic sector known as black market or underground economy. The data so far are clear; thanks to the mailing of unsolicited messages, a market share sufficient to enrich a sector devoted to fraudulent activity is achieved. Different attacker communities that worked separately have multiplied their benefits while joining their efforts: new discovered vulnerabilities are sold, which are processed by exploit creators to be included in malicious web sites to spread the malware, which in turn is automatically integrated into large networks of computers, or botnets managed remotely controlled by malicious organizations, which at the end offer services such as: spam campaigns, DDoS, phishing services, a host of fraudulent activities that are generating more business opportunities...

Within the spam problem, most research and products focus on improving spam classification and filtering. According to Kaspersky Lab data, the average of spam in email traffic for the year 2015 stood at 59.2% [2].

To deal with this problem researchers started to design and develop different spam detection systems. Among others, spam filtering techniques are commonly used by both scientific and industrial communities.

This work provides means to validate the assumption that being a spam message a commercial communication, the semantics of its content should be shaped with a positive meaning. Thus, the main objective of this paper is to analyze if the polarity of the message is a useful feature for spam classification. It also aims to validate the hypothesis that polarity feature can improve the results of the typical spam filtering techniques.

On the one hand, we apply the most effective spam classification filters to a known dataset, whereby we obtain the algorithms that better classified the content into spam and ham classes. On the other hand, we analyze different settings of two sentiment classifiers: one API for diving into common natural language processing tasks and other developed by our own. Once we got the best classifiers and settings, we determine the polarity of the messages in the previous dataset, and we create new datasets adding the polarity feature per email. Then a descriptive analysis of the new dataset is carried out. Finally we apply the spam filtering classifiers that obtained the best results in the original dataset to the new ones. The main contribution of this work is that we improve spam filtering rates using the polarity.

The remainder is organized as follows. Section 2 describes the previous work conducted in the area of spam filtering techniques, natural language processing and sentiment analysis. Section 3 describes the process of the aforementioned experiments, regarding Bayesian spam filtering and email polarity classifiers. In Section 4, the obtained results are described, comparing Bayesian filtering results and the filtering results using the polarity of the messages. Finally, we summarize our findings and give conclusions in Section 5.

2 Related Work

2.1 Spam filtering techniques

During the last years several techniques to detect unsolicited emails have been developed [3]. Among all proposed automatic classifying techniques, machine learning algorithms have achieved more success [4]. For instance, different studies such as [5] obtained precisions up to 94.4% using those kind of techniques.

In this work we focus on filters that are able to work with the content of the messages: content-based filters. As authors described in [6] those filters are based on analyzing the content of the emails. There are several different types of content-based spam filters such as heuristic filtering, learning-based filtering and filtering by compression.

Teli et al. presented in [7] a comparison between various existing spam detection methods including rule-based system, IP blacklist, Heuristic-based filters,

Bayesian network-based filters, white list and DNS black holes. They concluded that the most effective, accurate, and reliable spam detection method are the Bayesian based filters.

In [1] some of the content-based filtering techniques are studied and analyzed, and the Bayesian method was selected as the most effective one (classifying correctly the 96.5 % of messages). Furthermore, in [8] authors demonstrated that although more sophisticated methods have been implemented, Bayesian methods of text classification are still useful.

2.2 Sentiment analysis

In [9] Natural Language Processing (NLP) is defined as a theoretical motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. As described in [10] NLP techniques are becoming more and more useful to detect and classify spam messages. Authors says that their model blocked spam messages based on the sender and the content of the text thanks to NLP techniques.

Other studies like [11] demonstrate that it is possible to create applications able to detect spam using text mining techniques. In other words, using process to extract interesting and non-trivial information or knowledge from text documents. In addition, in [12] authors developed a system that integrated a semantic language model to detect opinion spam in different web pages.

While most researchers are working on opinion spam detection using NLP and/or text mining techniques, we focus on the use of NLP and text mining techniques in conjunction with Sentiment Analysis (SA) to improve the detection of spam emails.

In [13] SA or opinion mining is defined as the computational study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes. In SA NLP, text analysis and computational linguistics are used to identify and extract subjective information in source material. As explained in [14], the area of SA has had a huge burst of research activity during these last years, but there has been a continued interest for a while. Currently there are several research topics on opinion mining and the most important ones are explained in [13]. Among those topics we identified the document sentiment classification as a possible option for spam filtering.

The main objective of this area is classifying the positive or negative character of a document [14]. In order to classify such sentiment, some researchers use supervised learning techniques, where three classes are previously defined (positive, negative and neutral) [15]. Some other authors propose the use of unsupervised learning. In unsupervised learning techniques, opinion words or phrases are the dominating indicators for sentiment classification [16].

There are several tools developed during the last years focused on NLP and sentiment analysis. One of the most used for sentiment analysis is known as SentiWordNet. It was presented in [17] and the last version, which is an improved version of the first one, was carried out by Baccianella et al. [18]. It is an enhanced

lexical resource explicitly devised for supporting sentiment classification and opinion mining applications. As they explained in the paper SentiWordNet is the result of the automatic annotation of all the synsets of WordNet according to the notions of positivity, negativity, and neutrality. For instance, author in [19] used SentiWordNet for sentiment classification of reviews obtaining an accuracy of 65.85 % using term counting method.

3 Improving spam filtering using sentiment analysis

Our study has been carried out in three different phases. In the first phase, we apply several spam filtering models with different settings to a certain dataset. Thus, we identify the best classifiers and the best settings to filter spam messages.

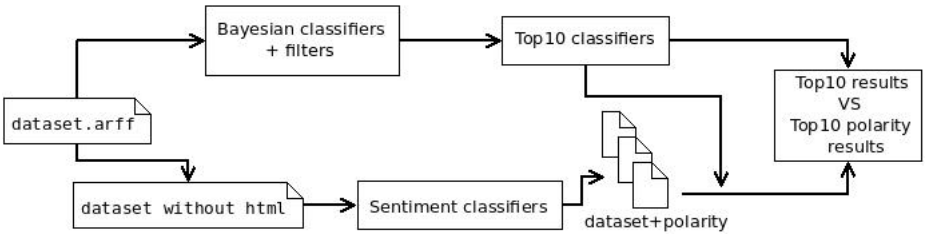

Fig. 1. Full process

As our objective is to improve the best classifiers, in the second phase we work to obtain better results than previously mentioned filters using the polarity of the messages. For that, first of all we need to determine the polarity of each email. To do so we developed our own sentiment classifier, and also used a publicly available API for NLP tasks known as TextBlob[3]. Comparing different settings of each classifier we selected the best ones, which were applied to the dataset used in the previous phase. Using the polarity of each message as new attribute, we carry out a descriptive analysis of these datasets. Finally, the best spam classifiers were applied to the new datasets and made a comparison of the results.

3.1 Bayesian spam filtering

Those filters, which are based on Bayes' Theorem, use Bayes logic to evaluate the header and content of an incoming e-mail message and determine the probability that it constitutes spam.

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

The main objective is to identify the best spam filtering classifiers and the best settings. We apply different combinations of classifiers, filters and settings to compare the results and to select the best ones.

As it is explained in Section 2, Bayesian classifiers are considered as the best techniques to detect and to filter spam messages. Based in this, only the next Bayesian classifiers have been used:

- Bayesian Logistic Regression.
- Complement Naive Bayes.
- DMNBtext.
- Naive Bayes Multinomial Updateable.
- Naive Bayes.
- Naive Bayes Multinomial.
- Naive Bayes Updateable.

Following a text mining process, a set of different filters have been applied to the text. Next, we detail the settings that have been used:

- A filter to convert a string to feature vector containing the words. We use the next options:
  - Words are converted to lower case.
  - A number of words to keep is defined.
  - The maximum number of words and the minimum term frequency is not enforced on a per-class basis but based on the documents in all the classes.
  - Two type of tokenizers are used:
    - One that splits the text removing the special characters.
    - And the other that removes the characters and to split a string into an n-gram with min and max grams.
  - To obtain roots of the words a stemmer based on the Lovins stemmer is used.
  - Weights:
    - IDFTransform False, TFTransform False, outputWordCounts False
    - IDFTransform False, TFTransform False, outputWordCounts True
    - IDFTransform True, TFTransform False, outputWordCounts True
- Attribute Selection: a ranker to evaluate the worth of an attribute by measuring the information gain with respect to the class is used.

At the end of this phase, the best ten settings and classifiers for spam classification have been identified. To do this selection we have use the accuracy of the classifiers, being the accuracy the percentage of testing set examples correctly classified by the classifier.

$$Accuracy = \frac{(True\ Positives + True\ Negatives)}{(Positives + Negatives)}$$

3.2 Sentiment analysis

The objective of this phase is to carry out a sentiment classification of the dataset, in order to later add the polarity of each message as a new feature for spam detection. Later, influence of the polarity in spam filtering and classification.

First a sentiment classifier is needed, so in this task two different options have been considered: to develop our own classifier or to use an existing one. In order to obtain the best possible results, both options have been considered.

Own sentiment classifier. In order to design and implement a classifier, sentiment dictionaries become useful tools, so the commonly used SentiWordNet has been chosen in this case. As shown in researchers have obtained up to a 65% of accuracy using this dictionary.

SentiWordNet is a dictionary that returns to the user the polarity of a certain word depending on its grammatical properties. Using this tool, the average polarity of the email messages have been calculated.

Five sentiment classifiers have been developed with different settings. On the one hand: *Adjective*, *Adverb*, *Verb* and *Noun*. In each classifier every word was considered to be a certain part of speech (depending on the name of the classifier), so we have obtained the polarity of those words that have that grammatical property. For instance: in the *Adjective* classifier every word was considered to be an adjective, so we have obtained the polarity of those words that can be considered as adjectives. And on the other hand, *AllPosition* classifier, which considers every part of speech per each word.

TextBlob classifier. With the objective of comparing different results, TextBlob has been used because it provides a simple API for diving into common NLP tasks. Specifically, giving a string the sentiment analyzer function returns a float value within the range [-1.0,1.0] for the polarity.

Comparison between classifiers. Once the classifiers have been defined, we improve the efficiency of those classifiers by changing settings and selection thresholds. For this work, a previously tagged dataset is mandatory. One commonly used dataset is called *Movie Reviews*[4]. This dataset collects movie-review documents tagged in terms of polarity (positive or negative) or subjectivity rating. Also sentences are tagged with respect to their status or polarity. Among all these options the *polarity dataset v2.0* is used in this task, which is composed by 1,000 positive and 1,000 negative processed reviews introduced in [20]. The objective is to obtain the best accuracy classifying those reviews to find the most efficient settings and thresholds.

In the following table a comparison between the best settings and thresholds is shown. The next criteria is used to define each classifier:

– *TextBlob* means that a classifier based on Textblob library has been used.

[4] http://www.cs.cornell.edu/People/pabo/movie-review-data/

– Some names are followed by a number. This number represent the used threshold for polarity classification. For instance, 0.1 means that every message with score higher than 0.1 has been considered to be positive, and those message with score lower than 0.1 negatives.
– "All without verbs" means that all part of speech but verb has been taken into account during the score calculation.

Table 1. Comparison between classifiers

Name	TP	TN	FP	FN	Accuracy
TextBlob 0.1	719	773	227	281	0.7460
TextBlob 0.05	901	467	533	99	0.6840
Adjectives	775	499	501	225	0.6370
All without verbs	798	460	540	202	0.6290
AllPositions	849	370	630	151	0.6095
Nouns	723	483	517	277	0.6030
TextBlob 0	971	229	771	29	0.6000

Using this information the best three classifiers are selected. To decide which ones can be considered as the best classifiers, the *Accuracy* measure is used.

Descriptive experiments In this step several experiment have been carried out to see how sentiment analysis can affect in spam filtering.

First of all, the selected three classifiers have been applied to the dataset which is explained in the following section. This step offers an idea about the distribution of the email messages in terms of polarity. The number and the percentage of the positive and negative messages has been obtained. Moreover, this information has been used to created one file per each selected classifier, in which the polarity of each message has been added.

Then, we generate a ranking of the most important attributes based on the information gain criteria, and also by analyzing the features that better divide a *J48* classification tree node. Doing that, we preliminarily analyze how the polarity affects in terms of spam filtering.

Predictive experiments During this task the 10 classifiers that obtained the best results in the spam filtering experiments has been applied to the different datasets files. Those files have been created during the descriptive experiments and it consists in a certain spam dataset with the polarity of each message. So, at the end of the experiment the accuracy of the best 10 classifiers applied to the sentimentally classified messages have been obtained.

Finally, all the results are compared. Using the accuracy of each classifiers we demonstrate that the polarity of the messages can help to improve Bayesian spam filtering.

4 Experimental Results

In this Section the results obtained during the previously explained experiments are shown. To carry out those experiment the *CSDMC 2010 Spam corpus*[5] is used. In this dataset 2,949 non-spam messages and 1,378 spam messages are publicly available.

4.1 Bayesian spam filtering experiment

First of all Bayesian classifiers with different settings have been applied to the *CSDMC2010* dataset. In total, 392 different combinations are analyzed. In the following table the best 10 classifiers in terms of accuracy are shown.

Table 2. Top10 Bayesian classifiers

#	Name	TP	TN	FP	FN	Accuracy
1	b.BLR.i.t.c.stwv.go.wtok	1,355	2,936	13	24	99.1451
2	b.DMNBtext.c.stwv.go.wtok	1,362	2,928	21	17	99.1220
3	b.DMNBtext.i.c.stwv.go.wtok	1,362	2,928	21	17	99.1220
4	b.DMNBtext.i.t.c.stwv.go.wtok	1,362	2,928	21	17	99.1220
5	b.DMNBtext.stwv.go.wtok	1,362	2,928	21	17	99.1220
6	b.DMNBtext.c.stwv.go.stemmer	1,360	2,927	22	19	99.0527
7	b.DMNBtext.i.c.stwv.go.stemmer	1,360	2,927	22	19	99.0527
8	b.DMNBtext.i.t.c.stwv.go.stemmer	1,360	2,927	22	19	99.0527
9	b.DMNBtext.stwv.go.stemmer	1,360	2,927	22	19	99.0527
10	b.BLR.i.t.c.stwv.go.ngtok.stemmer.igain	1,351	2,935	14	28	99.0296

In this study the objective is to improve the accuracies of the Bayesian classifier. So, we focus only on these 10 classifiers in the following steps, instead of focus on all combinations used previously.

To understand the nomenclatures used in the table 2 the next summary is presented:

Table 3. Nomenclatures

	Meaning		Meaning
.b	Bayesian classifier	.stemmer	Stemmer
.BLR	BayesianLogisticRegression	.c	idft F, tft F, outwc T
.stwv	StringToWordVector	.i.c	idft T, tft F, outwc T
.go	general options: -L -O -W 10000000	.i.t.c	idft T, tft T, outwc T
.wtok	WordTokenizer	.ngtok	NGramTokenizer 1-3
.igain	Attribute selection using InfoGainAttributeEval		

[5] http://csmining.org/index.php/spam-email-datasets-.html

4.2 Sentiment analysis

Descriptive experiments During the data exploration part, the following results are presented.

Firstly, a sentiment analysis of the dataset has been done. The polarity of each email is identified, this polarity is added to the dataset and statistics of the number of positive and negative spam or legitimate emails are extracted as it is shown in the next table. As we showed that an important number of messages obtained score equal to 0 using *Adjective* classifier, *Adjective Plus* classifier is added in this point. It classified those emails like positive messages. So at the end of this step four different dataset are created, one per each classifier.

Table 4. Sentiment analysis of emails

		Adj		**Adjplus**		**Tb_005**		**Tb_01**	
	Total	*P*	*N*	*P*	*N*	*P*	*N*	*P*	*N*
spam	*1,378*	913	433	945	433	1,044	332	848	516
ham	*2,949*	1,103	1,831	1,118	1,831	1,934	1,009	1,419	1,514
Percentages (%)									
spam	100	66	31	68	31	76	24	62	37
ham	100	37	62	37	62	66	34	48	51

Analysing the data in the table 4 it is possible to see that spam messages are more positive than non-spam or ham messages.

While this experiments gives good results, the results obtained in the rankings and in the trees were not such goods. We observed that polarity appears like a decisive attribute but not like a top one. And different results have been obtained depending on the used sentiment classifier. The best results have been obtained by the dataset analized by *Adjective* classifier. The polarity is ranked in the position 130, and is considered a bit decisive attribute in *J48* decision tree. *Adjective Plus* classifier obtains similar but worse results. And significantly worse results have been obtained by the TextBlob-based classifiers.

Predictive experiments and comparison Once known that polarity can affect in spam filtering, an experiment to demonstrate the real influence are carried out. The best classifiers that appears in table 2 are applied to the four new datasets.

In the following two tables the results are displayed. In both tables the results obtained during the Bayesian filtering are shown for a proper comparison between the results.

In the first one (table 5) the original results are compared with the results obtained applying the filtering classifier to the dataset tagged by our own developed classifier.

Table 5. Comparing original result with the results obtained using own polarity classifiers

	Bayes			*Adjective*			*Adjective+*		
#	**FP**	**FN**	**Accuracy**	**FP**	**FN**	**Accuracy**	**FP**	**FN**	**Accuracy**
1	13	24	99.1451	14	22	**99.1682**	14	23	99.1451
2	21	17	99.1220	24	15	99.0989	24	15	99.0989
3	21	17	99.1220	24	15	99.0989	24	15	99.0989
4	21	17	99.1220	24	15	99.0989	24	15	99.0989
5	21	17	99.1220	24	15	99.0989	24	15	99.0989
6	22	19	99.0527	21	17	**99.1220**	22	16	**99.1220**
7	22	19	99.0527	21	17	**99.1220**	22	16	**99.1220**
8	22	19	99.0527	21	17	**99.1220**	22	16	**99.1220**
9	22	19	99.0527	21	17	**99.1220**	22	16	**99.1220**
10	14	28	99.0296	14	24	**99.1220**	15	23	**99.1220**

As we can see in those first results, *Adjective* sentiment classifier is able to improve the best accuracy of Bayesian algorithms.

In the next table (6) the original results are compared with the results obtained applying the filtering classifiers to the dataset tagged by TextBlob-based classifiers.

Table 6. Comparing original result with the results obtained using TextBlob polarity classifiers

	Bayes			*TextBlob005*			*TextBlob01*		
#	**FP**	**FN**	**Accuracy**	**FP**	**FN**	**Accuracy**	**FP**	**FN**	**Accuracy**
1	13	24	99.1451	13	25	99.1220	14	24	99.1220
2	21	17	99.1220	24	15	99.0989	22	12	**99.2144**
3	21	17	99.1220	24	15	99.0989	22	12	**99.2144**
4	21	17	99.1220	24	15	99.0989	22	12	**99.2144**
5	21	17	99.1220	24	15	99.0989	22	12	**99.2144**
6	22	19	99.0527	21	15	**99.1682**	22	15	**99.1451**
7	22	19	99.0527	21	15	**99.1682**	22	15	**99.1451**
8	22	19	99.0527	21	15	**99.1682**	22	15	**99.1451**
9	22	19	99.0527	21	15	**99.1682**	22	15	**99.1451**
10	14	28	99.0296	14	24	**99.1220**	14	28	99.0296

If we analyze these data, we can realize that polarity helps to improve the accuracy in most cases, and also that the best result obtained using Bayesian spam filtering is improved. While without polarity the best result is 99.1451%, using the polarity feature we reached the rate of 99.2144%.

Focusing on the results of the *TextBlob01* sentiment classifier, we see that in eight out of ten cases the accuracy is better than in the original result. And in case number 9 the same accuracy is obtained.

5 Conclusions

This work shows that the top 10 results of Bayesian filtering classifiers have been improved both generally and per each sentiment classifier.

In addition, considering that the sentiment classifier used is independent from the text, the conclusion is positive. It is supposed that the potential of a training-based one will be better.

As the main conclusions we can say that is possible to improve spam filtering classifiers adding the polarity of the messages. We have demonstrated that sentiment analysis of the emails can help to detect spam emails.

Nonetheless, in future studies we are going to try to confirm our conclusions. To do that, we are going to carry out the same experiments but with different datasets and more learning algorithms. And also it would be interesting to develop and to use a learning-based sentiment classifier calibrated with emails instead of a lexicon-based classifiers.

During following studies we will explore the possibility of using this approach per spam message types. For instance, commercial messages are positive, while other message types like Nigerian scams are negative.

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. Malarvizhi, R.: Content-based spam filtering and detection algorithms-an efficient analysis & comparison 1. (2013)
2. KasperskyLab: Spam and phishing in 2015 q1. `http://www.kaspersky.com/about/news/virus/2015/Spam-and-Phishing-in-Q1-New-domains-revitalize-old-spam` (2015)
3. Saadat, N.: Survey on spam filtering techniques. Communications and Network (2011)
4. Cormack, G.V.: Email spam filtering: A systematic review. Foundations and Trends in Information Retrieval **1**(4) (2007) 335–455
5. Tretyakov, K.: Machine learning techniques in spam filtering. In: Data Mining Problem-oriented Seminar, MTAT. Volume 3. (2004) 60–79
6. Sanz, E.P., Hidalgo, J.M.G., Cortizo, J.C.: Email spam filtering. Advances in Computers (2008) 45–114
7. Savita Teli, S.B.: Effective spam detection method for email. In: International Conference on Advances in Engineering & Technology. (2014)
8. Eberhardt, J.J.: Bayesian spam detection. Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal (2015)
9. Liddy, E.: Natural language processing (2001)
10. Giyanani, R., Desai, M.: Spam detection using natural language processing. International Journal of Computer Science Research & Technilogy **1** (August 2013) 55–58

11. 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)
12. 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
13. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. Mining Text Data (2012) 415–463
14. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval **2**(1-2) (2008) 1–135
15. 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
16. 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
17. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. Volume 6., Citeseer (2006) 417–422
18. 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
19. Ohana, B., Tierney, B.: Sentiment classification of reviews using sentiwordnet. In: 9th. IT & T Conference. (2009) 13
20. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the ACL. (2004)