A Simple Proposal for Sentiment Analysis on Movies Reviews with Hidden Markov Models

Billy Peralta[1(✉)], Victor Tirapegui[2], Christian Pieringer[3], and Luis Caro[2]

[1] Andres Bello University, Santiago, Chile
billy.peralta@unab.cl
[2] Catholic University of Temuco, Temuco, Chile
{vtirapegui,lcaro}@uct.cl
[3] INACAP, Santiago, Chile
cpieringer@inacap.cl

Abstract. Sentiment analysis of texts is the field of study which analyses and studies opinions, sentiments, value judgments, affections and emotions in texts like blogs, news and treating of products, organisations, events and topics. If information on subjective content is required, such as the emotion aroused by an event, computer techniques must be applied to analyse the pattern of public opinion. A common technique for analysing texts is the "Bag of Words", which provides good results assuming that the words are independent of one another. In this work we propose the use of Hidden Markov Chains to determine the polarity of the opinions expressed on movie reviews. We propose a method for simulating hidden states through clustering techniques; we then carry out a sensitivity analysis of the model in which we apply variations to model parameters such as the number of hidden states or the number of words used. The results show that our proposal gives a 3% improvement over the basic model using F-score for real databases of public opinion.

Sentiment analysis of texts is the field of study which analyses and studies opinions, sentiments, value judgments, affections and emotions in texts like blogs, news and treating of products, organisations, events and topics. If information on subjective content is required, such as the emotion aroused by an event, computer techniques must be applied to analyse the pattern of public opinion. A common technique for analysing texts is the "Bag of Words", which provides good results assuming that the words are independent of one another. In this work we propose the use of Hidden Markov Chains to determine the polarity of the opinions expressed on movie reviews. We propose a method for simulating hidden states through clustering techniques; we then carry out a sensitivity analysis of the model in which we apply variations to model parameters such as the number of hidden states or the number of words used. The results show that our proposal gives a 3% improvement over the basic model using F-score for real databases of public opinion.

Keywords: Sentimental analysis · Hidden Markov Models · Clustering

I. Nyström et al. (Eds.): CIARP 2019, LNCS 11896, pp. 152–162, 2019.
https://doi.org/10.1007/978-3-030-33904-3_14

1 Introduction

The growth of Internet-based means of communication like blogs and social networks has promoted interest in sentiment analysis. With the proliferation of opinions, value judgments, recommendations and other forms of expression in the net, on-line opinion has become a sort of virtual currency for companies seeking to sell their products, identify new opportunities and manage their reputations [1]. As companies seek ways to automate processes such as filtering, understanding of conversations, identification of relevant content and appropriate execution, many of them are looking towards sentiment analysis. There are many factors which determine how opinions, value judgments and criticisms are written. Cultural factors, linguistic subtleties and differential contexts make it difficult to interpret a chain of text and obtain subjective information such as a person's emotions or posture with respect to a particular context. Sentiment analysis is an area of study which analyses opinions, sentiments, evaluations, aptitudes and emotions towards entities such as products, services, organisations, individuals, topics, events and their attributes. The problem has applications in a wide range of fields; it is also known as text mining, subjectivity analysis, review mining, emotion analysis, opinion mining and opinion extraction, depending on the use to be given to the information.

During the past ten years, the amount of subjective information posted in the Internet has grown exponentially due to the expansion of Web 2.0. The ability to extract and apply a set of subjective information related with a specific context, using methods such as Hidden Markov Chains, logistic regression, SVM or deep neural networks, make it possible to obtain data to which sentiment analysis can be applied.

Sentiment analysis of texts is acquiring greater importance every year in the Internet; for example, on-line opinion has become an important factor for companies seeking to identify new business opportunities or understand correctly what customers think about the company or its products. Different techniques are used to process the information, for example word filters or identification of relevant content. However these are not entirely appropriate since they are simple filter processes which do not include a search for patterns. Many people are therefore turning to sentiment analysis which offers more appropriate tools for determining text qualities. Some methods of sentiment analysis allow the construction of models which can determine a text's qualities, but as this is a relatively new field of informatics, it is not yet known exactly which methods are best suited to this kind of problem. There are many feasible methods, each with its degree of complexity in implementation. Hidden Markov Chains are a probabilistic model for modelling a Markov process with unknown parameters; more explicitly, we can determine the unknown parameters of the chain through observable parameters. This type of model has many applications, e.g. face [2] and voice [3] recognition. The Bag of Words method on the other hand is used to process natural language and for recovering information in order to represent documents, principally for document classification [4]. The HMM method is based on the relation between two words. This enriches the descriptive power

of the model compared to the Bag of Words method, which ignores word order. Although there are many works on HMM, there appear to be none which propose the application of this technique to sentiment analysis. This research proposes an HMM-based methodology for sentiment analysis. We consider the particular problem of predicting opinions about films and different phases in opinion formation, and especially obtaining HMM states, knowing that these are inaccessible.

2 Sentiment Analysis

Among the existing applications in the field of sentiment analysis, [5] present a model capable of identifying and determining opinions and value judgments about products using the comments posted by product users. The method consists in extracting all the characteristics of the opinion, giving greater weight to words with greater significance for each polarity. These are used as the entry parameters of the model, while irrelevant opinions are discarded. The proposed system, which they call "Wikisent", does not require class distinction for training.

Studies of sentiment analysis using social network data as the data source already exist: [6] carry out a sentiment analysis based on information from Twitter. This work takes tweets and classifies them as "positive", "negative" or "neutral" with respect to a specific topic. Word chains are treated as "trees"; in other words a sentence is taken and broken down to identify all the words by type, e.g. noun, pronoun, verb or adjective. This enables the model to filter the sentences and give greater weight to words more strongly oriented towards a polarity, according to the system, and to ignore those of little importance for classification. Twitter users often make use of emoticons, such as ":), :D, :(), :c", and acronyms, like "gr8t", "lol" and "roft". These are also ways of expressing polarity within a sentence and the model presented in the study cited allows expressions of this kind to be converted into value judgments or emotions.

Looking at existing methods of carrying out sentiment analysis, [7] compare the effectiveness of different classifiers in the context of sentiment analysis. The same work also presents a new method based on hybrid use of multiple classifiers to improve sentiment analysis performance; the idea of this method is that if one classifier fails, the system passes automatically to another until the opinion is classified or no further classifiers exist. The methods presented in the paper are: General Inquirer Based Classifier (GIBC), Rule-Based Classifier (RBC), Statistics Based Classifier (SBD), Induction Rule Based Classifier (IRBC), Support Vector Machine (SVM) and the hybrid classification which combines all these methods.

There are various techniques for sentiment analysis. The work of [8] presents a study of the whole corpus called EmotiBlog. This is a collection of blog entries in which the focus is on detecting subjective expressions in new texts given the context of opinions about telephones. It also shows a comparison between the results of the EmotiBlog corpus and those of a bigger corpus known as JRC; EmotiBlog was found to have a better performance.

More recently, Deep Learning techniques have been applied to the sentiment analysis problem in Twitter [9]. These techniques consist in the application of neural networks using a high number of layers and convolution. Unfortunately, these techniques tend to abstract the reasoning used for the decision.

[10] present different problems which arise in sentiment analysis. It explains the different ways of approaching a set of subjective data, such as: "sentiment classification", which focuses on determining moods in a text; "polarity classification", which is aimed at determining the orientation of words as positive or negative; "subjectivity classification" which focuses on determining how subjective the user's opinion is, referring to a particular context; and "text summaries" which seeks to summarise the information in order to clearly understand opinions within long paragraphs.

Finally, Hidden Markov Models (HMM) have also been applied in sentiment analysis. [3] explains in detail the concept of HMM and its application in voice recognition. [11] apply HMM to sentiment analysis by considering the label information as positive, negative or neutral. The hidden state and the position of words are both considered known. Although this is an interesting work it assumes knowledge of the labels, which requires greater human effort. In the present work, we propose an alternative strategy for applying Hidden Markov Models to sentimental analysis without knowing states labels focusing on the prediction of the polarity of opinions, that is, whether they are positive or negative. For such case, we emulate the state labels using the clusters obtained by a clustering algorithm. We detail our method in the next section.

3 Proposed Method

The use of Hidden Markov Models in sentiment analysis is justified because an opinion consists of a limited number of words which together represent what the person is trying to express; to understand it, a person proceeds to read the words sequentially from left to right, since usually each word is related to the previous one to create a meaningful sentence. Thus the words used to write an opinion can be modelled as observations in a Hidden Markov Model. Nonetheless, this method requires to know the HMM states.

In this work we propose an HMM-based method for sentiment analysis where our main idea is that the HMM state can be modelled approximately by considering a hidden variable given by patterns which are independent of the class of text. In our case, we propose the use of word clusters since these may indicate a significant word pattern. Now we describe the steps of our proposed method:

3.1 Construction of a Word Dictionary

In this stage a dictionary of words is constructed for use by the sentiment analysis models. Only words which are neither numbers nor symbols are included. A unique identifier is assigned to each word. The omission of numbers and symbols allows the size of the training data set to be reduced, thus accelerating the training process.

3.2 Filtering the Dictionary of Words

The object of filtering is to select the words which are most strongly polarised as positive or negative opinions. Firstly the database files are represented to show the occurrence of each word in each file of the database. For example, the first word, whose identifier is 1, will have an occurrence value of 1 if it is present in the text file under review and 0 if it is not present. This process is repeated for all the words in the database for all the opinion files. All the values are then totalled in order to obtain a unique vector representing the occurrence of each word in all the files of a training database.

Once the representations are obtained for both classes, these vectors are taken and the occurrence value of the word in the negative class is subtracted from the occurrence value of the word in the positive class; the result is saved in a new vector whose dimension is the number of words in the database. This can be seen in Fig. 1.

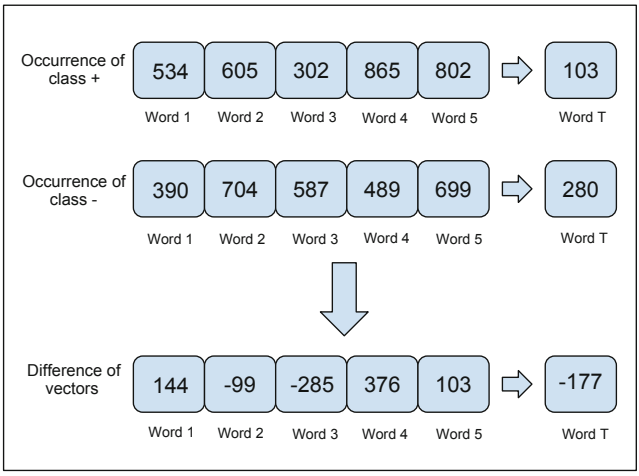

Fig. 1. Result of the count of occurrences in each class. Each word is represented by the difference of occurrences between positive and negative opinions.

The vectors are now ordered such that words with a tendency towards the negative polarity are placed to the left and words with a tendency towards the positive polarity are placed to the right. A set of words taken from either extreme is selected according to a certain criterion and these will be the significant words for our data set. Figure 2 shows a visual example of this step.

The final occurrence vector is used to reduce the number of words; the words from each extreme are selected because they are the words with the greatest difference in occurrence between classifications, and therefore are the most discriminatory.

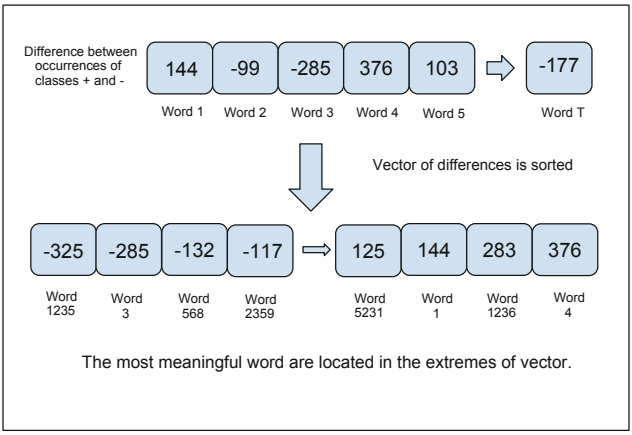

Fig. 2. Ordering the occurrence vectors based on polarity. We expect that a meaningful word for polarity to be placed in the extremes of the list.

3.3 Simulation of the Model's Hidden States

Since the states of the words are unknown, we propose simulating the states using the K-Means clustering algorithm. Using the significant words filtered in the previous step, an occurrence count is done in each file as to establish whether every significant word is present; vectors are created whose length is equal to the number of significant words, and the number of vectors is equal to the number of files per class in the database.

The K-means algorithm is applied to these occurrence vectors, to cluster each opinion file by closeness. This information is used to total the occurrence of each word in each file. The totals are divided by the total number of words in the database in order to calculate the probability of a word in consideration of each centroid, which can be interpreted as the probability of an observation belonging to a state. Figure 3 shows how each word has a certain probability of association with a certain cluster.

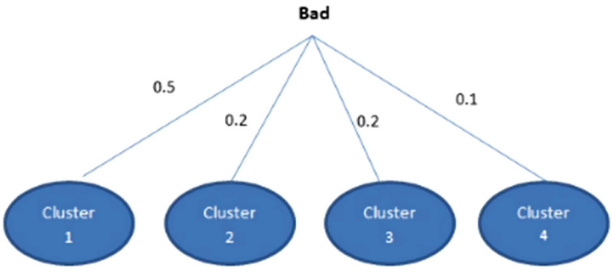

Fig. 3. Simulation of hidden states using the K-Means algorithm. The state is represented by a cluster, which indicates a similar group of words.

By grouping the appearance of all the significant words by the file of each, polarity patterns can be found in the databases which can be used during training as states in hidden Markov models so as to establish a relation with the observations.

3.4 Standardisation of Model Entry Data

Each entry file is divided into sets of ten words. For example, if an opinion file contains 55 significant words, 6 vectors are created which contain the words in groups of 10; the remaining words are placed in the last vector and repeated until the vector is filled. Thus every file usually represents multiple entries into the HMM model.

3.5 Training the Hidden Markov Model

The hidden states of the HMM are represented by the centroids of the clusters, assuming that there are N states. The observations correspond to the significant words in the training set, taking M as the cardinal. The probability distribution of the observations is defined as the probability of the occurrence of a significant word in a cluster. The probability of observations by state is represented stochastically considering the probability of an observation, given a state and considering the frequency of words within clusters. The initial probability of states is random. We apply the Baum-Welch algorithm to train the HMM model.

3.6 Classification of Opinions Using Hidden Markov Models

A Hidden Markov Model is trained for each class, in this case positive and negative. A test text entry S is then sent to each model and the probabilities of occurrence are calculated using the Forward algorithm. Each HMM returns a probability and the model with the highest probability of occurrence will indicate the class of this test text.

$$Class(S) = \arg\max_c p(c)p(S/HMM_c) = \arg\max_c p(c) \prod_{i=1}^{K_S} p(S_i/HMM_c)$$

We divide each file with K_S disjoint parts ($S = \cup_{i=1}^{K_S} S_i$), where each part has 10 words and assumes that they are conditionally independent given the class. Then, each part is processed by an HMM and multiply the probabilities given by the Forward algorithm for each one.

4 Results

In this section we present and discuss the results obtained by implementing the proposed model. The base method used for comparison is the typical Bag of Words model, which considers that all the words are independent. The databases

used in this work are opinions about films; each opinion is pre-defined to a polarity, either positive or negative. Each text file contains one user's opinion about a particular film.

The first database, "Review Movie Dataset", was used in the article "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" [12]. The second database, "Large Movie Review Dataset", is a data set assembled in "Learning Word Vectors for Sentimental Analysis" [13]. Finally, the third database "Movie reviews sentiment", focuses mainly on polarity of movie reviews [14]. The characteristics of the databases used are summarised in Table 1.

Table 1. Details of Datasets used. All the datasets have two classes in these experiments.

Database	Records	Classes	Sample	Average number of words by sample
DB 1: Review Movie	2000	2	2000	632
DB 2: Large Movie R.	50000	2	2000	281
DB 3: Movie Review S.	10662	2	2000	20

4.1 Experiments

In our experiments we tested the sensitivity of the model for the number of words and the number of hidden states in the Markov model. Each modification resulted in a change in the model's behaviour, since when the number of significant words is modified, the number of symbols accepted by the Hidden Markov Model increases or diminishes; while when the number of hidden states is modified, the observations may be associated differently with the states. The results of these experiments are shown in Table 2.

In all the databases the HMM model obtains better results than the Bag of Words method. In the first database (MR), the best configuration for HMM modelling is with 200 words and 10 hidden states, with a performance of 83.5%; the best configuration for the Bag of Words technique was with 100 words giving a performance of 81.7%. In the second database (LMR), the best configuration for HMM modelling is with 400 words and 10 hidden states, with a performance of 83.4%; the best configuration for the Bag of Words technique was with 400 words giving a performance of 79.4%. In the third database (MRS), the best configuration for HMM modelling is with 400 words and 20 hidden states, with a performance of 74.6%; the best configuration for the Bag of Words technique was with 400 words giving a performance of 66.9%. In the experimental phase we sought to maximise the probability for both techniques, achieving improvements of up to 8% for HMM and 4% for Bag of Words. It is possible to go on modifying the number of entry words for the models since a configuration may exist which gives better results. Finally the two techniques require quite different computing times. Bag of Words requires minutes to carry out cross-validation while HMM

Table 2. Results of RM Database with 100, 200 and 400 observations. In general, the use of HMM overcomes BoW in all the tested settings, where the best result for F-value is considering 400 observations.

Method	Accuracy	Precision	F-Value
Considering 100 observations			
HMM(100, 5)	82.4 (2.1)	82.7 (1.8)	82.3 (2.4)
HMM(100, 10)	82.2 (1.7)	83.6 (1.9)	81.8 (2.0)
HMM(100, 20)	**82.7 (2.2)**	**83.0 (1.8)**	**82.6 (2.5)**
BoW(100)	81.7 (2.2)	81.4 (2.7)	81.7 (2.3)
Considering 200 observations			
HMM(200, 5)	82.9 (2.4)	83.0 (2.5)	82.9 (2.4)
HMM(200, 10)	**83.5 (2.6)**	**83.9 (2.8)**	**83.3 (2.7)**
HMM(200, 20)	83.4 (2.4)	83.6 (2.4)	83.3 (2.5)
Bag-of-Words(200)	80.8 (2.0)	80.9 (2.4)	83.3 (2.5)
Considering 400 observations			
HMM(400, 5)	**83.3 (2.6)**	**82.3 (2.2)**	**83.5 (2.8)**
HMM(400, 10)	83.1 (2.3)	82.8 (2.2)	83.2 (2.4)
HMM(400, 20)	83.0 (2.2)	81.6 (1.8)	83.3 (2.6)
Bag-of-Words(400)	81.5 (2.2)	82.3 (2.6)	81.7 (2.2)

Table 3. Results of LMR dataset with 100, 200 and 400 observations. In general, the use of HMM overcomes BoW in all the tested settings, where the best result for F-value is again considering 400 observations.

Method	Accuracy	Precision	F-Value
Considering 100 observations			
HMM(100, 5)	78.2 (2.1)	78.1 (2.9)	78.2 (1.9)
HMM(100, 10)	**78.5 (2.2)**	**78.9 (3.7)**	**78.4 (1.8)**
HMM(100, 20)	78.2 (2.4)	78.7 (2.8)	78.0 (2.3)
Bag-of-Words(100)	76.2 (1.9)	76.3 (2.7)	76.1 (1.7)
Considering 200 observations			
HMM(200, 5)	79.9 (2.2)	80.4 (3.2)	79.7 (2.1)
HMM(200, 10)	**80.3 (1.6)**	**79.5 (2.8)**	**80.6 (1.3)**
HMM(200, 20)	79.7 (2.0)	80.1 (2.8)	79.6 (2.0)
Bag-of-Words(200)	75.8 (3.2)	75.0 (3.4)	75.5 (3.3)
Considering 400 observations			
HMM(400, 5)	82.7 (2.6)	82.6 (3.1)	82.7 (2.5)
HMM(400, 10)	**83.4 (2.6)**	**83.7 (3.6)**	**83.3 (2.3**)
HMM(400, 20)	83.0 (2.6)	83.9 (2.9)	82.8 (2.6)
Bag-of-Words(400)	79.4 (1.9)	79.0 (2.0)	79.5 (2.1)

Table 4. Results of MRS dataset with 100, 200 and 400 observations. In general, the use of HMM overcomes BoW in all the tested settings, where the F-value best result considers 400 observations.

Method	Accuracy	Precision	F-Value
Considering 100 observations			
HMM(100, 5)	65.5 (3.4)	65.0 (4.0)	66.4 (2.7)
HMM(100, 10)	**66.4 (2.3)**	**67.7 (3.5)**	**66.3 (1.8)**
HMM(100, 20)	66.1 (2.8)	65.2 (3.2)	67.3 (2.3)
Bag-of-Words(100)	63.9 (2.2)	64.0 (2.3)	63.7 (2.7)
Considering 200 observations			
HMM(200, 5)	69.5 (3.4)	68.7 (3.5)	70.1 (3.3)
HMM(200, 10)	**70.0 (3.0)**	**70.3 (2.9)**	**69.7 (3.3)**
HMM(200, 20)	69.0 (3.9)	68.2 (3.9)	69.7 (3.7)
Bag-of-Words(200)	65.7 (2.8)	65.6 (3.5)	66.0 (2.5)
Considering 400 observations			
HMM(400, 5)	73.7 (2.5)	73.6 (3.1)	73.8 (2.3)
HMM(400, 10)	73.7 (2.8)	74.1 (3.6)	73.6 (2.5)
HMM(400, 20)	**74.6 (3.3)**	**74.8 (4.2)**	**74.6 (2.9)**
Bag-of-Words(400)	66.9 (3.6)	66.6 (4.3)	67.3 (2.9)

requires at least 1 h, or even more depending on the number of words used as possible entry values and the number of model entry data. The details of our implementation and the used datasets are available in[1] (Tables 3 and 4).

5 Conclusions

We conclude that the Hidden Markov Models technique is able to model the sentiment analysis, in particular the proposed technique can be used to classify the polarity in opinions about movies. Moreover, we also compared the proposed HMM-based technique with the Bag of Words method, obtaining better results with the proposed technique in all the real databases tested, indicating that the proposed method is competitive for sentiment analysis. Variations of the HMM model were applied to determine whether better performance could be obtained. The improvements varied with different configurations for each database, from which we conclude that it is necessary to experiment with the parameters of the HMM to find the best configuration for each database tested. As a future work, we propose to use a more powerful model as Hidden Semi-Markovian Models and mixture of Hidden Markov Models.

[1] https://drive.google.com/open?id=1Ke84q27ovr3bnfxC4BeDDyC1tqHcosf1.

References

1. Mukherjee, S., Bhattacharyya, P.: Sentiment analysis: a literature survey. Technical report, Indian Institute of Technology, Bombay (2013)
2. Nefian, A.V., Hayes III, M.H.: Face detection and recognition using hidden markov models. In: Proceedings of International Conference on Image Processing, vol. 1, pp. 141–145. IEEE (1998)
3. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
4. Guzella, T.S., Caminhas, W.M.: A review of machine learning approaches to spam filtering. Expert Syst. Appl. **36**(7), 10206–10222 (2009)
5. Mukherjee, S., Bhattacharyya, P.: Feature specific sentiment analysis for product reviews. In: Gelbukh, A. (ed.) CICLing 2012. LNCS, vol. 7181, pp. 475–487. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28604-9_39
6. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media, Association for Computational Linguistics, pp. 30–38, June 2011
7. Prabowo, R., Thelwall, M.: Sentiment analysis: a combined approach. J. Informetrics **3**(2), 143–157 (2009)
8. Fernández, J., Boldrini, E., Gómez, J.M., Martínez-Barco, P.: Análisis de sentimientos y minería de opiniones: el corpus emotiblog. Procesamiento del lenguaje natural **47**, 179–187 (2011)
9. Tang, D., Wei, F., Qin, B., Liu, T., Zhou, M.: Coooolll: a deep learning system for twitter sentiment classification. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 208–212 (2014)
10. Kim, H.D., Ganesan, K., Sondhi, P., Zhai, C.: Comprehensive review of opinion summarization. Technical report, University of Illinois at Urbana-Champaign (2011)
11. Jin, W., Ho, H.H., Srihari, R.K.: Opinionminer: a novel machine learning system for web opinion mining and extraction. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1195–1204. ACM (2009)
12. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. ACL 2004, Association for Computational Linguistics (2004)
13. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, Association for Computational Linguistics, pp. 142–150, June 2011
14. Pang, B., Lee, L.: Sentence polarity dataset v1.0 (2017). https://www.kaggle.com/nltkdata/sentence-polarity. Accessed 10 Jan 2017