Sentiment analysis and opinion mining: on optimal parameters and performances

Björn Schuller, Amr El-Desoky Mousa, Vasileios Vryniotis

Angaben zur Veröffentlichung / Publication details:

Schuller, Björn, Amr El-Desoky Mousa, and Vasileios Vryniotis. 2015. “Sentiment analysis and opinion mining: on optimal parameters and performances.” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5 (5): 255–63. https://doi.org/10.1002/widm.1159.

Sentiment analysis and opinion mining: on optimal parameters and performances

Björn Schuller,[1*] Amr El-Desoky Mousa[2] and Vasileios Vryniotis[1]

Sentiment analysis is the task of identifying the polarity and subjectivity of documents using a combination of machine learning, information retrieval, and natural language processing techniques. The problem is studied within the scope of statistical machine learning. Different feature selection methods, dimensionality reduction algorithms and classification techniques are investigated and compared. The main focus of this work is on finding the factors that affect the accuracy of learnt models. Extensive statistical analysis is performed to identify the best algorithmic configurations. Moreover, a novel approach is introduced based on long short-term memory recurrent neural network language models that do not require any special preprocessing or feature selection. Finally, benchmark results are presented on seven well-known datasets from different domains.

INTRODUCTION

Sentiment analysis is the task of automatically identifying the polarity and the subjectivity of documents. With the fast development of Internet technology, users became able not only to consume information from the web but also easily generate content, publish reviews, and express their opinions. Therefore, companies, governments, and organizations became increasingly interested in analyzing these big data in order to automatically monitor the public opinion and assist decision making. As a result, sentiment analysis attracted the interest of many researchers.

The task of sentiment analysis can be seen as a text classification problem. It involves the parsing and tokenization of raw documents, extraction of important features, and classification of the documents into categories. The classes can be described by continuous primitives such as valence, a polarity state (positive or negative attitude), or a subjectivity state (objective, subjective). Such detection of the polarity is one of the hardest text classification problems due to the existence of noise, sentimental drifts, vocabulary changes, idioms, irony, jargon, abbreviations, and domain specific terminology.

Sentiment analysis aims at developing fast, scalable and robust algorithms that are used in *Opinion mining*. To solve this problem, several different approaches have been proposed including statistical methods, grammatical rules and lexicon based techniques. For example, Pang et al.[1] have considered different classifiers, such as Naïve Bayes (NB), maximum entropy (MaxEnt), and support vector machines (SVM) to detect the polarity of movie reviews. Pang and Lee[2] have combined polarity and subjectivity analysis and proposed a technique to filter out the objective sentences of the text. Taboada et al.[3] and Ding et al.[4] have focused on lexicon-based techniques that use the sentimental orientation score of each word to evaluate the document polarity and the potency toward a topic.

Language models (LMs) have also been used for sentiment analysis. For example, Liu et al.[5] have used an emoticon smoothed unigram LM to perform sentiment classification. Hu et al.[6] have built different LMs from subjective sentences. Then, a classifying

*Correspondence to: bjoern.schuller@imperial.ac.uk

[1]Department of Computing, Imperial College London, London, UK

[2]Machine Intelligence & Signal Processing group, MMK, Technische Universität München, Munich, Germany

Conflict of interest: The authors have declared no conflicts of interest for this article.

function based on the generation of a test document has been shown to outperform SVMs on a Chinese digital product review corpus.

Furthermore, artificial neural networks (ANNs) have recently been considered for the sentiment problem. For example, Dong et al.[7] have used a recurrent neural network (RNN) classifier on the Stanford sentiment Treebank corpus. Dhande and Patnaik[8] have combined an NB with a feed-forward neural network (FFNN) classifier on a movie review dataset. Santos and Gatti[9] have used a deep convolutional neural network (CNN) that exploits from character- to sentence-level information on the Stanford Twitter sentiment corpus. Socher et al.[10] have used recurrent neural tensor networks (RNTN) to predict the compositional sentimantic effects on sentence level.

Other aspects beyond classifier optimization include, e.g., handling of irony and sarcasm as polarity reversers in sentiment analysis, which has recently received research attention. For example, Barbieri and Saggion[11] detected irony in a Twitter dataset using Random Forests and Decision Trees. Liebrecht et al.[12] used the Balanced Winnow Classification algorithm with uni-, bi-, and trigram features to detect irony on Dutch tweets. Reyes and Rosso[13] presented a new linguistic model that represents irony by three conceptual layers using eight textual features. A comprehensive survey text that covers all important topics and latest developments in the field of sentiment analysis and opinion mining can be found in the book by Liu.[14] Further topics discussed in this book include unsupervised sentiment classification, aspect-based sentiment analysis, analysis of comparative opinions, opinion spam detection. However, these cannot be dealt with here due to space restrictions.

In the following, we discuss the typical steps of performing sentiment analysis including the document tokenization process, feature selection methods, dimensionality reduction algorithms, and sentiment classification techniques. Then, we present our experiments by description of the used datasets, parameter optimization experiments, and benchmark results on large datasets. This includes results for the newly proposed approach based on memory-enhanced RNNs.

Document Tokenization

The target of *document tokenization* is to convert documents from sequences of words to lists of features (also called terms). Here, the bag-of-words framework is used and terms are extracted in a form of *m-grams*. The tokenization algorithm works as follows for each document separately:

1. Perform the necessary clean-up of the document, removing accents, multiple spaces, tokenizing URLs and punctuation.
2. Within a predefined window length generate all possible keyword combinations (unigrams, bigrams, trigrams, etc.). Clustering of morphological variants (stemming) may precede this.
3. For the keywords within the window, locate if they appear regularly in the whole text and count their frequencies normalized by the occurrences in all documents.
4. Filter out the *m*-grams that have frequencies less than a predefined threshold and add the remaining terms in the list of selected terms.
5. Move the window across the document by one word and repeat the process from step 2 until the end of the document is reached.

FEATURE SELECTION

The target of feature selection is to reduce the amount of available features by selecting the ones that are more important for the analysis. Initially, the features/terms of the documents are evaluated based on a particular criterion and then a subset of them are selected. We will next describe those considered in our experiments, but note that obviously, further ones exist and are used in the field such as the entropy feature selection algorithm.

Mutual Information

The mutual information (MI) measures how much information a particular term contributes to making the correct classification decision on a class.[15] It is defined as:

$$\mathrm{MI}(T, C) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} P\left(T = e_t, C = e_c\right) \log \frac{P\left(T = e_t, C = e_c\right)}{P\left(T = e_t\right) P\left(C = e_c\right)}, \quad (1)$$

where T is a random variable that takes values ($e_t = 1$: document contains term t, $e_t = 0$: document does not contains term t); and C is a random variable that takes values ($e_c = 1$: document is in class c, $e_c = 0$: document is not in class c). The probabilities in (1) are estimated by dividing the counts of underlying events by the total number of documents. The top scoring terms are selected as features.

Chi-square

The Chi-square feature selection uses a nonparametric hypothesis test that evaluates the independence of two events.[15] Here, the two events are the occurrence of the term and the occurrence of the class. Thus, we test if the assignment of a document to a particular class depends on the occurrence of a particular term in the document. Thus, terms are ranked with respect to:

$$\chi^2(t,c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{\left(N_{e_t e_c} - E_{e_t e_c}\right)^2}{E_{e_t e_c}}, \tag{2}$$

where e_t and e_c are defined as in Eq. (1). N is the observed frequency in the dataset and E is the expected frequency. For example, E_{11} is the expected frequency of t and c occurring together in a document assuming that term and class are independent.

Term Frequency Inverse Document Frequency Score

The term frequency inverse document frequency (TF-IDF) score is based on the observation that the important terms of a document have high frequencies in that document and less frequencies in the whole dataset. It is defined as:

$$\text{TFIDF}\left(t,d\right) = \text{TF}\left(t,d\right) \cdot \frac{N}{\left|d_i \in D : t \in d_i\right|}, \tag{3}$$

where TF(t, d) is the number of times that the term t appears in a particular document d, *IDF* is the total size of documents N divided by the number of documents in the whole dataset D that contain the term t. It can be easily noticed that this score does not exploit any class specific information.

DIMENSIONALITY REDUCTION METHODS

The dimensionality reduction is the process of converting the observations from a high- to a low-dimensional space. This can be achieved by grouping the terms in semantic categories/topics, or by grouping the documents in clusters.

Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) is a generative probabilistic hierarchical topic selection model used to identify the set of topics of documents. It can be seen as a clustering technique for grouping documents by modeling the co-occurrence of terms within the topics. It provides a list of activated topics along with their weights for each document. Then, dimensionality reduction is achieved by running LDA on the document dataset and selecting a relatively large number of topics. Thus, Instead of representing texts as bags-of-words, they are represented as mixtures of activated topics. A detailed description of the LDA probability model is given by Griffiths and Steyvers.[16]

Dirichlet Process Mixture Model clustering

The Dirichlet process mixture model (DPMM) is a Bayesian nonparametric method that can be used to perform clustering. It does not require a predefined number of clusters and it is fully Bayesian, thus it describes the generation process of the documents. First, a clustering is done for the original dataset. Then, probabilities of assigning each record to every cluster are calculated. These probabilities are usually significantly lower than the original features and thus can be used to represent the original documents. A detailed description of the model is given by Fan and Bouguila.[17]

CLASSIFICATION METHODS

Naïve Bayes

The NB is a probabilistic classifier that assumes the conditional independence of the features.[15] It uses the maximum-a-posteriori (MAP) rule in order to classify the new entries:

$$c_{\text{map}} = \arg\max_{c \in C} P(c|x) = \arg\max_{c \in C} P(c) \prod_{j=1}^{d} P\left(x_j|c\right). \tag{4}$$

The NB classifier has several different variations. The multinomial NB uses the term frequencies to estimate the required probabilities. Whereas, Binarized NB uses frequencies clipped to 0 or 1. However, the Bernoulli NB takes into account not only the occurrence of a term in the document but also its absence from the document.

Maximum Entropy

The MaxEnt is a discriminative probabilistic classifier that aims at selecting the most uniform model that best fits the training data.[18] In other words, it selects the model that maximizes the information entropy subject to the constraints imposed by the information. According to the MaxEnt criterion, the probability of classifying document x as class y is written as:

$$p^{*}(y|x) = \frac{\exp\left(\sum_{j} \lambda_j f_j(x, y)\right)}{\sum_{y} \exp\left(\sum_{j} \lambda_j f_j(x, y)\right)}, \tag{5}$$

where $f_j(x,y)$ is an indicator feature function. There are as many feature functions as the number of word-class combinations in the dataset. Thus, f_j is equivalent to $f_{w,c}$ and it is activated when the word w appears in document x and the class y is equal to c. Thus, $f_{w,c} = 1$ if $y = c$ and $w \epsilon x$; otherwise $f_{w,c} = 0$. The model parameters λ_j are the weights of the functions and can be estimated using iterative methods, such as the Improved Iterative Scaling (IIS) or the Generalized Iterative Scaling (GIS).

Multinomial Logistic Regression

The multinomial logistic regression is a generalization of logistic regression for multiple classes.[19] It is also known as SoftMax regression due to its exponential hypothesis function. It is very similar to the MaxEnt model; however, the problem is formulated differently and another algorithm is used to estimate the weights. Each word has a different weight for each class. If the number of classes is k and the size of the vocabulary is d, then the model requires $\theta_1, \theta_2, \ldots, \theta_k \epsilon R^{d+1}$ as parameters (with an additional dimension for the intercept terms). The documents can be represented as sparse vectors with term frequencies or binary occurrence/absence values. The classification probability is given by Eq. (6). The theta parameters can be approximated using the gradient descent method:

$$p\left(y^{(i)} = j|x^{(i)}; \theta\right) = \frac{\exp\left(\theta_j^T x^{(i)}\right)}{\sum_{1 \le l \le k} \exp\left(\theta_l^T x^{(i)}\right)}. \tag{6}$$

Ordinal Logistic Regression

The ordinal logistic regression is an extension of the logistic regression when the predicted variable is discrete and ordered.[20] Given a training dataset of (x_i, y_i) where x_i is the input, y_i is the output and $x_i \epsilon R^d$, then instead of taking into account the values of y_i, we focus on their order. Assuming that there are k different classes, the parameters of the model consist of a single weight vector $w \epsilon R^d$ and a threshold vector $\theta \epsilon R^{k+1}$ which segments the real line into k sections. The threshold θ is a nondecreasing vector with $\theta_1 \le \theta_2 \le \ldots \le \theta_{k-1}$ assuming that $\theta_0 = \theta_k = -\infty$.

The immediate-threshold ordinal logistic regression model assumes that all the hyperplanes that separate the different classes are parallel and thresholds θ are used in order to decide to which class a particular observation should be assigned. Thus, the model assumes that all the observations lie across a regression line, and depending on the position of the point on this line, it is classified on different categories. The logistic ordinal regression models the cumulative probability as in Eq. (7). The theta parameters can also be approximated using the gradient descent method:

$$p\left(y \le j|x_i\right) = \frac{1}{1 + \exp\left(x_i^T w - \theta_j\right)}. \tag{7}$$

Recurrent Neural Network LMs

The LM is a statistical model that estimates the probability distribution $p\left(w_1^N\right)$ over word sequences w_1^N. It is considered a very useful model in many natural language processing applications. Usually, the assumption of the $(m-1)$th order Markov process is used, in which the probability of a current word is estimated based on the preceding $(m-1)$ history words:

$$p\left(w_1^N\right) = \prod_{n=1}^{N} p\left(w_n|w_1^{n-1}\right) \approx \prod_{n=1}^{N} p\left(w_n|w_{n-m+1}^{n-1}\right). \tag{8}$$

To utilize the LMs in text classification, a different LM is estimated for each class of documents and the perplexities of the test documents with each LM are used to decide for class assignment. The advantages of this approach are that it does not depend on the bag-of-words principle and does not require any prior application of feature selection or dimensionality reduction methods. The perplexity of a model p over some text corpus $T = w_1^N$ is defined as:

$$PP\left(w_1^N\right) = \left[\prod_{n=1}^{N} p\left(w_n|w_1^{n-1}\right)\right]^{-1/N}. \tag{9}$$

A conventional approach to estimate the LM probabilities is the backoff m-gram model which is based on the smoothed count statistics collected from the training text. A major drawback of this model is that it backs-off to a shorter history whenever insufficient statistics are observed for a given m-gram. Recently, RNNs are utilized to provide long-span LMs that take into account all the predecessor words. These models are improved using a special type of RNN called Long Short-Term Memory (LSTM) RNN[21] that avoids the vanishing gradient problem during the application of the famous Back-Propagation Through Time (BPTT) training algorithm. Here, we

TABLE 1 | Overview of the Used Datasets

Dataset Name	#+	#−	Reviews	Abbr.
Amazon Multidomain Dataset[15]	4000	4000	Products	AMAZ
Polarity Dataset v2.0[16]	1000	1000	Movies	PD2
News and Blogs Dataset[17]	288	255	Various	N&BD
Sentence Polarity Dataset v1.0[18]	5331	5331	Movies	SPD1
Amazon MP3 Dataset[19]	21,987	6482	Music	AMP3
Large Movie Review Dataset v1.0[20]	25,000	25,000	Movies	LMRD
Twitter Dataset[21]	400,000	400,000	Tweets	TWIT

#+, number of positive instances; #−, number of negative instances; reviews, type of review; abbr., abbreviation.
The first four datasets are used for parameter optimization.

propose this approach in the context of sentiment analysis including a solution based on our LSTM implementation.[22]

EXPERIMENTS

Here, an extensive series of tests are discussed aiming at finding the relevance of certain parameters and revealing their optimal choice. The optimization of parameters is performed on four small-medium sized datasets. To investigate the effects on accuracy, all configurable variables are identified, then their Cartesian product is taken and each combination is tested on the training datasets. A large number of validation metrics are estimated for each model including: the overall accuracy, F1, Recall and Precision for each class and the average F1, Recall and Precision across all classes. However, preference is given to the average F1 score over individual classes. A 10-fold cross validation is used.

An exception to the above plan occurs in the application of the LSTM LM model which is only tested on a single large dataset (the *Large Movie Review Dataset v1.0*).

Datasets

Table 1 provides a summary of the used datasets. All of them are well-known, publicly available, and extensively analyzed in previous publications. They consist of reviews from different domains and various texts coming from blogs and news websites. The first four small-/medium-sized datasets are used for parameter tuning and to analyze the effect of various factors on the accuracy.

Experimental Results

One of the most important factors to tune is the number and type of *m*-grams extracted from the documents during the tokenization. Unigrams as well as bigrams and trigrams are used. Different classification methods are considered, each of which can be further configured: The Multinomial NB can weight the terms based on their frequencies, while the MaxEnt and the regression models are configured by the number of training iterations. Another possible configuration is the feature selection method parameterized by the number of selected features. Additionally, the Chi-square algorithm requires the determination of the level of statistical significance. Moreover, we have the option to apply a dimensionality reduction method. For DPMM, we need to configure the number of iterations. For LDA, we also need to specify the number of selected topics.

Table 2 shows the results of our parameter optimization experiments. Two of 10 parameters are found as not significant with respect to the tested variants, namely the vocabulary size and the type of *m*-grams. The observed optimal configuration is used for the final benchmarks in Table 3 showing highly competitive results on all datasets with un-weighted average F1 scores above 80%.

The application of the LSTM LM interpolated with a 5-g Backoff LM showed a promising score of 90.5% on the *Large Movie Review Dataset v1.0*, even though the LSTM LM is trained only on 40% of the available training data due to the large time and memory requirements. Nevertheless, the LM approach does not require any kind of feature selection or dimensionality reduction.

CONCLUSIONS

We discussed various techniques that are used in sentiment analysis. We optimized and presented the performances of our algorithms on various datasets. The first thing to notice is that the accuracy heavily depends on the type and size of the dataset. In general, lengthy reviews are easier to analyze in comparison to short messages. This is likely because when people

TABLE 2 | Experimental Results as Percentage Un-Weighted Average F1 Measures over the Four First Datasets and All Runs

[%]	#runs	mean	std dev	std err	LB	UB	min	max
Size of the vocabulary not significant (*p*-value = 0.646/0.750)								
5000	150	80.45	6.52	0.53	79.40	81.51	66.02	88.84
10,000	150	80.08	6.77	0.55	78.99	81.17	63.03	88.96
15,000	150	79.84	6.93	0.57	78.72	80.95	60.77	89.20
20,000	150	79.63	7.04	0.57	78.49	80.77	60.56	88.83
Selected *N*-grams not significant (*p*-value = 0.786/0.386)								
2	300	79.75	6.79	0.39	78.98	80.52	60.77	88.44
3	300	80.25	6.83	0.39	79.47	81.03	60.56	89.20
Feature selection method significant (*p*-value = 0.017/0.003)								
Mutual information	120	78.88	7.39	0.67	77.54	80.22	60.56	87.14
Chi-square select	480	**80.28**	6.63	0.30	79.68	80.87	64.25	89.20
Chi-square alpha value significant (*p*-value = 0.000/0.000)								
0.01	120	79.06	5.51	0.50	78.06	80.05	67.66	85.36
0.05	120	81.08	6.95	0.63	79.82	82.34	67.34	89.20
0.1	120	**81.12**	6.83	0.62	79.89	82.36	66.61	88.49
0.5	120	79.86	6.97	0.64	78.60	81.12	64.25	87.50
Classification model significant (*p*-value = 0.000/0.000)								
Binarized Naïve Bayes	40	85.61	1.23	0.19	85.22	86.00	83.37	87.09
Bernoulli Naïve Bayes	40	85.14	1.23	0.20	84.75	85.54	82.67	86.55
Multinomial Naïve Bayes	80	85.30	1.26	0.14	85.02	85.58	82.65	87.35
Maximum entropy	160	**86.36**	1.61	0.13	86.10	86.61	82.86	89.20
Ordinal regression	120	75.05	2.65	0.24	74.58	75.53	69.35	78.90
SoftMax regression	160	72.02	3.65	0.29	71.45	72.59	60.56	76.03
Number of training Iterations significant (*p*-value = 0.720/0.000)								
10	40	85.43	1.66	0.26	84.90	85.96	82.86	87.69
25	40	86.60	1.56	0.25	86.10	87.10	84.13	88.90
50	40	**86.72**	1.43	0.23	86.26	87.18	84.48	89.20
100	40	86.68	1.45	0.23	86.21	87.14	84.12	88.80
Weighting probabilities by TF significant (*p*-value = 0.814/0.024)								
Without weighting	40	**85.61**	1.22	0.19	85.22	86.00	83.47	87.35
With weighting	40	84.98	1.22	0.19	84.59	85.37	82.65	86.78
Dimensionality reduction significant (*p*-value = 0.509/0.000)								
None	280	**73.32**	3.59	0.21	72.90	73.74	61.00	79.00
LDA	64	62.12	3.32	0.42	61.29	62.95	54.00	66.00
DPMM	30	58.02	3.58	0.65	56.68	59.36	54.00	65.00
Feature selection method significant (*p*-value = 0.488/0.000)								
Mutual information	20	**62.50**	3.56	0.80	60.84	64.17	55.18	66.37
Chi-square select	56	61.25	3.66	0.49	60.27	62.24	53.81	66.12
TF-IDF	18	57.55	3.17	0.75	55.98	59.13	54.48	64.15
LDA topics significant (*p*-value = 0.093/0.005)								
50	30	60.95	3.07	0.56	59.81	62.10	54.50	64.40
100	30	62.78	3.27	0.60	61.56	64.00	54.48	66.37
200	4	**65.94**	0.15	0.07	65.71	66.17	65.77	66.12

std-dev, standard deviation; std-err, standard error; LB, UB, 95% confidence interval for the mean's lower and upper bound.
For the test of significance: First, Levene's test is used to check for the homogeneity of variances. Second, Kruskal–Wallis or ANOVA test is performed. A significant effect is assumed if the *p*-value < 0.05 by Kruskal–Wallis or ANOVA.

TABLE 3 | Experimental Results Per Dataset and Classifier

	AMAZ	PD2	N&BD	SPD1	AMP3	LMRD	TWIT
Vocabulary size	10 k	5 k	5 k	5 k	10 k	10 k	10 k
Mean F1 score [%]							
Binarized Naïve Bayes	89.98	94.72	79.70	81.24	89.44	89.09	79.26
Bernoulli Naïve Bayes	90.32	94.64	76.93	81.37	**89.62**	89.33	79.28
Multinomial Naïve Bayes	89.99	94.64	79.83	81.07	89.47	89.10	79.25
Maximum entropy	**91.91**	**96.99**	**80.81**	**81.72**	89.22	**91.55**	**80.83**
Ordinal regression	79.32	89.30	71.00	78.75	86.17	85.65	78.24
SoftMax regression	75.18	84.61	66.52	75.87	75.64	83.90	77.15
Backoff 5-g LM	—	—	—	—	—	90.00	—
+ LSTM-RNN LM	—	—	—	—	—	90.50	—

Best results are highlighted. The vocabulary size depends on the size of the corpus. Other parameters are in optimal configuration as found above. The use of LMs is limited to the LMRD dataset. The 5-g Backoff LM is smoothed with Modified Kneser-Ney smoothing. The LSTM LM is a trained with a single hidden layer of 300 cells using only 40% of the available training data and interpolated with the 5-g Backoff LM.

write reviews they tend to explicitly mention why they liked or disliked a particular product or service. Moreover, most review datasets are highly targeted to a particular domain, enabling to identify and learn easily the domain-specific jargon and adjectives.

Some particular domains might require special handling (preprocessing, tokenization, feature selection, etc.). All these can heavily affect the accuracy. In our preliminary research, we tried to use dimensionality reduction methods. The results showed that working on a reduced feature space negatively affected the accuracy. In general, it is advised to work on the original feature space and use feature selection methods to reduce the number of available features.

We performed detailed statistical tests and analyzed the factors that affected the overall accuracy. The tokenization algorithm should be configured to extract relevant number of unigrams and bigrams. The size of the vocabulary depends on the size of the dataset. It is also observed that the feature selection algorithms that make use of class information lead to better results and tuning their configurations can lead to significant improvements. The superiority of the MaxEnt classifier became apparent from early on. However, in most of the cases, the NB classifier led to close results. The regression models appear to deliver poor accuracy—likely to be the result of multicollinearity problems as the words/features are highly correlated.

Our novel approach of using a long short-term memory language model showed promising results, even though it is only trained on 40% of the available training data. The advantages of this approach are that it operates directly on the raw sequences of words rather than bags-of-words or *m*-grams, it does not need prior stages with careful optimizations such as feature selection or dimensionality reduction. However, the large time and memory requirements are crucial problems.

In the future, it is important to further investigate the LSTM-based approach as well as exploring other dimensionality reduction techniques. One alternative method is to use the neural network as a direct classifier rather than to build an LM. Another way to reduce the feature space is to perform statistical word stemming.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement no. 338164 (ERC StG iHEARu) and the European Union's Horizon 2020 Programme under grant agreement no. 645378 (RIA ARIA-VALUSPA).

FURTHER READINGS

Balahur-Dobrescu A, Taboada M, Schuller B. Computational methods for affect detection from natural language. *Comput Soc Sci*. Springer International Publishing; 2015.

Schuller B, Batliner A. *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. Wiley; 2014.

Cambria E, Schuller B, Xia Y, Havasi C. New avenues in opinion mining and sentiment analysis. *IEEE Intell Syst Mag* 2013, 28:15–21.

REFERENCES

1. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002, 79–86. Philadelphia, PA, USA: ACL.
2. 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, 271–278. Barcelona, Spain: ACL. doi: 10.3115/1218955.1218990.
3. Taboada M, Brooke J, Tofiloski M, Voll K, Stede M. Lexicon-based methods for sentiment analysis. *Comput Linguist* 2011, 37:267–307. doi:10.1162/COLI_a_00049.
4. Ding X, Liu B, Yu PS. A holistic lexicon-based approach to opinion mining. In: *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, 2008, 231–240. Palo Alto, CA, USA: ACM. doi: 10.1145/1341531.1341561.
5. Liu K-L., Li W-J, Guo M. Emoticon smoothed language models for twitter sentiment analysis. In: *Proceedings of the 26th Conference Association for the Advancement of Artificial Intelligence (AAAI) and 24th Innovative Applications of Artificial Intelligence (IAAI)*, Toronto, ON, Canada, 2012, 1678–1684.
6. Hu Y, Lu R, Chen Y, Duan J. Using a generative model for sentiment analysis. *Int. J. Comput. Linguist. Chin. Lang. Process.* 2007, 12:107–126.
7. Dong L, Wei F, Zhou M, Xu K. Adaptive multi-compositionality for recursive neural nodels with applications to sentiment analysis. In: *Proceedings of the 28th Conference of Association for the Advancement of Artificial Intelligence (AAAI)*, Québec, Canada, 2014, 1537–1543.
8. Dhande L, Patnaik G. Analyzing sentiment of movie review data using Naive Bayes neural classifier. *Int J Emerg Trends Technol Comput Sci* 2014, 3:313–320.
9. Santos C, Gatti M. Deep convolutional neural networks for sentiment analysis of short texts. In: *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, Dublin, Ireland, 2014, 23–29.
10. Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, WA, 2013, 1631–1642.
11. Barbieri F, Saggion H. Modelling irony in Twitter. In: *Proceedings of the Student Research Workshop at 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Gothenburg, Sweden, 2014, 56–64.
12. Liebrecht C, Kunneman F, van den Bosch A. The perfect solution for detecting sarcasm in tweets #not. In: *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics*, Atlanta, GA, 2013, 29–37.
13. Reyes A, Rosso P. On the difficulty of automatically detecting irony: Beyond a simple case of negation. *Knowl Inf Syst* 2014, 40:595–614.
14. Liu B. *Sentiment Analysis and Opinion Mining*. New York: Morgan & Claypool Publishers; 2012.
15. Manning C, Raghavan P, Schütze H. *An Introduction to Information Retrieval*. Cambridge: Cambridge University Press; 2009.
16. Griffiths T, Steyvers M. Finding scientific topics. *Proc Natl Acad Sci USA* 2004, 101:5228–5235.
17. Fan W, Bouguila N. Online learning of a Dirichlet process mixture of generalized Dirichlet distributions for simultaneous clustering and localized feature selection. In: *Proceedings of the 4th Asian Conference on Machine Learning (ACML)*, Singapore, 2012, 113–128.
18. Nigam K, Lafferty J, McCallum A. Using maximum entropy for text classification. In: *Proceedings of the IJCAI 99 Workshop on Machine Learning for Information Filtering*, Stockholm, Sweden, 1999, 61–67.

19. Genkin A, Lewis D, Madigan D. Large-scale Bayesian logistic regression for text classification. *Am Stat Assoc* 2007, 49:291–304. doi:10.1198/004017007000000245.

20. Long S, Freese J. *Regression Models for Categorical Dependent Variables using Stata*. Stata Press; 2014.

21. Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling. In: *Proceedings of Interspeech*, Portland, OR, 2012.

22. Weninger F, Bergmann J, Schuller B. Introducing CURRENNT—the Munich open-source CUDA RecurREnt Neural Network Toolkit. *J Mach Learn Res* 2015, 16:547–551.