International Journal of Neural Systems, Vol. 0, No. 0 (2005) 1–17

NEURAL NETWORKS WITH EMOTION ASSOCIATIONS, TOPIC MODELING AND SUPERVISED TERM WEIGHTING FOR SENTIMENT ANALYSIS

PETR HAJEK*
Science and Research Centre, Faculty of Economics and Administration, University of Pardubice, Studentska 84, 532 10 Pardubice, Czech Republic
E-mail: petr.hajek@upce.cz

ALIAKSANDR BARUSHKA
Science and Research Centre, Faculty of Economics and Administration, University of Pardubice, Studentska 84, 532 10 Pardubice, Czech Republic
E-mail: aliaksandr.barushka@upce.cz

MICHAL MUNK
[1]*Department of Computer Science, Constantine the Philosopher University in Nitra, 949 74 Nitra, Slovakia*
[2]*Science and Research Centre, Faculty of Economics and Administration, University of Pardubice, Studentska 84, 532 10 Pardubice, Czech Republic*
E-mail: mmunk@ukf.sk

Automated sentiment analysis is becoming increasingly recognized due to the growing importance of social media and e-commerce platform review websites. Deep neural networks outperform traditional lexicon-based and machine learning methods by effectively exploiting contextual word embeddings to generate dense document representation. However, this representation model is not fully adequate to capture topical semantics and the sentiment polarity of words. To overcome these problems, a novel sentiment analysis model is proposed that utilizes richer document representations of word-emotion associations and topic models, which is the main computational novelty of this study. The sentiment analysis model integrates word embeddings with lexicon-based sentiment and emotion indicators, including negations and emoticons, and to further improve its performance, a topic modeling component is utilized together with a bag-of-words model based on a supervised term weighting scheme. The effectiveness of the proposed model is evaluated using large datasets of Amazon product reviews and hotel reviews. Experimental results prove that the proposed document representation is valid for the sentiment analysis of product and hotel reviews, irrespective of their class imbalance. The results also show that the proposed model improves on existing machine learning methods.

Keywords: Sentiment analysis, word embedding, term weighting, topic model, deep neural network.

1. Introduction

Sentiment analysis is intended to reveal users' real opinions or attitudes toward different aspects of products and services.[1] For example, consumers tend to post their reviews on online shopping platforms, particularly when their experience was exceptionally good or bad. Product reviews also help businesses and other consumers understand consumers' concerns and make purchase decisions, respectively.

*corresponding author

The last two decades have witnessed considerable developments in automated sentiment analysis, which has become a widely studied text categorization task. Its aim is to label text documents as having a positive or negative orientation. Sentiment orientation has a major impact on the perceived helpfulness of online comments.[2] The steadily increasing number of online comments across major shopping platforms and social media has led to the necessity of developing automated sentiment analysis systems.[1] On one hand, various sentiment lexicons have been constructed to produce sentiment scores. On the other, numerous machine learning models have been proposed for the task, including ones with unsupervised,[3] semi-supervised[4] and supervised learning.[5]

Three levels of granularity have been considered in the sentiment analysis of online comments, namely the document, sentence and aspect levels. At the document level, it is assumed that sentiment is consistent within the online comment, which are categorized into positive or negative sentiment classes. In other words, this classification task assumes that the online comments concern a single entity. For the sentence-level sentiment analysis, the classification task only selects and considers opinion sentences. For the aspect-level sentiment analysis, it must first identify the comment's aspect (target), which in turn leads to two subtasks, namely aspect extraction and aspect sentiment classification.

Looking at the features used for sentiment analysis, the bag-of-words model represents a traditional document representation that calculates term frequencies for each word or phrase in the vocabulary.[6] However, this approach suffers from high-dimensional sparse document representation. Moreover, only a limited context can be taken into account when using n-grams rather than single words. To address these issues, scholars introduced word embeddings to generate low-dimensional dense word representations.[7–11] Word embeddings are also more effective than the bag-of-words approach in modelling word context and word meaning.

Deep neural networks (DNNs) have recently attracted particular interest and proved an effective text and image classification tool due to their capacity to learn complex feature representations.[12–18] To avoid the above-mentioned high-dimensional sparse word representations, DNNs utilize word embeddings to model local word context. This in turn leads to a lower-dimensional dense word representation. Alternatively, DNNs can be used to produce such word representation and by averaging all words in the document provide inputs to other machine learning–based classification models, such as support vector machines (SVMs).[39]

A major issue with traditional word embeddings is that they fail to consider the sentiment of the words in terms of both sentiment polarity and intensity. Moreover, different aspects of comments are often neglected. DNNs proved an emerging prospect for aspect extraction and the sentiment analysis of online comments due to their ability to capture both semantic and syntactic high-level features without requiring prior feature engineering.[39] The aspect-specific word embedding model proposed by Du et al.[20] remains the only study investigating word vectors with respect to topics extracted using latent Dirichlet allocation (LDA). However, this approach also suffers from some serious drawbacks. First, as with all the previously mentioned methods, the sentiment polarity / intensity of words is overlooked. Second, such models consider the topics regardless of the capacity of words to discriminate between positive and negative comment orientation, particularly when considering slow LDA inference.

This study aims to overcome the above problems by developing a DNN model with richer document representation, which integrates word-emotion associations, topic modeling component and supervised term weighting. This document representation builds upon recent work combining word embeddings with sentiment scores.[5] However, several major novelties are presented in our model. First, compared with previous work, multiple lexicon-based sentiment and emotion indicators are used that provide the words contained in word embeddings with a more thorough assessment of sentiment polarity (positive, negative, or neutral), sentiment intensity (strength of positive and negative sentiments) and emotions (mood states),[21–23] including emoticons and negating words. Furthermore, this novel document representation is combined with a topic modeling component performed using LDA. Finally, this is the first study to demonstrate the effect of supervised term weighting in a DNN model for sentiment analysis. Specifically, bag-of-words is selected based on a supervised term-weighting scheme, thus considering

terms' power to discriminate between positive and negative sentiment orientation. Supervised learning is preferred in this study because a large number of labeled training documents can be obtained from existing datasets for sentiment analysis. In summary, the contributions of our study are twofold:

- A novel DNN-based sentiment analysis model is proposed that, as far as we know, is the first to integrate word-emotion associations with a topic modeling component and computationally effective bag-of-words component.
- Two benchmark datasets of Amazon product reviews and hotel reviews are used to demonstrate the effectiveness of the proposed integrated document representation model in sentiment analysis, and report significant improvements of classification performance over state-of-the-art sentiment analysis methods.

This article is a significantly extended version of the conference paper,[24] which demonstrated the effectiveness of word-emotion association for sentiment analysis. Here, an improved sentiment analysis model is proposed that is equipped with the topic modeling component and supervised term weighting. This allows us to examine the effects of different document representations on sentiment classification performance. In addition, an in-depth comparative statistical analysis is performed against existing sentiment analysis methods on the Amazon product review and hotel review datasets.

The remainder of this paper is structured as follows. Section 2 reviews recent advances in the automated sentiment analysis of online comments. Section 3 introduces the details of the proposed sentiment analysis model. Section 4 presents the datasets used for model evaluation. Section 5 presents experimental results and compares the model performance with existing models. Section 6 concludes, highlighting further research directions.

2. Related Work

Over the last two decades, there has been a considerable amount of literature on the automated sentiment analysis of online comments. Notably, recent years have seen considerable interest in DNN–based approaches. This section reviews previous machine learning–based approaches to the sentiment analysis of online comments, as presented in the list of related studies in Table 1.

2.1. *Bag-of-words Models*

As shown in earlier studies, neural networks (NNs) outperform other traditional machine learning methods such as SVM and Naïve Bayes (NB) in this classification task, regardless of the context of balanced/unbalanced datasets.[25] The traditional approach uses the bag-of-words model to generate sparse and high-dimensional document representation.[26,27] However, shallow NNs have a limited ability to deal with sparse datasets.[28] By contrast, DNNs can capture more complex features from documents. Glorot et al.[29] proposed a DNN approach employing unsupervised learning to demonstrate that effective word representation is possible by learning a stacked denoising autoencoder. They also conclusively showed that such representation can be easily adapted to different product and service domains. To overcome the problem of the scalability of the traditional autoencoders with the high-dimensional bag-of-words model, Zhai and Zhang[30] proposed a semi-supervised autoencoder. Specifically, they introduced supervision into the model via the loss function obtained from a linear classifier. Initially, convolutional NNs (CNNs) also used the bag-of-words representation,[6] which was the first attempt to make use of word order for sentiment analysis.

2.2. *Word Embedding Models*

To further improve the performance of DNNs in sentiment analysis, other studies employed vector representation models, such as Word2Vec (including continuous bag of words (CBOW) and SkipGram models),[31,32] bidirectional encoder representations from transformers (BERT)[33] and GloVe.[34] The decisive advantage of these models is that they produce dense word / sentence/ document representations by reconstructing the linguistic context of the words. In other words, this approach takes advantage of words with a common context are located close in the vector space. Thus, the originally high dimensionality of the space can be reduced to several hundred features representing word embeddings. Tang et al.[7] employed long short-term memory (LSTM) and CNN to learn sentiment representation based on word embeddings and, consequently, gated recur-

rent units (GRUs) were used to learn the document representation. They decided that word embeddings combined in a CNN model provide the best sentiment classification performance, as compared with NB and SVM. Another CNN model combined word embeddings with user preferences extracted from the consumer reviews.[8] Similarly, Chen et al.[9] exploited product and user information in an LSTM classification model equipped with word and sentence attention. To address the issue of LSTM memory unit with long texts, Xu et al.[10] developed a cached LSTM model that captures the overall semantic representation. Vector representation models were also modified with respect to sentiment polarity to improve the performance of sentiment analysis models.[78] An intriguing area in the field is sentiment classification across domains, which Li et al.[36] addressed by an end-to-end adversarial memory network. To adaptively focus on aspect-related words, Tay et al.[37] developed an aspect fusion LSTM model that ameliorates the drawback of simple word-aspect similarities. Indeed, aspect-based sentiment analysis has become increasingly popular recently.[38–40] Lexicon- and corpus-based sentiment scores were assigned to aspects identified by pre-defined lexicons.[41]

2.3. *Combinations of Word Representation Models*

Another challenging task is combining different word representations. Context features, including word location, part-of-speech (POS) and sentiment score, can append embedding representations in the feature-based compositing memory networks, showing that ignoring words without sentiment is more effective than document representations without context features.[42] Zhang et al.[43] a cross-modality consistent regression model to take advantage of three different CNNs used to model semantic, sentiment and lexicon representations. Lexicon and sentiment representations reportedly address the disadvantages of semantic word embeddings in sentiment analysis.[43] However, the word embedding representations used in prior studies ignore the sentiment polarity / intensity of the words. Consequently, words with different sentiment polarity are combined in one feature, which may limit the classification performance of machine learning methods in sentiment analysis tasks. In other words, this may lead to the misrepresentation of documents in the context of sentiment analysis. Moreover, hybrid representation models combining word embeddings with different sentiment and semantic representations may further improve classification performance in related tasks due to highly domain-specific context.[44,45] Product and service reviews from different domains represent exactly such a task. Inspired by these observations, the original contribution of this study is the proposal of a DNN model integrating word embeddings with lexicon-based sentiment and emotion features. Notably, the proposed word-emotion associations enable us to obtain both the meaning and sentiment polarity / intensity / emotions of the words in the online comment representation. In agreement with earlier research,[46] the proposed model considers different topics by extracting latent features from the word representation. Finally, the used bag-of-words representation utilizes a supervised term-weighting scheme. The discriminative power of terms was also considered in previous studies,[46] but learning term weights during the neural network training process turned out to be prone to overfitting and highly time-consuming for high-dimensional data.[47] Therefore, the proposed model considers the discriminative power of terms as early as in the bag-of-words representation.

3. Neural Network Model

Fig. 1 depicts the architecture of the proposed DNN model with word-emotion associations, topic modeling and bag-of-words (BoW) component selected using supervised term weighting for the sentiment analysis of online comments. A DNN with convolutional, pooling and two dense hidden layers was used to capture high-level features from the hybrid document representation obtained from the word-emotion representation, topic modeling and BoW representation.

3.1. *Word-Emotion Representation*

The word-emotion representation is produced in two stages. In the first stage, the Skip-Gram model[31] is trained to obtain word embeddings. This model was used because it is reportedly more effective than its competitors in exploiting the word context.[31] Unlike the CBOW model, the target word is used as input while the context words represent the output layer in the Skip-Gram model. In the second stage, the

Table 1. Summary of previous studies on sentiment analysis of consumer reviews

Study	Features	Method	Dataset
Kang (2012)[48]	BoW, POS, SentiWordNet sentiment score	NB	70K restaurant reviews
Johnson (2015)[6]	binary BoW	CNN	25K electronic product reviews from Amazon
Zhang (2015)[49]	characters	Temporal CNN	4M Amazon product reviews
Du (2016)[20]	word-aspect CBOW	CNN	∼44M Amazon product reviews
Chen (2016)[9]	SkipGram, user/product specific words	Hierarchical LSTM	231K Yelp reviews
Poria (2016)[50]	CBOW, POS	CNN	7.7K reviews from the SemEval 2014 dataset
Zhai (2016)[30]	BoW	Semi-supervised Autoencoder	62K Amazon product reviews
Wang (2016)[51]	Word2Vec	Recursive neural conditional random fields	7.7K reviews from the SemEval 2014 dataset
Poria (2016)[52]	Word2Vec, POS	Temporal CNN	448 multimodal utterances
Gu (2017)[53]	SkipGram	Cascaded CNN	13K Amazon smartphone reviews, 18K Taobao skirt reviews
Chen (2017)[54]	Word2Vec	BiLSTM with CRF + CNN	4K sentences from Amazon
Li (2017)[36]	Word2Vec	Adversial memory network	78K Amazon product reviews
Mubarok (2017)[55]	BoW, POS	NB	3.6K reviews from the SemEval 2014 dataset
Catal (2017)[56]	BoW	Bagging + SVM + NB	9K Turkish e-commerce reviews
Peng (2018)[38]	SkipGram	LSTM	2.3K Chinese reviews
Tay (2018)[37]	GloVe	LSTM	17K reviews from the SemEval 2014 and 2015 datasets
Rathor (2018)[57]	weighted unigrams	SVM	24.5K Amazon product reviews
Asghar (2019)[41]	BoW, POS	Lexicon- and corpus-based SentiWordNet	84K sentences from electronic product reviews
Gamal (2019)[58]	n-grams with $tf.idf$ weights	PA, RR	1K Amazon product reviews
Huang (2019)[59]	Word2Vec	CNN + RNN	∼500K Amazon fine food reviews
Jagdale (2019)[60]	BoW, sentiment score	SVM, NB	12K Amazon product reviews
Kausar (2019)[5]	BoW, POS, SentiWordNet sentiment score	RF, DT, NB, SVM, Gradient Boosting, LSTM	31K Amazon product reviews
Ma (2019)[42]	Location, POS, NRC Hashtag sentiment score	FCMN	8K reviews from the SemEval 2014 dataset
Riaz (2019)[3]	Sentiment strength, keyword extraction, $tf.idf$ weights	k-means	1.2M product reviews from Amazon, eBay and Alibaba
Mandhula (2020)[46]	keyword extraction using LDA	CNN + LSTM	∼35M Amazon product reviews
Miao (2020)[4]	BERT	Semi-supervised learning	71K reviews from the SemEval 2014 dataset
This study	Word-emotion associations, topic modeling using LDA, supervised $tf.idf$-based BoW	CNN	400K Amazon product reviews, 515K hotel reviews

BERT – bidirectional encoder representations from transformers, BoW – bag-of-words, CNN – convolutional neural network, CBOW – continuous bag of words, CRF – conditional random field, DT – decision tree, FCMN – feature-based compositing memory network, LDA – latent Dirichlet allocation, LSTM – long short-term memory, NB – naïve Bayes, PA – passive aggressive, POS – part-of-speech tagging, RF – random forest, RR – ridge regression, RNN – recurrent neural network, and SVM – support vector machine.

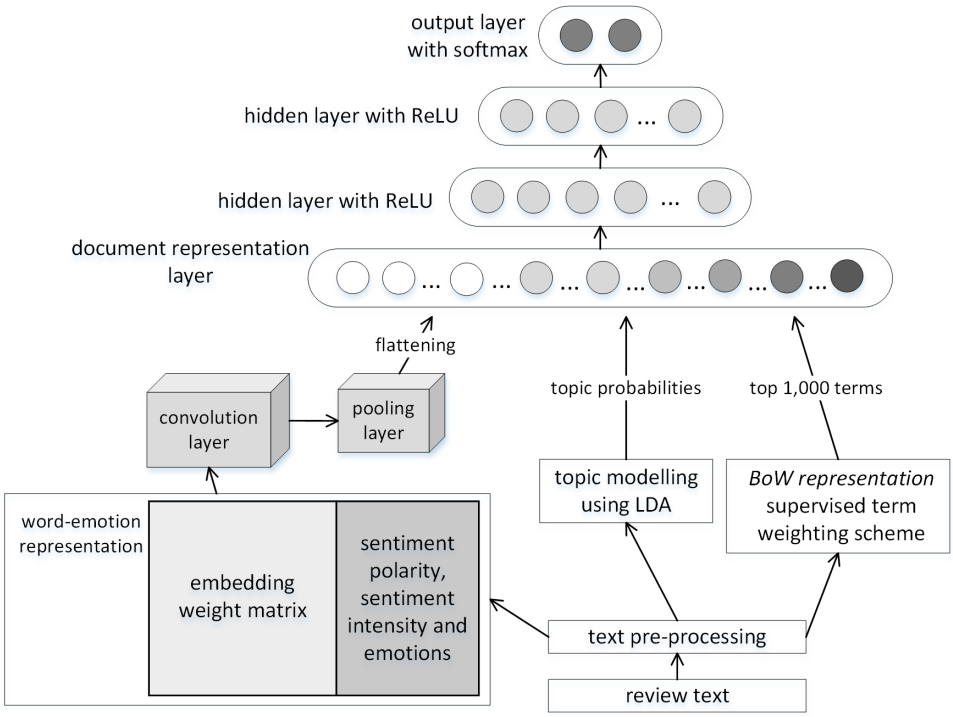

Figure 1. The proposed DNN-WEAE architecture for the sentiment analysis of online comments.

vocabulary generated from the corpus of consumer reviews is compared with sentiment-based lexicons to identify various sentiment polarity and sentiment intensity features.

To generate the embedding weight matrix, the embedding function is learnt and applied to each word w_t in the vocabulary. The embedding function is produced for the sequence of training words $W = \{w_1, w_2, \ldots, w_t, \ldots, w_T\}$ so that the following loss function is maximized:

$$E = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c} \log p(w_{t+j}|w_t) \ , \qquad (1)$$

where c denotes the context window radius (the number of surrounding words examined); and $p(w_{t+j}|w_t)$ is the probability of the output target word given the input context words, calculated using the hierarchical softmax algorithm as follows:

$$\begin{aligned} p(w_O|w_I) &= \prod_{j=1}^{L(w)-1} \sigma([n(w,j+1) \\ &= ch(n(w,j))]\nu'^{T}_{n(w_O,j)}\nu_{w_I} \ , \end{aligned} \qquad (2)$$

where w_I and w_O represent the input and output words, respectively; ν_w and ν'_w are the vector representations of the input and output words, respectively; $n(w,j)$ represents the j-th node in the binary tree; $L(w)$ denotes the path length in the tree; $ch(n)$ is a child node; and $\sigma(x)$ is a sigmoidal function (if x is true, then $[x] = 1$; otherwise $[x] = -1$). This can produce good embeddings by maximizing the loss function E, i.e., similar words have similar vectors. Words are represented by leaf nodes in the binary tree, and the tree structure substantially reduces the complexity by decomposing the probability calculations to at most $L(w)$ nodes. To generate the word tree, the Huffman-based approach was used.[31] The hyper-parameters in the model were set as follows: learning rate = 0.025, window size = 5 and word vector dimensionality = {100, 200, 400}.

To complement the word-emotion representation with the sentiment polarity and intensity of the words, several existing sentiment lexicons were used. To obtain a reliable lexicon-based emotion evaluation, it is best not to rely on a single lexicon.[61] In addition, the combination of various lexicon-based emotion indicators ensures wider lexical coverage and addresses the issue of susceptibility to indirect opinions, typically present in the machine learning-based models.[61]

To calculate sentiment polarity, two handcrafted lexicons of positive and negative words were used: Bing Liu's opinion lexicon[62] and OpinionFinder.[61] OpinionFinder is an annotated extended edition of the Multi-Perspective Question-Answering data. Bing Liu's opinion lexicon also includes slang and misspelled words, which results in this lexicon being more unique than the OpinionFinder lexicon.[63] One disadvantage of these lexicons is that equal weights are assigned to all words irrespective of their sentiment intensity. To overcome this problem, the sentiment intensity indicators from several pretrained lexicons[61,64] were incorporated: (1) SentiWordNet, (2) Sentiment140, (3) NRC Hashtag, and (4) AFINN. SentiWordNet extends the well-known WordNet database by annotating each synset with scores of positivity, neutrality and negativity in the range [0,1]. This annotation was performed automatically using a semi-supervised algorithm. Sentiment140 and NRC Hashtag are lexicons generated automatically from words with emotional tags. More precisely, the sentiment scores of Sentiment140 use positive and negative emoticons, while the NRC Hashtag lexicon uses positive and negative hashtags. Sentiment scores are obtained ranging from -5 to 5 for NRC Hashtag based on the point-wise mutual information between each word and the polarity of the corresponding message. The NRC emotion-based lexicon was considered that covers eight emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, and trust) adopted from the Plutchik wheel.[65] Again, these word lists are the result of human-based tagging. Finally, the list of emoticons was taken from the AFINN lexicon.[66] The list of the lexicon-based features we used is presented in Table 2, showing their source and description. Detailed description of the calculation of the sentiment polarity and intensity features is given in the original papers.[61,64]

3.2. *Latent Dirichlet Allocation*

LDA represents an enhanced generative topic model, in which documents are multinomial distributions of latent topics (mixtures of words).[67] Topics, on the other hand, are represented by word distributions. Each document is generated by a two-step probabilistic process. First, the word probability distribution Φ_k is sampled for the k-th topic in the Dirichlet distribution $Dir(\beta)$ with the topic parameter β. Second, the topic probability distribution θ_j is sampled for the j-th document in the Dirichlet distribution $Dir(\alpha)$, where the latent variable z_n follows a multinomial distribution θ. Given the parameters α and β, which determine the Dirichlet priors on θ and Φ, the joint probability distribution is as follows:

$$p(\theta, z, d|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \ , \quad (3)$$

where word w_n is generated from the multinomial distribution and the model parameters $p(\theta|\alpha), p(z_n|\theta)$ and $p(w_n|z_n, \beta)$ can be estimated by optimization algorithms. In this study, the collapsed variational Bayes approximation (with iteration limit = 100, data pass limit = 1, mini-batch size = 1,000, and learning rate decay = 0.5) was used due to its faster convergence rate compared with collapsed Gibbs sampling.[68] In agreement with previous studies,[69] only verbs and nouns were used for topic modeling. The use of these features is justified because most aspect terms are nouns or noun chunks.[50] The Stanford Tagger was employed for POS tagging. In addition to topic probabilities identified using LDA, we followed Poria at al.[50] and used six POS tags (noun, verb, adverb, conjunction, adjective, and preposition), calculated as the absolute frequencies of the terms selected using the supervised term weighting scheme.

3.3. *Supervised Term Weighting Scheme for Sentiment Analysis*

Let D^1 and D^2 be the sets of documents of positive and negative opinion classes, respectively. The j-th document d_j is represented by a vector of term weights $d_j = (w_{j1}, w_{j2}, \ldots, w_{jm})$, defined for terms $f_1, f_2, \ldots, f_m$. In the used supervised weighting scheme, w_{ij} is calculated as follows:

$$w_{ij} = ITD(f_i, d_j) \times ITS(f_i) \ , \quad (4)$$

where $ITD(f_i, d_j)$ denotes the term frequency of term f_i in document d_j and $ITS(f_i)$ is the capacity of f_i to evaluate sentiment. To consider different lengths of documents, the raw term frequency was normalized to calculate $ITD(f_i, d_j)$. As presented in earlier research, the traditional document frequency df_i indicates the importance of term f_i in all documents, thus neglecting differences between positive and negative opinion classes. To overcome this shortcoming, $ITS(f_i)$ is used based on a weighted frequency and odds (WFO) feature selection method[70]

Table 2. List of lexicon-based features.

Category	Lexicon	Feature	Range
Polarity	Bing Liu	BLPos (# positive words)	$\{0, 1, \ldots, n\}$
		BLNeg (# negative words)	$\{0, 1, \ldots, n\}$
	OpionFinder	OFPos (# positive words)	$\{0, 1, \ldots, n\}$
		OFNeg (# negative words)	$\{0, 1, \ldots, n\}$
Intensity	SentiWordNet	SWNPos (sum of the scores for positive words)	$[0, \infty]$
		SWNNeg (sum of the scores for negative words)	$[-\infty, 0]$
	Sentiment140	S140Pos (sum of the scores for positive words)	$[0, \infty]$
		S140Pos (sum of the scores for negative words)	$[-\infty, 0]$
	NRC Hashtag	NRCPos (sum of the scores for positive words)	$[0, \infty]$
		NRCNeg (sum of the scores for negative words)	$[-\infty, 0]$
	AFINN	APos (sum of the scores for positive words)	$\{0, 1, \ldots, n\}$
		ANeg (sum of the scores for negative words)	$\{-n, \ldots, 0\}$
Emotion	NRC Emotion	ANG (# words matching the anger word list)	$\{0, 1, \ldots, n\}$
		ANT (# words matching the anticipation word list)	$\{0, 1, \ldots, n\}$
		DIS (# words matching the disgust word list)	$\{0, 1, \ldots, n\}$
		FEA (# words matching the fear word list)	$\{0, 1, \ldots, n\}$
		JOY (# words matching the joy word list)	$\{0, 1, \ldots, n\}$
		SAD (# words matching the sadness word list)	$\{0, 1, \ldots, n\}$
		SUR (# words matching the surprise word list)	$\{0, 1, \ldots, n\}$
		TRU (# words matching the trust word list)	$\{0, 1, \ldots, n\}$
	Emoticons	EmoticonPos (# positive emoticons)	$\{0, 1, \ldots, n\}$
		EmoticonNeg (# negative emoticons)	$\{0, 1, \ldots, n\}$

particularly effective for sentiment classification. The probabilities of frequency and odds (WFO) can be estimated as follows:

$$WFO(f_i, D^k) \approx \left(\frac{x_i^k}{N_k}\right)^{\lambda} \log\left(\frac{x_i^k (N_1 + N_2 + N_k)}{y_i^k N_k}\right)^{1-\lambda}, \tag{5}$$

where x_i^k is the number of documents from D^k that contain the term f_i, y_i^k is the number of documents that do not belong to D^k that contain the term f_i, λ is the ratio between frequency and odds, and N_k is the number of documents in D^k. Following extensive experiments performed by Deng et al.[47] on multiple sentiment analysis datasets, the value of the hyper-parameter λ was set to 0.1 in this study to ensure stability between frequency and odds. To obtain $ITS(f_i)$, the maximum of WFO for the positive and negative sentiment classes was calculated. For further processing, the terms f_i were ranked according to their weights w_{ij}, and selected the top n=1,000 terms[71] to enter the document representation layer.

3.4. *Training the Neural Network Model*

The DNN model comprises one convolution layer with 50 feature maps (filters) with filter size 3, followed by a max-pooling layer of size 2. The maximum numbers of words in the reviews were used to fix the size of the inputs. The next two hidden layers in the DNN architecture are used to process the complex relationship between the document representation and the outputted positive / negative sentiment class of the online comment. To avoid overfitting and make the training more effective, dropout regularization was applied with dropout rates of 0.2 and 0.5 for the input and hidden layers, respectively. Rectified linear units (ReLU) were used to represent the convolutional and dense hidden layers. Training the DNN using the mini-batch gradient descent algorithm with $b = 100$ mini-batches, a learning rate of 0.1 and $I = 1{,}000$ iterations provided us with stable convergence and computationally efficient behavior. Cross-entropy loss was used as the objective function. Different numbers of filters were tested in the convolutional layer = $\{20, 50, 100\}$, and n_{h1} and n_{h2} of ReLU in the two dense layers = $\{2^4, 2^5, \ldots, 2^9\}$ to obtain the optimal DNN architecture. Experiments for two convolutional and one / three dense layers were performed, but without improvement. The results for these architectures are not presented in this study due to space limitations.

The computational complexity of the proposed

DNN model can be expressed as $O(b \times I \times (k \times n \times d^2 + m \times n_{h1} + n_{h1} \times n_{h2} + n_{h2} \times n_O))$, where k is the length of the filter in the convolutional layer, n is the sequence length in the convolutional layer, d is word vector dimensionality, m is the number of features in the document representation layer and n_{h1}, n_{h2} and n_O denote the numbers of neurons in the dense and output layers, respectively. This implies that the number of iterations, word vector dimensionality and number of terms in the bag-of-words model are the most the most important determinants of the computational complexity of the proposed model.

4. Data and Preprocessing

For the experiments, two large datasets were used, namely the Amazon and Hotel review datasets openly accessible on Kaggle[ab]. The Amazon dataset was provided by Xiang Zhang and originally used to classify the sentiment of consumer reviews using temporal CNNs with character-level features.[49] The dataset has been gradually expanded within the Stanford Network Analysis Project since 1994,[72] currently resulting in ∼34 million reviews from ∼6.6 million users on ∼2.4 million products. The mean character length of the consumer reviews in the dataset was 764 (90.9 words). Extremely short and long consumer reviews were discarded, and duplicates were removed by Xiang Zhang. Users' rating scores serve to categorize the consumer reviews into positive and negative sentiment orientation. More precisely, scores 1 and 2 indicate negative sentiment, whereas 4 and 5 scores indicate positive sentiment. To evaluate the effectiveness of the proposed DNN model, the testing data from the original dataset was used in this study, represented by 400,000 consumer reviews evenly distributed into positive and negative sentiment classes. The text of the reviews was represented by review title and review content. Regarding the Hotel review dataset, of 515,738 customer reviews in total, 485,035 were negative (with overall ratings $<$ 5) and 30,703 were negative (with overall ratings $\geq$ 5). In other words, the Hotel review dataset was imbalanced 15.8 to 1 in favour of negative reviews. It is worth noting that experiments were performed with random undersampling and oversampling to address the data imbalance problem but without improvement in accuracy. The mean number of words for this dataset was 35.6. In the text pre-processing stage, we carried out tokenization (using the following delimiters: ".,;:'"()?!") and transformation to lowercase letters. A prefix was also added to words occurring in negated contexts in case of the bag-of-words model.

5. Experimental Results

First, the effectiveness of each component of the proposed document representation model was investigated. Two evaluation measures were considered, namely accuracy (Acc = (true positives + true negatives) / (true positives + true negatives + false positives + false negatives)), and area under receiver operating characteristic curve (AUC). To evaluate classification performance, the datasets were divided into training and testing sets containing 80% and 20% of data instances, respectively. This data split proved to be effective for deep learning methods in sentiment analysis.[73] Stratified split was applied to maintain the sentiment class prevalence between data splits. To ensure reliable and consistent results, this procedure was repeated ten times; the mean values and standard deviations are presented for the testing set.

To obtain word embeddings, the Skip-Gram model was trained on the original Amazon product review dataset with ∼34 million reviews for the Amazon dataset, while the Hotel review dataset was used to produce word embeddings for the hospitality domain. Fig. 2 illustrates that different settings of the Skip-Gram model were examined. The best performance was achieved with 200 and 400 word embeddings for the Amazon dataset and Hotel dataset, respectively. We trained the Skip-Gram model in the Deeplearning4j environment (distributed, open-source DNN library written for Java, compatible with Clojure and Scala, and integrated with distributed computing frameworks Hadoop and Apache Spark).

[a]https://www.kaggle.com/bittlingmayer/amazonreviews
[b]https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe

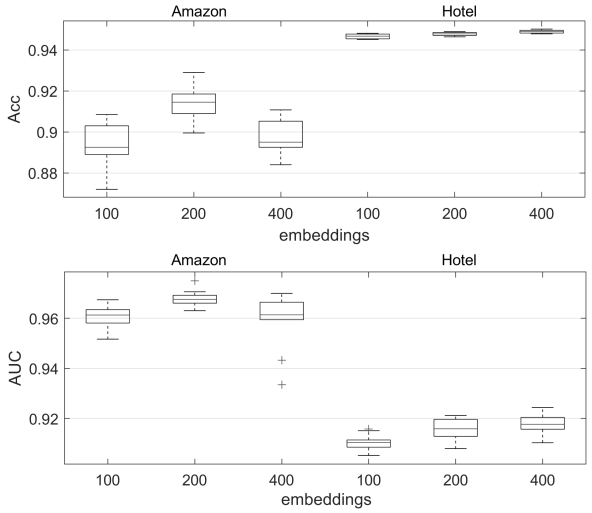

Figure 2. The effect of the number of word embeddings on the classification performance of the DNN model with 50 filters, $n_{h1} = 2^5$ and $n_{h2} = 2^4$ neurons in the hidden layers.

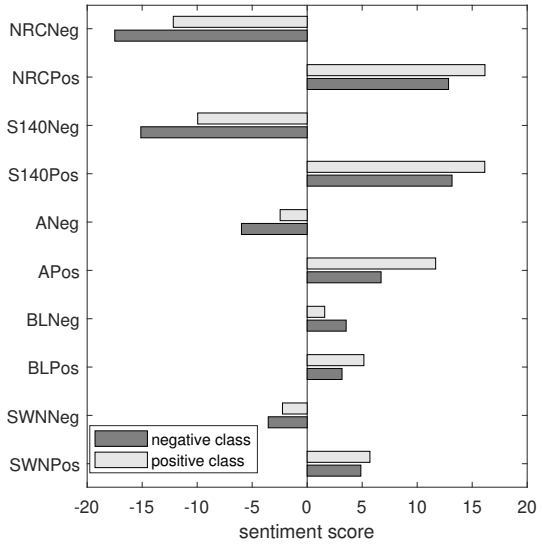

Figure 3. Average sentiment polarity and sentiment intensity of Amazon product reviews.

As shown in Fig. 3, the positive (negative) sentiment polarity / intensity indicators of the consumer reviews in the positive sentiment class have higher (lower) mean values than those in the negative class. Fig. 4 illustrates that reviews in the positive class are characterized by higher values of positive engagement (joy, anticipation, trust and surprise), whereas the negative class is distinguished by emotions with negative engagement, such as sadness, fear, disgust and anger. The mean values of the emoticon positive and negative scores for the positive class were 0.027 and -0.013, respectively. In contrast, it was only 0.006 and -0.021 for the negative class. Overall, these results indicate the valuable role of sentiment- and emotion-based indicators in the sentiment analysis of consumer reviews. Similar results were observed for Hotel reviews. To calculate the values of the sentiment polarity / intensity and emotion-based indicators, we used the AffectiveTweets package.[74]

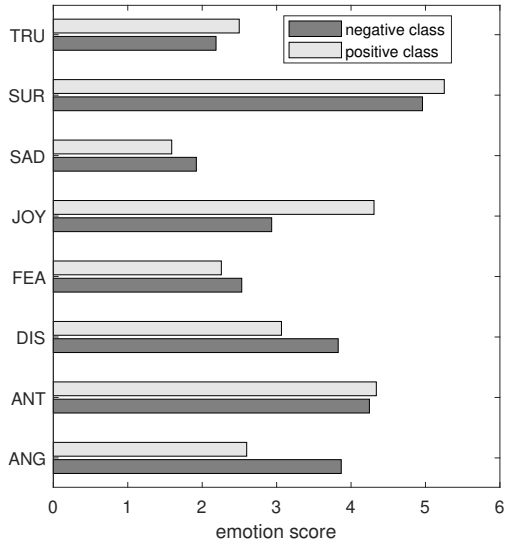

Figure 4. Mean values of emotion-based indicators for Amazon product reviews.

The LDA model was trained using the collapsed variational Bayes approximation, implemented in the Text Analytics Toolbox using Matlab 2019b. The maximum number of iterations was set to 1,000. To select the appropriate number of topics in LDA, different numbers of topics were examined in the range $\{5, 10, \ldots, 60\}$. A tuning procedure was employed to minimize the LDA model's perplexity on 10% of held-out data. Fig. 5 shows the minimum validation perplexity was achieved for thirty and five topics, respectively; therefore, the number of topics was K=30 for the Amazon dataset and K=5 for the Hotel dataset. For the latter, the generated word clouds indicated that the five topics represented hotel food, staff, location, hotel services, and room quality.

Regarding the discriminative power of terms, terms with strong sentiment engagement were selected, ranked for the Amazon dataset as follows: "great," "waste," "money," "love," "worst," "poor,"

"excellent," "bad," "disappointed," etc. This suggests that such a weighting scheme is appropriate for the sentiment analysis of consumer reviews. Fig. 6 illustrates the effectiveness of the supervised term weighting scheme for the bag-of-words (BoW) representation. Traditional $tf.idf$ (term frequency – inverse document frequency) weights for the top 1,000 n-grams (unigrams, bigrams and trigrams) were used for comparison.[75]

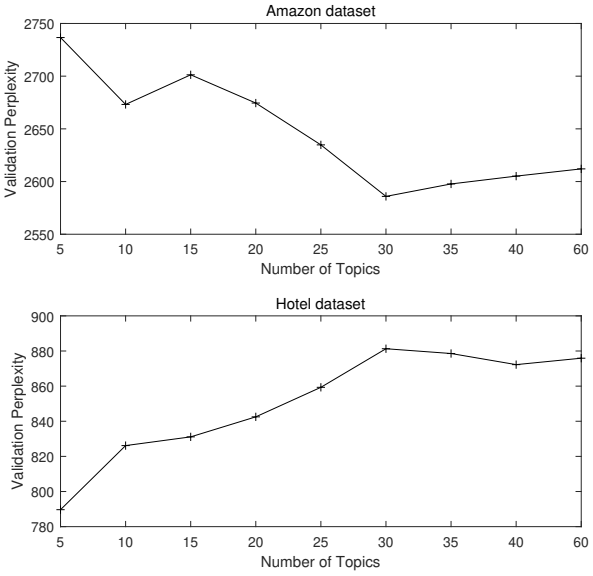

Figure 5. The effect of the number of topics in LDA on validation perplexity.

The above results indicate the separate effectiveness of the three document representation components. In a further set of experiments, the synergic effects of combining these components into an integrated model were investigated.

The quality of the proposed models were evaluated using the Acc and AUC evaluation measures. Since the examined variables had a normal distribution (Kolmogorov-Smirnov test: $N = 10$, max $D < 0.324$ (0.381), $p > 0.05$), parametric tests for repeated measures were used. The Mauchley sphericity test was used to verify the sphericity assumption for repeated measures with five levels ((1) DNN-TM: DNN with the topic modeling component; (2) DNN-WE: DNN with word-emotion representation; (3) DNN-BoW: BoW with adjusted (supervised) term weights; (4) $\text{DNN}_{BoW+TM+WE}$: DNN with all document representations; and (5) $\text{DNN}_{unadj.BoW+TM+WE}$: $\text{DNN}_{BoW+TM+WE}$ with unadjusted BoW term weights). For both datasets, the test was significant (Acc: $p = 0.0127$; AUC: $p = 0.1059$ for the Amazon dataset, and Acc: $p = 0.3067$; AUC: $p = 0.00005$ for the Hotel dataset). The assumption was violated, indicating that the type I error increases. The degrees of freedom were adjusted using Greenhouse-Geisser and Huynh-Feldt adjustments for the F-test to achieve the declared level of significance. The results showed that the null hypotheses, that there is no statistically significant difference in the values of the evaluation measures between the investigated models, were rejected at the 0.001 significance level.

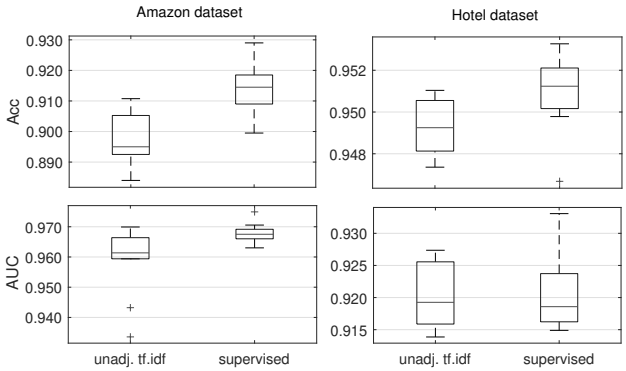

Figure 6. The effect of term weighting on the classification performance of the DNN model with 50 filters, $n_{h1} = 2^5$ and $n_{h2} = 2^4$ neurons in the hidden layers.

After rejecting the global null hypotheses, statistically significant differences in performance were tested between models. For multiple comparisons, the Newman-Keuls test was used, which has more power than common post-hoc tests. From multiple comparisons based on Acc for the Hotel dataset, only one homogeneous group was identified: DNN-WE and $\text{DNN}_{unadj.BoW+TM+WE}$ performed the same ($p > 0.05$). Statistically significant differences in performance between all investigated models were identified for both evaluation measures in other cases ($p < 0.05$). $\text{DNN}_{BoW+TM+WE}$ models with unadjusted as well as adjusted $tf.idf$ achieved high quality.

Fig. 7 and Fig. 8 show that the DNN model using the topic modeling component had the worst performance. More precisely, the DNNs with word-emotion representation and supervised term weights increased accuracy by 17.4% and 1.1%, respectively,

compared with DNN-TM. DNN-WE and DNN-BoW performed similarly in terms of both the evaluation measures. The $\text{DNN}_{BoW+TM+WE}$ model performed best with a 5.1% and 0.5% increase in accuracy compared with the DNN-BoW model for the Amazon dataset and Hotel review dataset, respectively. Overall, strong evidence of the effectiveness of the combination of the three components was found. Further statistical tests revealed that $\text{DNN}_{BoW+TM+WE}$ performed significantly better than the baseline models at $p < 0.01$. Other statistical tests (Friedman ANOVA and multiple comparisons based upon the mean rank differences) also revealed that $\text{DNN}_{BoW+TM+WE}$ performed significantly better than the baseline models at $p < 0.01$. The results of the parametric and nonparametric approaches agree and can be considered robust.

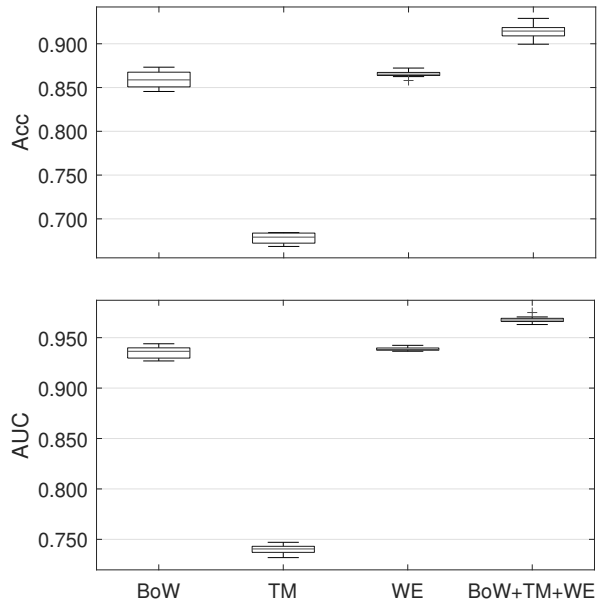

Figure 7. The performance of the DNN model for the Amazon dataset using a) word-emotion representation (WE), b) topic modeling (TM), c) bag-of-words with supervised term weights (BoW), d) all document representation components (BoW+TM+WE). The models were trained using 50 filters, $n_{h1} = 2^5$ and $n_{h2} = 2^4$ neurons.

To comprehensively evaluate the effectiveness of the proposed DNN models, their performance was compared against the following existing models, used in earlier studies on the sentiment analysis of consumer reviews:

- Improved NB (INB-1)[48] accommodates word sentiment using the SentiWordNet lexicon in the feature extraction process. Like Kang,[48] unigrams, bigrams and sentiment patterns were extracted.
- LSTM[7] and CNN[7] were used to capture semantic sentence-level representation. In agreement with Chen et al.,[9] the dimension of hidden / cell states in LSTM was set to 200, corresponding to the number of word embeddings. The CNN model comprised the convolutional layer with five filters of size 5 and a max pooling layer of size 4. For both models, the sentence representation was fixed using the number of words in the longest review. The document representation for both models was generated as a composition of sentence representations using GRUs. Stochastic gradient descent with an Adam optimizer was the learning algorithm used to train both models in the Deeplearning4j environment.
- Aspect-specific sentiment word embedding (ASSWE)[20] is based on the CBOW model generated for each word-aspect pair. LDA was trained with the collapsed Gibbs sampling algorithm to assign aspects to each term. The remaining training parameters of the CNN model were the same as in the previous comparative model.
- The CNN+LP (linguistic pattern) model[50] is also based on the pretrained CBOW model. In addition, six basic POS tags were used as input features. Again, the CNN+LP model was trained using the Deeplearning4j environment.
- The ensemble classifier model NB + SVM + Bagging combines three baseline classifiers, namely NB, SVM and bagging.[56] Following the original study, unigrams were used as input features and voting was employed as the meta-classifier to obtain the final review classification.
- The aspect-based NB (ANB) model[55] uses three types of input features, namely POS tags (obtained using the Stanford CoreNLP library) and two bags-of-words containing, respectively, aspect- and sentiment polarity-related words. The Chi-square feature selection algorithm was used to reduce the dimensionality of the word representation, and the NB classifier was employed to classify the product reviews into the sentiment categories.
- The ridge regression (RR) classifier uses the top 1,000 n-grams according to their $tf.idf$ weights.[58] The RR model was selected because it performed

best in the original study for the Amazon product dataset, as compared with different machine learning algorithms, such as SVM, NB, AdaBoost and logistic regression.

- The NB classifier uses sentiment polarity scores at sentence level (NB-SPS).[60] The SentiWordNet lexicon was used to calculate the positive and negative polarity of each sentence.
- An SVM with word sense disambiguation (SVM-WSD)[5] utilizes adverbs scored using the SentiWordNet lexicon as input features. Adverbs were assigned positive and negative SentiWordNet scores, and the SVM was trained using the LibLINEAR library. The L2-regularized L2-loss SVM model was trained with the cost parameter C=1.

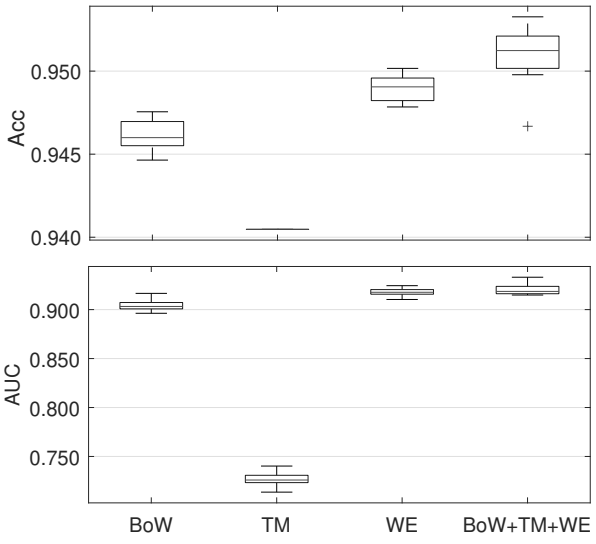

Figure 8. The performance of the DNN model for the Hotel dataset using a) word-emotion representation (WE), b) topic modeling (TM), c) bag-of-words with supervised term weights (BoW), d) all document representation components (BoW+TM+WE). The models were trained using 50 filters, $n_{h1} = 2^5$ and $n_{h2} = 2^4$ neurons.

Table 3 shows the results of $\text{DNN}_{BoW+TM+WE}$ in comparison with the above sentiment analysis models. Remarkably, the proposed model not only performs best in terms of all the evaluation measures used but also performs significantly better at $p < 0.01$ using nonparametric approaches (Friedman ANOVA and multiple comparisons based upon the mean rank differences), which emphasizes the validity of the proposed model and the robustness of the achieved results. SVM-WSD also performed well in terms of accuracy, especially when considering its low computational time.

Following previous studies,[45] the testing time criterion (measured as wall-clock time per 1,000 reviews) was adopted to demonstrate the real-time capacity of consumer review classifiers. The proposed $\text{DNN}_{BoW+TM+WE}$ model performed the worst in terms of time efficiency, but it can still be considered time efficient, with approximately 2,300 consumer reviews classified per second. The average training time of the proposed DNN model was approximately 1,650 s and 2,000 s for the Amazon product review dataset and Hotel review dataset, respectively. Recall that the crucial determinant of the computational complexity is the word vector dimensionality, leading to higher complexity for the Hotel dataset. Moreover, better time efficiency can be expected with a decrease in the number of n-grams. Overall, the model performed well for both sentiment classes, as indicated by the high value of AUC.

To verify the effectiveness of the proposed models, adjusted tests for repeated measures were used. Epsilon represents the degree to which sphericity has been violated. When comparing the proposed models to the existing ones, the Epsilon values were considerably less than one. The null hypotheses was thus rejected, claiming that there is no statistically significant difference in the values of Acc and AUC among the investigated models at $p < 0.001$.

In terms of multiple comparisons, one-sided tests were used to examine the effectiveness of the individual proposed models against existing models, i.e., many-to-one comparisons (existing models to proposed DNN model). The Dunnett test was used, which tests a null hypothesis — there is no statistically significant difference in efficiency (model performance) between the proposed model and existing models.

For the DNN-WE model, the null hypotheses was rejected for the existing ANB,[55] INB-1,[48] NB-SPS[60] and SVM-WSD[5] models, based on both evaluation measures at $p < 0.001$, i.e., the DNN-WE model was more efficient than the existing ANB[55] and INB-1[48] models for both datasets. This supports the dominance of word embedding models over bag-of-words models reported in earlier studies.[26,27] Similarly, the DNN-BoW was more efficient than the existing ANB,[55] INB-1,[48] NB-SPS,[60] SVM-WSD[5] and NB+SVM+Bagging[56] models, based on both evalu-

Table 3. Results of the experiments

	Amazon data			Hotel data		
Model	Acc	AUC	Testing time [s]	Acc	AUC	Testing time [s]
INB–1[48]	0.769±0.009	0.763±0.009	0.0348±0.0055	0.765±0.075	0.791±0.006	0.5223±0.0113
LSTM[7]	0.866±0.007	0.932±0.005	0.0245±0.0020	0.948±0.009	**0.920±0.004**	0.2435±0.0148
CNN[7]	0.859±0.004	0.932±0.002	0.0270±0.0001	0.941±0.008	0.907±0.013	0.3463±0.0036
ASSWE[20]	0.860±0.005	0.932±0.003	0.0330±0.0001	0.944±0.016	0.915±0.004	0.2571±0.0038
CNN+LP[50]	0.867±0.004	0.937±0.002	0.0529±0.0001	0.945±0.029	0.917±0.004	0.3440±0.0019
NB+SVM+Bagging[56]	0.848±0.007	0.922±0.004	**0.0003±0.0004**	0.933±0.024	0.842±0.003	**0.0028±0.0006**
ANB[55]	0.742±0.008	0.788±0.008	0.0331±0.0009	0.787±0.058	0.785±0.008	0.5845±0.0059
RR[58]	0.869±0.002	0.939±0.002	0.0007±0.0004	0.945±0.007	0.568±0.004	0.4730±0.0075
NB-SPS[60]	0.857±0.011	0.857±0.011	0.0008±0.0001	0.795±0.076	0.795±0.006	0.1609±0.0069
SVM-WSD[5]	0.861±0.011	0.861±0.012	0.0006±0.0000	0.931±0.177	0.675±0.081	0.0048±0.0005
$\text{DNN}_{BoW+TM+WE}$	**0.914±0.008**	**0.968±0.003**	0.2337±0.0057	**0.951±0.017**	**0.920±0.005**	0.8209±0.0023

Notes: The best results are in bold. The experiments were conducted on a server computer using an AMD Opteron 6180SE 2.50 GHz with twelve cores/threads and 256 GB RAM on a Windows 10 oper. system in the Deeplearning4j environment.

ation measures ($p < 0.001$). This can be attributed to the more effective feature selection in the bag-of-words model. Note that this improvement was observed mainly for the Amazon dataset with sufficient number of instances in both sentiment classes.

Based on the evaluation measures, the $\text{DNN}_{BoW+TM+WE}$ models with the unadjusted as well as adjusted $tf.idf$ had the highest quality compared to existing models. The null hypotheses was rejected for all existing models at $p < 0.001$ for the Amazon dataset. For the Hotel dataset, the $\text{DNN}_{BoW+TM+WE}$ models significantly outperformed most of the existing models except CNN,[7] ASSWE,[20] CNN+LP[50] and LSTM.[7] This can be explained by more effective learning of word embeddings in case of generally shorter hotel reviews.

6. Conclusion

This study proposes an efficient DNN model integrating word-emotion associations, topic modeling and supervised term weighting for the sentiment analysis of online comments. The DNN model is proved to perform better than baseline document representations for the Amazon product review and hotel review datasets, irrespective of the difference in their class imbalance ratio. The average value of sentiment classification accuracy of the proposed model was 91.0% and 95.1%. The improvement over the baseline document representations was achieved through the integrated representation. Compared with the baseline representations, the proposed model allowed us to increase Acc by on average 4.3% and 0.3%, respectively.

The proposed DNN-WEAE model was compared with ten state-of-the-art sentiment analysis methods combining sentiment analysis and topic modeling in different ways. In contrast to those approaches, this study considered various sentiment polarity / intensity and emotion indicators in word-emotion representation. In addition, the proposed model utilized a supervised term weighting scheme to improve BoW selection. The combination of these components performed best, indicating that the combination of a low-dimensional dense representation of word embeddings and high-dimensional sparse representation of BoW with high discriminative power caused the improved performance. However, such a document representation model leads to a partly sparse dataset, which necessitates further requirements for the sentiment classification methods. It was demonstrated that the proposed DNN model can handle such a document representation. The average AUC performance of existing CNN and LSTM architectures was improved using the proposed DNN model by 3% for the Amazon dataset, while no improvement was obtained for the Hotel dataset. This can be attributed to the reduced effect of supervised term weighting scheme in presence of limited number of reviews in one of sentiment classes.

Future research should investigate the word-emotion associations directly at the entity / aspect level, rather than separately. A limitation of the

proposed model is that it captures only local features. Therefore, future studies should investigate alternative DNN models with attention mechanisms. More research is also needed to better understand the cross-domain modifications of the model. To improve the understanding of sentences, recently developed pattern-based methods can be used.[77] Alternative embedding-based schemes, such as GloVe, fastText, Sentence-BERT, Universal Sentence Encoder and Word Mover's Embedding, can serve to generate word-emotion association. The proposed model should also be used in multi-class sentiment analysis, and new powerful supervised machine learning methods should be employed to automate the design of neural network models, such as neural dynamic classification[78] and dynamic ensembles of neural networks.[79] Finally, the time efficiency of the model can be improved using specialized TPU accelerators.

Acknowledgments

This article was supported by the scientific research project of the Czech Sciences Foundation Grant No. 19-15498S.

Bibliography

1. B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions* (The Cambridge University Press, Cambridge, 2015).
2. M. S. I. Malik and A. Hussain, Helpfulness of product reviews as a function of discrete positive and negative emotions, *Computers in Human Behavior* **73** (2017) 290–302.
3. S. Riaz, M. Fatima, M. Kamran and M. W. Nisar, Opinion mining on large scale data using sentiment analysis and k-means clustering, *Cluster Computing* **22**(3) (2019) 7149–7164.
4. Z. Miao, Y. Li, X. Wang and W. C. Tan, Snippext: Semi-supervised opinion mining with augmented data, in *Proc. of The Web Conference 2020* (2020), pp. 617–628.
5. S. Kausar, X. Huahu, M. Y. Shabir and W. Ahmad, A sentiment polarity categorization technique for online product reviews, *IEEE Access* **8** (2019) 3594–3605.
6. R. Johnson and T. Zhang, Effective use of word order for text categorization with convolutional neural networks, in *Proc. of the ACL* (2015) pp. 103–112.
7. D. Tang, B. Qin and T. Liu, Document modelling with gated recurrent neural network for sentiment classification, in *Proc. of EMNLP* (2015) pp. 1422–1432.
8. D. Tang, B. Qin and T. Liu, Learning semantic representations of users and products for document level sentiment classification, in *Proc. of the ACL* (2015) pp. 1014–1023.
9. H. Chen, M. Sun, C. Tu, Y. Lin and Z. Liu, Neural sentiment classification with user and product attention, in *Proc. of EMNLP* (2016) pp. 1650–1659.
10. J. Xu, D. Chen, X. Qiu and X. Huang, Cached long short-term memory neural networks for document-level sentiment classification, in *Proc. of the Conf. on Empirical Methods in Natural Language Processing* (2016) pp. 1660–1669.
11. C. Dos Santos and M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts, in *Proc. of COLING 2014*, (2004) pp. 69–78.
12. L. Zhang, W. Shuai and B. Liu, Deep learning for sentiment analysis: A survey. *Data Mining and Knowledge Discovery* **8**(4) (2018) e1253.
13. T. Wu, F. D. Bilbie, A. Paun, L. Pan and F. Neri, Simplified and yet Turing universal spiking neural P systems with communication on request, *International Journal of Neural Systems* **28**(8) (2018) 1850013.
14. G. Liu, W. Zhou and M. Geng, Automatic seizure detection based on S-transform and deep convolutional neural network, *International Journal of Neural Systems* **30**(4) (2020) 1950024.
15. C. Hua, H. Wang, H. Wang, S. Lu, C. Liu and S. M. Khalid, A novel method of building functional brain network using deep learning algorithm with application in proficiency detection, *International Journal of Neural Systems* **29**(1) (2019) 1850015.
16. M. O. Manzanera, S. K. Meles, K. L. Leenders, R. J. Renken, M. Pagani, D. Arnaldi, F. Nobili, J. Obeso, M. R. Oroz, S. Morbelli and N. M. Maurits, Scaled subprofile modeling and convolutional neural networks for the identification of Parkinson's disease in 3D nuclear imaging data, *International Journal of Neural Systems* **29**(9) (2019) 1950010.
17. O. Reyes and S. Ventura, Performing multi-target regression via a parameter sharing-based deep network, *International Journal of Neural Systems*, **29**(9) (2019) 1950014.
18. R. Yuvaraj, M. Murugappan, K. Sundaraj, M. I. Omar, N. M. Ibrahim, K. Mohamad, R. Palaniappan, U. R. Acharya, H. Adeli and E. Mesquita, Brain functional connectivity patterns for emotional state classification in Parkinson's disease patients without dementia, *Behavioural Brain Research* **298** (2016) 248–260.
19. A. S. Manek, P. D. Shenoy, M. C. Mohan and K. R. Venugopal, Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier, *World Wide Web* **20**(2) (2017) 135–154.
20. H. Du, X. Xu, X. Cheng, D. Wu, Y. Liu and Z. Yu, Aspect-specific sentimental word embedding for sentiment analysis of online reviews, in *Proc. of the 25th Int. Conf. Companion on World Wide Web* (2016) pp. 29–30.

21. J. Sorinas, M. D. Grima, J. M. Ferrandez and E. Fernandez, Identifying suitable brain regions and trial size segmentation for positive/negative emotion recognition, *International Journal of Neural Systems* **29**(2) (2019) 1850044.
22. J. Sorinas, J. C. Fernandez-Troyano, J. M. Ferrandez and E. Fernandez, Cortical asymmetries and connectivity patterns in the valence dimension of the emotional brain, *International Journal of Neural Systems* **30**(5) (2020) 2050021.
23. M. Val-Calvo, J. R. Alvarez-Sanchez, J. M. Ferrandez, A. Díaz-Morcillo and E. Fernandez-Jover, Real-time multi-modal estimation of dynamically evoked emotions using EEG, heart rate, and galvanic skin response, *International Journal of Neural Systems* **30**(4) (2020) 2050013.
24. P. Hajek, A. Barushka and M. Munk, Opinion mining of consumer reviews using deep neural networks with word-sentiment associations, in *AIAI* (Springer, Cham, 2020) pp. 419–429.
25. R. Moraes, J. F. Valiati and W. P. Neto, Document-level sentiment classification: An empirical comparison between SVM and ANN, *Expert Systems with Applications* **40** (2013) 621—633.
26. R. S. Wadawadagi and V. B. Pagi, Sentiment analysis with deep neural networks: Comparative study and performance assessment, *Artifcial Intelligence Review* **53**(8) 6155–6195.
27. A. Yadav and D. K. Vishwakarma, Sentiment analysis using deep learning architectures: A review, *Artifcial Intelligence Review* **53**(6) 4335–4385.
28. A. Barushka and P. Hajek, Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks, *Applied Intelligence* (2018) **48**(10) 3538—3556.
29. X. Glorot, A. Bordes and Y. Bengio, Domain adaption for large-scale sentiment classification: A deep learning approach, in *Proc. of the 28th ICML* (2011) pp. 513–520.
30. S. Zhai and Z. M. Zhang, Semisupervised autoencoder for sentiment analysis, in *Proc. of AAAI* (2016) pp. 1394–1400.
31. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems* **26** (2013) 3111–3119.
32. Q. Le and T. Mikolov, Distributed representations of sentences and documents, in *Int. Conf. on Machine Learning, JMLR* **32** (2014) pp. 1188–1196.
33. J. Devlin, M. W. Chang, K. Lee and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
34. J. Pennington, R. Socher and C. D. Manning, GloVe: Global vectors for word representation, in *Proc. of the Conf. on Empirical Methods on Natural Language Processing* (2014) pp. 1532–1543.
35. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and Ch. Potts, Learning word vectors for sentiment analysis, in *ACL 2011*, (2011) pp. 142–150.
36. Z. Li, Y. Zhang, Y. Wei, Y. Wu and Q. Yang, End-to-end adversarial memory network for cross-domain sentiment classification, in *IJCAI* (2017) pp. 2237–2243.
37. Y. Tay, A. T. Luu and S. C. Hui, Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis, in *AAAI-18* (2018) pp. 5956–5963.
38. H. Peng, Y. Ma, Y. Li and E. Cambria, Learning multi-grained aspect target sequence for Chinese sentiment analysis, *Knowledge-Based Systems* **148** (2018) 167–176.
39. M. Giménez, J. Palanca and V. J. Botti, Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis, *Neurocomputing* **378** (2020) 315–323.
40. A. A. Aziz and A. Starkey, Predicting supervise machine learning performances for sentiment analysis using contextual-based approaches, *IEEE Access* **378** (2020) 17722–17733.
41. M. Z. Asghar, A. Khan, S. R. Zahra, S. Ahmad and F. M. Kundi, Aspect-based opinion mining framework using heuristic patterns, *Cluster Computing* **22**(3) (2019) 7181–7199.
42. R. Ma, K. Wang, T. Qiu, A. K. Sangaiah, D. Lin and H. B. Liaqat, Feature-based compositing memory networks for aspect-based sentiment classification in social internet of things, *Future Generation Computer Systems* **92** (2019) 879–888.
43. Z. Zhang, Y. Zou and C. Gan, Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression, *Neurocomputing* **275** (2018) 1407–1415.
44. C. Sun, Q. Du and G. Tian, Exploiting product related review features for fake review detection, *Mathematical Problems in Engineering* **2016** (2016) 4935792.
45. P. Hajek, A. Barushka and M. Munk, Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining, *Neural Computing and Applications* **32** (2020) 17259—17274.
46. T. Mandhula, S. Pabboju and N. Gugulotu, Predicting the customer's opinion on amazon products using selective memory architecture-based convolutional neural network, *The Journal of Supercomputing* **76** (2020) 5923—5947.
47. Z. H. Deng, K. H. Luo and H. L. Yu, A study of supervised term weighting scheme for sentiment analysis, *Expert Systems with Applications* **41**(7) (2014) 3506–3513.
48. H. Kang, S. J. Yoo and D. Han, Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews, *Expert Systems with Applications* **39**(5) (2012) 6000—6010.
49. X. Zhang and Y. LeCun, Text understanding from

scratch, arXiv preprint arXiv:1502.01710 (2015).
50. S. Poria, E. Cambria and A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, *Knowledge-Based Systems* **108** (2016) 42–49.
51. W. Wang, S. J. Pan, D. Dahlmeier and X. Xiao, Recursive neural conditional random fields for aspect-based sentiment analysis, arXiv preprint arXiv:1603.06679 (2016).
52. S. Poria, I. Chaturvedi, E. Cambria and A. Hussain, Convolutional MKL based mul-timodal emotion recognition and sentiment analysis, in *2016 IEEE 16th Int. Conf. on Data Mining* (IEEE, 2016) pp. 439–448.
53. X. Gu, Y. Gu and H. Wu, Cascaded convolutional neural networks for aspect-based opinion summary, *Neural Processing Letters* **46**(2) (2017) 581–594.
54. T. Chen, R. Xu, Y. He and X. Wang, Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN, *Expert Systems with Applications* **72** (2017) 221–230.
55. M. S. Mubarok, Adiwijaya and M. D. Aldhi, Aspect-based sentiment analysis to review products using Naïve Bayes, in *AIP* (2017) **1867**(1) pp. 020060.
56. C. Catal and M. Nangir, A sentiment classification model based on multiple classifiers, *Applied Soft Computing* **50** (2017) 135—141.
57. A. S. Rathor, A. Agarwal and P. Dimri, Comparative study of machine learning approaches for Amazon reviews, *Procedia Computer Science* **132** (2018) 1552–1561.
58. D. Gamal, M. Alfonse, E. S. El-Horbaty and A. B. M Salem, Analysis of machine learning algorithms for opinion mining in different domains, *Machine Learning and Knowledge Extraction* **1**(1) (2019) 224–234.
59. P. Huang, X. Xie and S. Sun, Multi-view opinion mining with deep learning, *Neural Processing Letters* **50**(2) (2019) 1451–1463.
60. R. S. Jagdale, V. S. Shirsat and S. N. Deshmukh, Sentiment analysis on product reviews using machine learning techniques, in *Cognitive Informatics and Soft Computing* (Springer, Singapore, 2019) pp. 639–647.
61. F. Bravo-Marquez, E. Frank, S. M. Mohammad and B. Pfahringer, Determining word-emotion associations from tweets by multi-label classification, in *2016 IEEE/WIC/ACM Int. Conf. on Web Intelligence* (IEEE, 2016) pp. 536–539.
62. M. Hu and B. Liu, Mining and summarizing customer reviews, in *Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (ACM, 2004) pp. 168–177.
63. F. Bravo-Marquez, M. Mendoza and B. Poblete, Meta-level sentiment models for big social data analysis, *Knowledge-Based Systems* **69** (2014) 86–99.
64. S. Kiritchenko, X. Zhu and S. M. Mohammad, Sentiment analysis of short informal texts, *Journal of Artificial Intelligence Research* **50** (2014) 723–762.
65. S. M. Mohammad and P. D. Turney, Crowdsourcing a word–emotion association lexicon, *Computational Intelligence* **29**(3) (2013) 436–465.
66. F. Nielsen, A new ANEW: evaluation of a word list for sentiment analysis in microblogs, in *Proc. of the ESWC2011*, (2011) pp. 93–98.
67. D. M. Blei, A. Y. Ng and M. I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research* **3** (2003) 993–1022.
68. A. Asuncion, M. Welling, P. Smyth and Y. W. Teh, On smoothing and inference for topic models, in *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence* (2012) pp. 27–34.
69. J. F. Yeh, Y. S. Tan and C. H. Lee, Topic detection and tracking for conversational content by using conceptual dynamic latent Dirichlet allocation, *Neurocomputing* **216** (2016) 310–318.
70. S. Li, R. Xia, C. Zong and C. R. Huang, A framework of feature selection methods for text categorization, in *47th ACL* (2009) pp. 692–700.
71. B. Mitra, F. Diaz and N. Craswell, Learning to match using local and distributed representations of text for web search, in *Proc. of the 26th Int. Conf. on World Wide Web* (2017) pp. 1291–1299.
72. J. McAuley and J. Leskovec, Hidden factors and hidden topics: Understanding rating dimensions with review text, in *Proc. of the 7th ACM Conf. on Recommender Systems* (2013) pp. 165–172.
73. A. Mohammed and R. Kora, Deep learning approaches for Arabic sentiment analysis, *Social Network Analysis and Mining* **9**(1) (2019) 52.
74. F. Bravo-Marquez, E. Frank, B. Pfahringer and S. M. Mohammad, AffectiveTweets: a Weka package for analyzing affect in tweets, *Journal of Machine Learning Research* **20** (2019) 1–6.
75. E. Kouloumpis, T. Wilson and J. Moore, Twitter sentiment analysis: The good the bad and the omg!, in *Fifth Int. AAAI Conf. on Weblogs and Social Media* (2011) pp. 538–541.
76. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts, Learning word vectors for sentiment analysis, in *49th ACL* (2011) pp. 142-–150.
77. F. O. Gallego and R. Corchuelo, A deep-learning approach to mining conditions, *Knowledge-Based Systems* **193** (2020) 105422.
78. M. H. Rafiei and H. Adeli, A new neural dynamic classification algorithm, *IEEE Trans. on Neural Networks and Learning Systems* **28** (2017) 3074–3083.
79. K. M. R. Alam, N. Siddique and H. Adeli, A dynamic ensemble learning algorithm for neural networks, *Neural Computing and Applications* **32** (2020) 8675–8690.