

Article

A Domain-Independent Classification Model for Sentiment Analysis Using Neural Models

Nour Jnoub ^{1,*}, Fadi Al Machot ² and Wolfgang Klas ¹¹ Faculty of Computer Science, University of Vienna, 1090 Vienna, Austria; wolfgang.klas@univie.ac.at² Research Center Borstel—Leibniz Lung Center, 23845 Borstel, Germany; falmachot@fz-Borstel.de

* Correspondence: nour.jnoub@univie.ac.at

Received: 31 July 2020; Accepted: 28 August 2020; Published: 8 September 2020



Abstract: Most people nowadays depend on the Web as a primary source of information. Statistical studies show that young people obtain information mainly from Facebook, Twitter, and other social media platforms. By relying on these data, people may risk drawing the incorrect conclusions when reading the news or planning to buy a product. Therefore, systems that can detect and classify sentiments and assist users in finding the correct information on the Web is highly needed in order to prevent Web surfers from being easily deceived. This paper proposes an intensive study regarding domain-independent classification models for sentiment analysis that should be trained only once. The study consists of two phases: the first phase is based on a deep learning model which is training a neural network model once after extracting robust features and saving the model and its parameters. The second phase is based on applying the trained model on a totally new dataset, aiming at correctly classifying reviews as positive or negative. The proposed model is trained on the IMDb dataset and then tested on three different datasets: IMDb dataset, Movie Reviews dataset, and our own dataset collected from Amazon reviews that rate users' opinions regarding Apple products. The work shows high performance using different evaluation metrics compared to the state-of-the-art results.

Keywords: sentiment analysis; natural language processing; deep learning

1. Introduction

Sentiment analysis is the task of recognizing positive and negative opinions of users regarding different purposes, e.g., users' opinions about movies, products, music albums, and many other fields. To provide a better definition, the sentiment is referred to as a judgement, opinion, attitude, or emotional state prompted by feeling. Sentiment analysis is an automated process in which, by using the natural language processing (NLP), the subjective information is computationally identified, analyzed, and classified into positive, negative, or neutral to specify the sentiment of that text which is the result of its author's attitude [1]. There are different types of sentiment analysis, the most popular types are classified and described in the following:

1. Grained Sentiment Analysis: The results in this type are more than binary classification results where two labels (positive, negative) are presented. It is achieved in fine-grained granularity varying from strong negative, weakly negative, neutral, weakly positive to strong positive based on the determined polarity, mainly used when the polarity precision is highly important and binary results like negative or positive could not be useful and may provide incorrect classifications [2].
2. Emotion detection: It classifies different emotions in the text such as fear, anger, sadness, joy, disgust, etc. Sophisticated machine learning algorithms [3] are used to detect emotions for different goals.

3. Aspect-based Sentiment Analysis: The results in this type are achieved after splitting the text into different aspects and then assign each aspect a corresponding sentiment. For instance, the result of aspect-based sentiment analysis on a special product's review "It is so easy to use but Insanely Expensive" would be (a) Ease of use: positive and (b) Price: negative due to the nature of this type, it is mostly utilized in customer-centric businesses to have a deeper understanding of customer's requirements [4].

The sentiment analysis field has many applications. For example, in businesses and organizations, they need to find consumer or public opinions regarding their products and services. Individual consumers may also need to know the evaluation of other users of a product before purchasing it. Moreover, they might be interested in others' opinions concerning political candidates before making a voting decision in a political election. Furthermore, nowadays data are published enormously and freely on the Web, but with no data quality assurance, it is left to the readers to decide whether they believe it or not. This results in high demand for advanced fact checking techniques and applications that contribute to the assurance of data quality. In particular, users surfing the Web are more often inflicted with harm/damage by inconsistent information. In addition, designing and developing such fact checking systems need robust models of sentiment analysis. This task for fact checking detection can be fulfilled when fact checking systems are provided by a general or universal model that can be trained once and then applied to other reviews. Surely, this model should show high performance to increase the accuracy of such fact checking systems.

Hussein in [5] discussed the importance and effects of the challenges in sentiment analysis is the domain-dependence. Moreover, the author concluded that the nature of the topic and the review structure determine the suitable challenges for the evaluation of sentiment reviews. Hence, building a generalized model is a challenge that should be considered by researchers in this research field.

This work focuses on providing a generalized model for sentiment analysis. It has two main contributions: (a) it shows that convolutional neural networks (CNN) combined with our review to vector algorithm can lead to design models that can be trained once and work well using other types of data that might even be related to a different domain and (b) it shows high performance using different precision metrics compared to other approaches from the state of the art that use same datasets for evaluation. We believe that this is one of the few works that address the generalization capabilities of deep models w.r.t. domain-Independence.

The remainder of this paper is organized as follows: Section 2 outlines a set of related works. In Section 3, we present the proposed approach. Section 4 lists datasets that are used for evaluation purposes. In Section 5, we cover experimental results. Section 6 illustrates a detailed discussion containing the major contribution of the paper. Section 7 concludes the paper and gives an overview of future work.

2. Related Works

Although linguistics and natural language processing (NLP) have a long history of research, few works were published concerning sentiments before the year 2000 [6]. After 2000, the field has attracted the attention of researchers and many research groups to work on.

To provide an example, [7] studies the prediction of every review for being negative or positive in the aspect-oriented opinion in the opinion mining domain at the sentence level. The authors of this work propose groups of selected models based on conditional random fields (CRFs) with an added multi-label presentation that not only models the opinion in a review, but also models set of opinions in a single review. In [8], authors suggest a sentiment analysis system that is able to identify and relate the sentiment to every rated product or item in the reviews. They present a probabilistic model to investigate the structure of each review and to which cluster each of them is related to, where it represents a specific sentiment.

In [9], the researchers offer a flexible automated classification system that uses supervised machine learning techniques using Markov Logic for sentiment classification on a sub sentence level and incorporates polarity differentiations from different origins.

Furthermore, in [10], enhanced latent aspect rating analysis model is presented. This model does not require predefined keywords that are associated with specific aspects. This work investigates the reviews in order to define the topical aspects, the ratings of the individual aspect and assigning weights that differentiate depending on the aspects from a reviewer point of view. [11] proposes a simple hierarchical clustering approach (unsupervised model) for product aspects extraction, clustering, and also defining the relations between aspects (relevant and irrelevant). In [12], the authors introduce a novel supervised approach for joint topic aspects for choosing specific reviews that are considered to be helpful among a set of reviews.

Moreover, in [13], an employee dataset is created and a novel ensemble model for sentiment analysis is proposed on aspects level. In [14], a sentiment analysis is conducted on movie reviews. New features are extracted that have influence on determining the polarity scores of the opinion more accurately. Natural language processing approaches are applied using the impact of the unique extracted features. In [15], supervised and semi-supervised approaches are investigated for text classification.

Additionally, deep learning is also used for sentiment analysis. Authors of convolutional neural networks for sentence classification use CNN to classify users' reviews for movies. Others, as in [15], use bidirectional long-short term memory models which is applied to the IMDb dataset.

Table 1 shows a summary of the state-of-the-art approaches for sentiment analysis. More information regarding the performance of different approaches can be found in Section 6.

Table 1. A summary of the state-of-the-art approaches for sentiment analysis.

| Paper | Dataset | Labels | Approach |
|-------|--|---------------|---|
| [7] | Hotel Reviews | Multi-labels | Supervised machine learning techniques using conditional random fields models for aspect detection sentiment analyzing. |
| [8] | Multiple datasets (Restaurant reviews, medical descriptions, Yelp) | Two labels | Unsupervised machine learning technique using probabilistic topic modeling approaches for sentiment content clustering. |
| [9] | Product reviews | Two labels | Supervised machine learning techniques using Markov logic for sentiment classification. |
| [10] | Hotel Reviews and MP3 player product review from Amazon | 5 star rating | Unsupervised machine learning techniques for Latent Aspect Rating Analysis Model. |
| [11] | Chinese product reviews | Two labels | A hierarchical clustering approach for product aspects extraction and clustering. |
| [12] | Companies employee reviews | Ratings | A novel supervised joint topic model approach to select helpful reviews among a set of reviews. |
| [13] | Different products reviews | Ratings | A novel hybrid approach to implement aspect-level sentiment analysis that assigns sentiment labels to the reviews. |
| [14] | IMDb | Two labels | N-grams followed by a random forest classifier. |
| [15] | IMDb | Two labels | Bidirectional LSTM. |
| [16] | IMDb | Two labels | Maximum entropy classification combined with support vector machines using unigrams and bigrams. |
| [17] | IMDb | Two labels | Lexical filtering. |
| [18] | IMDb | Two labels | Context-Free Grammars (CFGs). |
| [19] | Movie Review | Two labels | Convolutional Neural Networks (CNN). |
| [20] | Movie Review | Two labels | Novel machine learning frame-work based on recursive autoencoders. |
| [21] | Movie Review | Two labels | Multiple classifiers—a hybrid approach. |

Based on the previous works, this research has started to be one of the highlights for scientific contributions because: (a) it has different applications for recommender systems and fact checking systems, and (b) it contains several challenging research problems that motivate researchers to work and improve their works on them.

3. Approach

In this section, we present the preprocessing, the review to vector algorithm and the design details of the proposed neural models for sentiment analysis, and then the evaluation metrics and the overall evaluation. We aim at training a neural model once using a batch of IMDb dataset and test it on other reviews' datasets to see how far the generalization is possible.

3.1. Review to Vector

Before features extraction, we removed the stop words from the given dataset, e.g., "the", "a", "an", and "in". Then, the next step includes removing punctuation. In this step, we extracted feature elements from a batch of IMDb dataset for positive and negative reviews. The batch size has been determined using grid search (see Section 5). We formulated a function which works like a dictionary where the keys are the words in the text and the values are the count associated with that word. The output is saved in *word_features*.

Algorithm 1 illustrates the procedure of converting reviews to vectors. It takes two inputs and returns the input vectors for all reviews saved in *all_features*. The output vectors will be fed later into our proposed neural models. The input parameters are *word_features* which is the first 4900 words after calculating the frequency distribution of each word from both training data, mainly the positive and negative reviews. The number 4900 words have been selected after applying grid search using different lengths which give the highest performance. The second input parameter is the *reviews*. The summary of the algorithm is as follows: for each review in the reviews, the function *word_tokenize* splits the current review into sub-strings (words). After that, for each word in *word_features*, it should be checked whether that word is a word in the current review. If yes, 1 is added to the features list or 0 is added if it is not. Finally, the function returns *all_features*, which is a matrix.

The vectors of the review to vector algorithm (see Algorithm 1) are used for training different classification models (see Section 3.2).

3.2. Classification

3.2.1. Convolutional Neural Network (CNN)

To perform the sentiment classification task, we propose a neurocomputing-based approach. A CNN is a kind of feed-forward network structure that consists of multiple layers of convolutional filters followed by subsampling filters and ends with a fully connected classification layer. The classical LeNet-5CNN was first proposed by LeCun et al. [22], which is the basic model of different CNN applications for object detection, localization, and prediction. First, the output vectors of the review to vector Algorithm 1 are converted to matrices where the goal is to make the application of CNN model possible. As illustrated in Figure 1, the proposed CNN model has one convolutional layer, one subsampling layer, and an output layer.

The convolutional layers generate feature maps using five (2×2) filters followed by a Scaled Exponential Linear Units (SELU) [23] as an activation function. Additionally, in the subsampling layers, the generated feature maps are spatially dissembled. In our proposed model, the feature maps in layers are subsampled to a corresponding feature map of size 2×2 in the subsequent layer.

The final layer, which is a fully CNN model that performs the classification process, consists of three layers. The first layer is the input layer which has 6125 nodes and the second that has five nodes. Each SELU activation function.

The final layer is the softmax output layer. The result of the mentioned layers is a 2D representation of extracted features from input feature map(s) based on the input features for the reviews.

Algorithm 1

```

1: Input: word_features, Reviews
2:
3: function REVIEW_TO_VECTOR(Reviews)
4:
5:   all_features = []
6:
7:   for review in reviews do
8:
9:     words = word_tokenize(review)
10:
11:    features = []
12:
13:    for w in word_features do
14:
15:      if (w in words) then
16:
17:        Features[w] = 1
18:
19:      else
20:
21:        Features[w] = 0
22:
23:      end if
24:
25:    end for
26:
27:    all_features+=[Features]
28:
29:  end for
30: return all_features
31: end function
32:

```

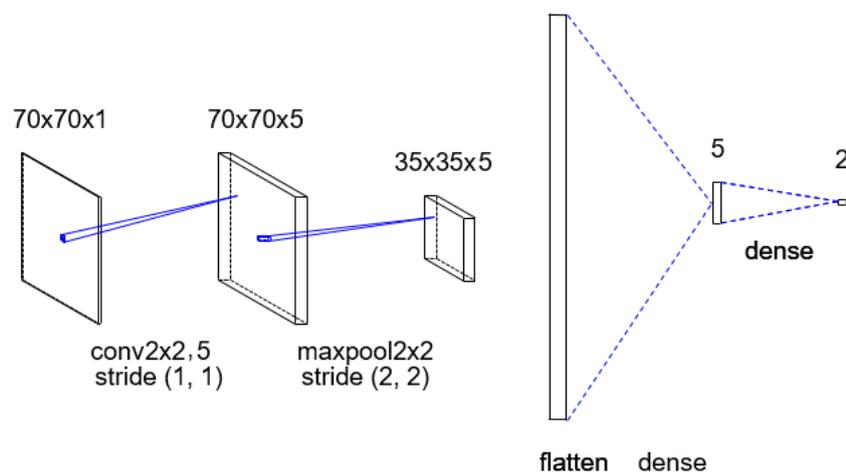


Figure 1. The proposed CNN model.

The proposed CNN consists of one convolutional layer and a max-pooling layer is because the small size of our input dimension does not require additional layers to extract features/patterns. The reason for using SELU is due to the fact that (a) SELUs performed better than Rectified Linear Units (RELUs), (b) SELUs offer self-normalization [23], and (c) they never lead to vanishing gradients problem. Since the dropout is a regularization technique to avoid over-fitting in neural networks

based on preventing complex co-adaptations on training data [24], our dropout for each layer was 0.75, which is related to the fraction of the input units to drop.

The proposed CNN model has been trained on IMDB. Then, the model has been saved to be tested on other datasets. The length of the considered feature vectors is 4900 words that are converted to matrices of size (70 × 70).

The parameters of CNN are selected by using grid search from a scikit-learn library considering different settings. Table 2 shows parameters used for all the layers of the proposed CNN model.

Table 2. Parameters used for all the layers of the proposed CNN model.

| Layer | Kernel, Units | Other Layers Parameters |
|-----------------|---------------|--------------------------------|
| Convolution | (2 × 2), 5 | Activation = Selu, Strides = 1 |
| Max Pooling | (2 × 2) | Strides = 2 |
| Dropout | 0.75 | |
| Fully Connected | | Units = 5, Activation = Selu |
| Softmax | - | NumbrOfClasses = 2 |

3.2.2. Shallow Neural Network (SNN)

To perform the sentiment classification task, we use a neural model [25,26]. First, the output vectors of the review to vector algorithm are fed into the neural model to the hidden layer which consists of three neurons and a hyperbolic activation function; then, the final layer is the output layer which consists of a softmax activation function, Adam optimizer [27], and a cross entropy loss function. The parameters are selected by using a grid search from scikit-learn library (<https://scikit-learn.org>, see Figure 2), where the optimizer is Adam and the loss function is the binary cross entropy.

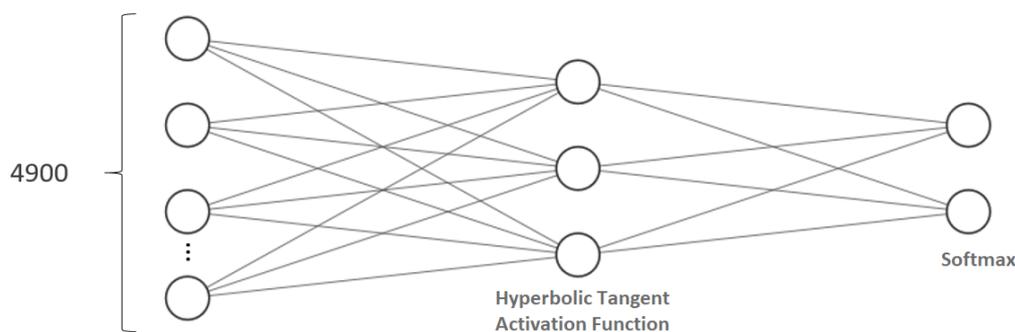


Figure 2. The proposed SNN model.

The proposed neural network model has been trained on a batch of IMDB datasets. Then, the model has been saved to be tested on other datasets. Table 3 shows parameters used for all the layers of the proposed CNN model.

Table 3. Parameters used for all the layers of the proposed SNN model.

| Layer | Units | Other Layers' Parameters |
|--------------|-------|---|
| Hidden layer | 3 | Activation = Hyperbolic tangent activation (tanh) |
| Output layer | 2 | Activation = softmax |

3.2.3. Other Classifiers

Additionally, we examine several classifiers to compare the performance of the existing models and the proposed ones, particularly Support Vector Machines (SVM) [28], K-Nearest Neighbor (KNN) [29], Naive Bayes [30], and Random Forest [31]. In addition, selecting the previous classifiers has

different advantages such as the objective of random forests that they consider a set of high-variance, low-bias decision trees, and the ability to convert them into a model that has both low variance and low bias. On the other hand, K-nearest neighbors is an algorithm which stores all the available cases and classifies new cases based on a similarity measure (e.g., distance functions). Therefore, KNN has been applied in statistical estimation and pattern recognition from the beginning of 1970s on as a non-parametric technique [29]. SVM are well-known in handling non linearly separable data based on their nonlinear kernel; e.g., SVM with a polynomial kernel (SVM (poly)) and the SVM with a radial basis kernel (SVM (rbf)). Therefore, we classify the reviews data using three types of SVMs; the standard linear SVM (SVM (linear)), SVM (poly), and SVM (rbf). Finally, we used a simple probabilistic model which is the Naive Bayes. The purpose of using such a probabilistic model is to show how it behaves w.r.t. different contexts.

Table 4 shows values of parameters for the proposed SNN, CNN, and all other classifiers.

Table 4. Values of parameters of proposed CNN, SNN, and other classifiers.

| Model | Parameters |
|----------------|---|
| SVM (poly) | Degree of the polynomial kernel function = 3, $\gamma = \frac{1}{\text{numberoffeatures}}$ |
| SVM (rbf) | $\gamma = \frac{1}{\text{numberoffeatures}}$ |
| Random Forest | Number of estimators estimators =10 trees, criterion = Gini impurity, The minimum number of samples required to split an internal node = 2 |
| Naive Bayes | Prior = probabilities of the classes |
| Proposed (CNN) | Loss = Softmax, optimizer = Adamax, <i>batch_size</i> =1000, epochs = 30 |
| Proposed (SNN) | Loss = cross entropy, optimizer = Adam, <i>batch_size</i> =128, epochs = 40, lr = 0.001, <i>beta</i> ₁ = 0.9, <i>beta</i> ₂ = 0.999, epsilon = 0.01, decay = 0.0 |

3.3. Evaluation Metrics and Validation Concept

To evaluate the overall performance of the classifiers, we consider several performance metrics. In particular, we use precision, recall, f-measure, and accuracy, as in [32].

Equations (1)–(4) show mathematical expressions of the metrics accuracy, precision, recall, and f-measure, respectively, where TP, TN, FP, and FN refer respectively to “True Positives”, “True Negatives”, “False Positives”, and “False Negatives”, respectively:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{4}$$

Regarding the evaluation scenarios, we consider two cases: the domain-dependent and domain-independent cases. Domain-dependent means training and testing have been performed for each dataset. Domain-independent means the training has been performed on a IMDB datasets of subjects and testing has been performed on a totally new datasets. The reason for training on IMDB is due to its large size and thus can support a better generalized model if the training has been preformed and regularized properly.

4. Datasets

4.1. IMDb Dataset

ACL-IMDb [33] dataset is a collection of reviews that are taken from Internet Movie Database (IMDb). The dataset size is 50 K and contains highly polar movie reviews annotated as positive or negative review, which makes it widely used for a binary classification tasks. The average length of a document in the training set is 25 k for training and 25 k for testing. The dataset also contains an additional bag of words formats and raw texts (<http://ai.stanford.edu/~amaas/data/sentiment/>).

4.2. Movie Reviews (MR)

Movie Reviews (MR) is a small sized dataset (<https://www.cs.cornell.edu/people/pabo/movie-review-data/>) [34] (5 k positive and 5 k negative reviews) that contains reviews in the form of labeled sentences, which can be specified as objective or subjective. Furthermore, the selected sentences have been gathered from IMDb and Rotten Tomatoes websites (<https://www.rottentomatoes.com/>), each selected sentence contains at least 10 words. The sentiments of these sentences have been classified as positive or negative.

4.3. Amazon Dataset (Amazon)

Reviews data from iPhone wireless earphones on Amazon were collected. Overall, we collected 480 negative reviews and 480 positive ones in order to use the data to check the overall performance of the proposed model. We annotated the data with 0 or 1 where every positive review is annotated or labeled by 1 and the negative reviews were annotated by 0.

5. Results

In this section, we want to demonstrate the performance of the proposed approach. The prototype is implemented in Python. In order to gain sufficient information and prove the applicability of our approach, the following libraries have been used: NLTK (<https://www.nltk.org/>) library for natural language processing, Keras (<https://keras.io/>) (Deep learning) and scikit-learn (<https://scikit-learn.org/stable/>) (machine learning), which is mainly used for testing the performance of the other classifiers. We applied 10-fold cross-validation for performance evaluation. The neural models have been trained on a GeForce GTX 1080-NVIDIA (<https://www.nvidia.com/de-de/geforce/products/10series/geforce-gtx-1080/>).

In order to conduct experimental results and check the performance of the proposed approach, we tested the algorithm using the extracted features based on three datasets, namely IMDb, Movie Reviews, and Amazon reviews.

To evaluate the overall performance of the classifiers, we consider several performance metrics. In particular, we use precision, recall, f1, and accuracy, as in [32].

Regarding the evaluation scenarios, we used the trained model in Section 3.2.

Tables 5–7 present the precision, the recall, and the f-measure using IMDb, Movie Reviews, and Amazon datasets, respectively.

Table 5. Performance metrics for IMDb, where SVM (poly): Support Vector Machine using a polynomial kernel, SVM (rbf): Support Vector Machine using a radial basis function kernel.

| Classifier | Precision | Recall | F-Measure | Accuracy |
|---------------------|-------------|-------------|-------------|-------------|
| Random Forest | 0.73 | 0.73 | 0.73 | 0.73 |
| Naive Bayes | 0.64 | 0.59 | 0.56 | 0.59 |
| SVM (poly) | 0.24 | 0.29 | 0.33 | 0.49 |
| SVM (rbf) | 0.24 | 0.29 | 0.33 | 0.49 |
| Proposed SNN | 0.87 | 0.87 | 0.87 | 0.87 |
| Proposed CNN | 0.81 | 0.81 | 0.81 | 0.81 |

Table 6. Performance metrics for MR, where SVM (poly): Support Vector Machine using a polynomial kernel, SVM (rbf): Support Vector Machine using a radial basis function kernel.

| Classifier | Precision | Recall | F-Measure | Accuracy |
|---------------------|-----------|--------|-----------|----------|
| Random Forest | 0.67 | 0.66 | 0.65 | 0.66 |
| Naive Bayes | 0.58 | 0.58 | 0.57 | 0.58 |
| SVM (poly) | 0.23 | 0.48 | 0.31 | 0.48 |
| SVM (rbf) | 0.75 | 0.61 | 0.55 | 0.48 |
| Proposed SNN | 0.82 | 0.82 | 0.82 | 0.82 |
| Proposed CNN | 0.75 | 0.75 | 0.75 | 0.75 |

Table 7. Performance metrics for Amazon reviews, where SVM (poly): Support Vector Machine using a polynomial kernel, SVM (rbf): Support Vector Machine using a radial basis function kernel.

| Classifier | Precision | Recall | F-Measure | Accuracy |
|---------------------|-----------|--------|-----------|----------|
| Random Forest | 0.80 | 0.80 | 0.80 | 0.80 |
| Naive Bayes | 0.71 | 0.67 | 0.67 | 0.67 |
| SVM (poly) | 0.31 | 0.55 | 0.40 | 0.55 |
| SVM (rbf) | 0.31 | 0.55 | 0.40 | 0.55 |
| Proposed SNN | 0.77 | 0.76 | 0.74 | 0.74 |
| Proposed CNN | 0.67 | 0.67 | 0.67 | 0.68 |

In all tables, the proposed neural models show the highest performance compared to random forest which is hereby the next best classifier. However, the support vector machine using a radial basis function kernel also performs well for the IMDb dataset, but the Naive Bayes classifier performs much better using our own Amazon dataset.

Additionally, it is remarkable to realize that the proposed shallow neural network performs better than the proposed convolutional neural network model. However, random forest and our proposed neural models show a robust behavior regarding sentiment classification.

Moreover, it should be realized that some classifiers show high precision and low recall or vice versa where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. This reflects the complexity of this classification task and shows the robust performance of the proposed neural models.

Furthermore, the trained neural models are applied on different datasets that are not related to each other (see Table 8). Despite of this fact, it still behaves well and, consequently, it can be extended for different applications in the research field of sentiment analysis.

Table 8. Performance metrics for Amazon reviews and MR reviews using the pretrained neural models on the IMDb dataset.

| Classifier | Precision | Recall | F-Measure | Accuracy |
|---------------------|-----------|--------|-----------|----------|
| SNN (Amazon) | 0.66 | 0.65 | 0.64 | 0.64 |
| CNN (Amazon) | 0.65 | 0.64 | 0.64 | 0.64 |
| SNN (MR) | 0.82 | 0.82 | 0.82 | 0.82 |
| CNN (MR) | 0.80 | 0.80 | 0.80 | 0.80 |

Figures 3 and 4 show the 10-folds cross-validation results for the trained CNN and SNN models, respectively. In addition, they show the mean accuracy and the standard deviation for each fold. We can observe a reasonable symmetric distribution and that the mean captures the central tendency well. The cross-validation results belong to the pre-trained model which has been applied to calculate the results in Table 8.

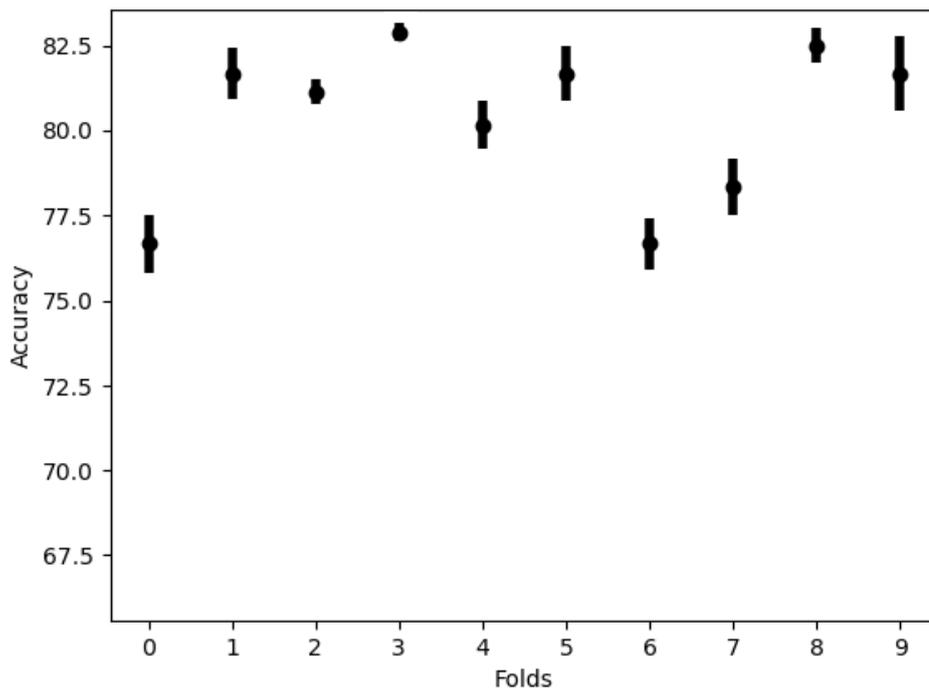


Figure 3. The cross-validation results for the trained CNN model using IMDb model, which has been used on MR and Amazon datasets.

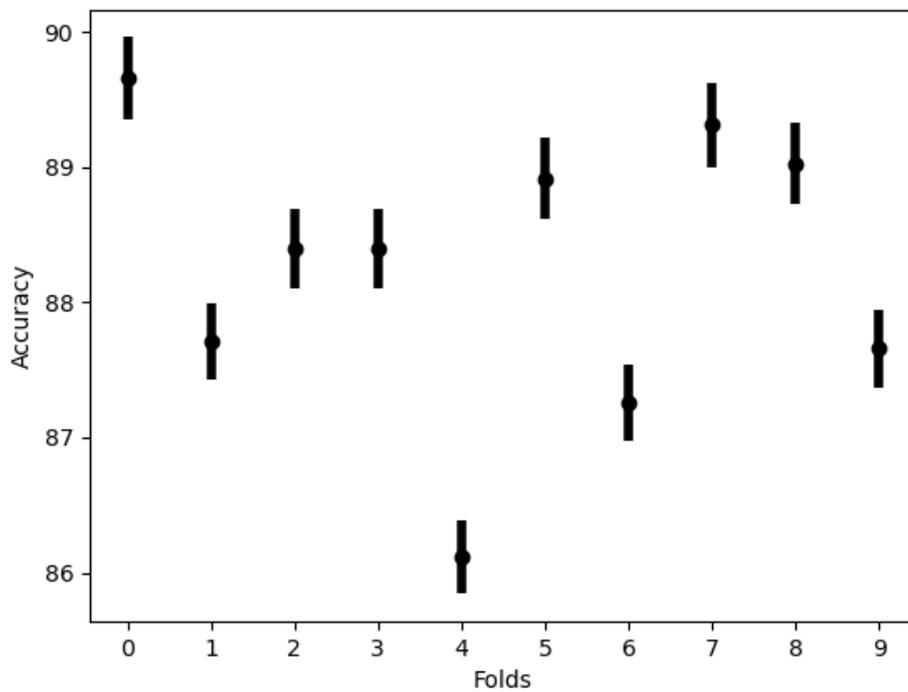


Figure 4. The 10-fold cross-validation results for the trained SNN model using the IMDb model, which has been used on the MR and Amazon datasets.

6. Discussion

Based on our results, we could demonstrate the following points:

1. Summarizing the final opinion in some words might lead to the problem that the extracted features did not take that sentence as a feature of interest. For example, a reviewer might have a

- positive opinion about acting and the overall story of a movie, but he was not satisfied by the music in certain scenes.
2. Sentiments can be expressed in different forms that might be even indirect expressions. Therefore, they require common sense reasoning techniques to be classified. In addition, it is challenging to analyze sentiments with complex structures of sentences, especially when negations do exist.
 3. Some reviewers may use expressions that have negative connotations, but at the end of the review, they summarize their overall opinion clearly. Consequently, this makes the classification a tough task.
 4. Feature engineering suffers from overcoming the previous problems.
 5. Sentiment analysis requires analyzing large units of individual words to capture the context in which those words appear.
 6. Table 8 shows that SNN shows a better generalization performance using the MR dataset compared to CNN. The results of other approaches are listed in Tables 9 and 10 for movie review data and IMDb, respectively. Some of the proposed approaches perform better; however, they do not consider the domain-independent sentiment classification. It means that the results are obtained for training and testing on the same dataset.

Table 9. A summary of state-of-the-art performance metrics for MR, where DCNN: Dynamic Convolutional Neural Network, SVM: Support Vector Machine.

| Paper | Classifier Used | Accuracy |
|-------|-----------------|----------|
| [19] | CNN | 0.81 |
| [20] | Autoencoders | 0.77 |
| [35] | SVM | 0.77 |
| [36] | DCNN | 0.86 |

Table 10. A summary of the state-of-the-art precision metrics for IMDb, where SVM: Support Vector Machine.

| Paper | Classifier Used | Accuracy |
|-------|----------------------|----------|
| [16] | SVM | 0.82 |
| [17] | Markov Model | 0.80 |
| [14] | Random Forest | 0.88 |
| [18] | statistical approach | 0.87 |

We can observe that this work is:

1. Able to classify binary reviews very well, especially, domain-independent reviews.
2. The first building block toward the generalization of sentiment classification where a model can be trained once and tested on totally new datasets that even may come from different contexts.
3. It inherits the advantages of neural models which is nowadays able to classify hundreds of objects using a pre-trained model.

This is due to the fact that CNN can overcome many challenges of sentiment analysis that have been highlighted previously. For example, words in a specific region are more likely to be related than words far away. Thus, CNN can automatically and adaptively extract spatial hierarchies of features out of written reviews that may capture different writing styles of users.

7. Conclusions

In this work, we proposed a sentiment analysis generalized approach that is able to classify the sentiments of different datasets robustly. Additionally, the proposed approach showed promising results in the context of domain-independent sentiment analysis. This is due to the fact that neural models can extract robust features when reviews are converted to proper input vectors using our proposed review for vector algorithms. Furthermore, it shows a high performance regarding generalization. The proposed model has been trained once and tested on three different datasets from different domains. The model could perform very well compared to other works that used the same datasets and showed a generalization capability for sentiment classification w.r.t. different domains. Furthermore, the paper covered a wide range of sentiment analysis approaches from the state of the art and compared the results obtained to the performance of the proposed neural models.

Additionally, in our future work, we will integrate the implemented version of the algorithm into different browsers and platforms aiming at using the power of this approach for fact checking purposes.

Author Contributions: Conceptualization: N.J.; Methodology: N.J. and F.A.M.; Supervision: F.A.M. and W.K.; Formal analysis: N.J., F.A.M., and W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest. The authors ensure that there are no personal circumstances, interest, or sponsors that may be perceived as inappropriately influencing the representation or interpretation of reported research results.

References

1. Cambria, E.; Das, D.; Bandyopadhyay, S.; Feraco, A. *A Practical Guide to Sentiment Analysis*; Springer: Berlin/Heidelberg, Germany, 2017.
2. Wang, Z.; Chong, C.S.; Lan, L.; Yang, Y.; Ho, S.B.; Tong, J.C. Fine-grained sentiment analysis of social media with emotion sensing. In Proceedings of the 2016 Future Technologies Conference (FTC), San Francisco, CA, USA, 6–7 December 2016; pp. 1361–1364.
3. Suhasini, M.; Srinivasu, B. Emotion Detection Framework for Twitter Data Using Supervised Classifiers. In *Data Engineering and Communication Technology*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 565–576.
4. Zainuddin, N.; Selamat, A.; Ibrahim, R. Discovering Hate Sentiment within Twitter Data through Aspect-Based Sentiment Analysis. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2020; Volume 1447, p. 012056.
5. Hussein, D.M.E.D.M. A survey on sentiment analysis challenges. *J. King Saud Univ. Eng. Sci.* **2018**, *30*, 330–338. [[CrossRef](#)]
6. Liu, B. Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* **2012**, *5*, 1–167. [[CrossRef](#)]
7. Marcheggiani, D.; Täckström, O.; Esuli, A.; Sebastiani, F. Hierarchical multi-label conditional random fields for aspect-oriented opinion mining. In *European Conference on Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 273–285.
8. Sauper, C.; Barzilay, R. Automatic aggregation by joint modeling of aspects and values. *J. Artif. Intell. Res.* **2013**, *46*, 89–127. [[CrossRef](#)]
9. Zirn, C.; Niepert, M.; Stuckenschmidt, H.; Strube, M. Fine-grained sentiment analysis with structural features. In Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, 8–13 November 2011; pp. 336–344.
10. Wang, H.; Lu, Y.; Zhai, C. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2011; pp. 618–626.
11. Zhao, Y.; Qin, B.; Liu, T. Clustering product aspects using two effective aspect relations for opinion mining. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 120–130.

12. Hai, Z.; Cong, G.; Chang, K.; Liu, W.; Cheng, P. Coarse-to-fine review selection via supervised joint aspect and sentiment model. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*; ACM: Berlin/Heidelberg, Germany, 2014; pp. 617–626.
13. Bajpai, R.; Hazarika, D.; Singh, K.; Gorantla, S.; Cambria, E.; Zimmerman, R. Aspect-Sentiment Embeddings for Company Profiling and Employee Opinion Mining. *arXiv* **2019**, arXiv:1902.08342.
14. Sahu, T.P.; Ahuja, S. Sentiment analysis of movie reviews: A study on feature selection & classification algorithms. In *Proceedings of the 2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*, Durgapur, India, 23–25 January 2016; pp. 1–6.
15. Sachan, D.S.; Zaheer, M.; Salakhutdinov, R. Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function. *Proc. AAAI Conf. Artif. Intell.* **2019**, 6940–6948. [[CrossRef](#)]
16. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs Up? Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing—Volume 10 (EMNLP '02)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 79–86. [[CrossRef](#)]
17. Salvetti, F.; Lewis, S.; Reichenbach, C. Automatic Opinion Polarity Classification of Movie Reviews. *Colo. Res. Linguist.* **2004**, *17*. [[CrossRef](#)]
18. Dong, L.; Wei, F.; Liu, S.; Zhou, M.; Xu, K. A statistical parsing framework for sentiment classification. *Comput. Linguist.* **2015**, *41*, 293–336. [[CrossRef](#)]
19. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
20. Socher, R.; Pennington, J.; Huang, E.H.; Ng, A.Y.; Manning, C.D. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; pp. 151–161.
21. Tsutsumi, K.; Shimada, K.; Endo, T. Movie review classification based on a multiple classifier. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation*, Seoul, Korea, 1–3 November 2007; pp. 481–488.
22. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995.
23. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc: Red Hook, NY, USA, 2017; pp. 971–980.
24. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
25. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
26. Funahashi, K.I. On the approximate realization of continuous mappings by neural networks. *Neural Netw.* **1989**, *2*, 183–192. [[CrossRef](#)]
27. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
28. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
29. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185.
30. Webb, G.I. Naïve Bayes. In *Encyclopedia of Machine Learning and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 895–896.
31. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
32. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63. [[CrossRef](#)]
33. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies—Volume 1*; Association for Computational Linguistics: Portland, OR, USA, 2011; pp. 142–150.
34. Pang, B.; Lee, L. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *arXiv* **2004**, arXiv:cs/0409058.

35. Yessenalina, A.; Yue, Y.; Cardie, C. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 1046–1056.
36. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).