SEMTec: Social Emotion Mining Techniques for Analysis and Prediction of Facebook Post Reactions

Tobias Moers, Florian Krebs, and Gerasimos Spanakis(✉)

Department of Data Science and Knowledge Engineering, Maastricht University, 6200 MD Maastricht, The Netherlands
{tobias.moers,florian.krebs}@student.maastrichtuniversity.nl, jerry.spanakis@maastrichtuniversity.nl

Abstract. Nowadays social media are utilized by many people in order to review products and services. Subsequently, companies can use this feedback in order to improve customer experience. Facebook provided its users with the ability to express their experienced emotions by using five so-called 'reactions'. Since this launch happened in 2016, this paper is one of the first approaches to provide a complete framework for evaluating different techniques for predicting reactions to user posts on public pages. For this purpose, we used the FacebookR dataset that contains Facebook posts (along with their comments and reactions) of the biggest international supermarket chains. In order to build a robust and accurate prediction pipeline state-of-the-art neural network architectures (convolutional and recurrent neural networks) were tested using pretrained word embeddings. The models are further improved by introducing a bootstrapping approach for sentiment and emotion mining on the comments for each post and a data augmentation technique to obtain an even more robust predictor. The final proposed pipeline is a combination of a neural network and a baseline emotion miner and is able to predict the reaction distribution on Facebook posts with a mean squared error (or misclassification rate) of 0.1326.

Keywords: Emotion mining · Social media · Deep learning
Natural language processing

1 Introduction

The ubiquitous use of social media has raised the need to improve techniques of analyzing short text messages' content and improve performance on tasks like topic modeling, topic classification, sentiment analysis, etc. Social media pages related to (and managed by) firms/companies are drowned in content posted every day by users who share their customer experience. These large amounts of

T. Moers and F. Krebs—Equal contribution.

J. van den Herik and A. P. Rocha (Eds.): ICAART 2018, LNAI 11352, pp. 361–382, 2019.
https://doi.org/10.1007/978-3-030-05453-3_17

data can be further analyzed and grasp the feelings, emotions and sentiments of the users which has yielded many research works with applications in political science, social sciences, business, education, etc. [1–3].

Customer experience (CX) represents a holistic perspective on customer interactions with a firm's products and/or services. If managers have enough information about customer experiences with product and service offerings, then it is possible to quantify these and through standardized measurements to improve future actions and decisions. The rise of social media analytics [4] offers managers a tool to manage this process since customer data (in terms of reviews and content sharing) are widely available in social media.

This paper is building on authors' previous work [5] on identifying the sentiment and emotion of Facebook posts and trying to predict user reactions and to our knowledge it was the first research work on working with Facebook posts reactions. Analyzing Facebook posts can help firm managers to better manage posts by allowing customer care teams to reply faster to unsatisfied customers or maybe even delegate posts to employees based on their expertise. Also, it would be possible to estimate how the reply on a post affects the reaction from other customers.

The main goals and contributions of this paper are the following: (a) highlight the use of an (augmented) dataset which can be used for predicting reactions on Facebook posts, useful for both machine learners and marketing experts and (b) perform improved sentiment analysis and emotion mining to Facebook posts and comments of several supermarket chains by predicting the distribution of the user reactions. Firstly, sentiment analysis and emotion mining baseline techniques are utilized in order to analyze the sentiment/emotion of a post and its comments. Afterwards, neural networks with pretrained word embeddings are used in order to accurately predict the distribution of reactions to a post. Combination of the two approaches gives a working final ensemble which leaves promising directions for future research.

The remainder of the paper is organized as follows. Section 2 presents related work about sentiment and emotion analysis on short informal text like from Facebook and Twitter. The dataset along with any pre-processing and augmentation steps are described in Sect. 3, followed by the model (pipeline) description in Sect. 4. Section 5 presents the detailed experimental results and finally, Sect. 6 concludes the paper and presents future research directions.

2 Related Work

Deep learning based approaches have recently become more popular for sentiment classification since they automatically extract features based on word embeddings. Convolutional Neural Networks (CNN), originally proposed in [6] for document recognition, have been extensively used for short sentence sentiment classification. [7] uses a CNN and achieves state-of-the art results in sentiment classification. They also highlight that one CNN layer in the model's architecture is sufficient to perform well on sentiment classification tasks. Recurrent

Neural Networks (RNN) and more specifically their variants Long Short Term Memory (LSTM) networks [8] and Gated Recurrent Units (GRU) networks [9] have also been extensively used for sentiment classification since they are able to capture long term relationships between words in a sentence while avoiding vanishing and exploding gradient problems of normal recurrent network architectures [10]. [11] proves that combining different architectures, such as CNN and GRU, in an ensemble learner improves the performance of individual base learners for sentiment classification, which makes it relevant for this research work as well.

Most of the work on short text sentiment classification concentrates around Twitter and different machine learning techniques [12–15]. These are some examples of the extensive research already done on Twitter sentiment analysis. Not many approaches for Facebook posts exist, partly because it is difficult to get a labeled dataset for such a purpose.

Emotion lexicons like EmoLex [16] can be used in order to annotate a corpus, however, results are not satisfactory and this is the reason that bootstrapping techniques have been attempted in the past. For example, [17] propose such a technique which enhances EmoLex with synonyms and then combines word vectors [18] in order to annotate more examples based on sentence similarity measures.

Recently, [19] presented some first results which associate Facebook reactions with emojis but their analysis stopped there. [20] utilized the actual reactions on posts in a distant supervised fashion to train a support vector machine classifier for emotion detection but they are not attempting at actually predicting the distribution of reactions.

Moreover, analysis of customer feedback is an area which gains interest for many companies over the years. Given the amount of text feedback available, there are many approaches around this topic, however none of them are handling the increasing amounts of information available through Facebook posts. For the sake of completeness, we highlight here some these approaches. Sentiment classification [21–23] deals only with the sentiment analysis (usually mapping sentiments to positive, negative and neutral (or other 5-scale classification)) and similarly emotion classification [24,25] only considers emotions. Some work exists on Twitter data [26] but does not take into account the reactions of Facebook. Moreover, work has been conducted towards customer review analysis [27–29] but none of them are dealing with the specific nature of Facebook (or social media in general).

Due to the lack of enough labeled data, data augmentation is necessary to extend the dataset for systems like neural networks. Typically, data augmentation for images is done by adding noise, or transforming the image like rotating or scaling [30]. Also, for time-series data of signals like sensory data, one is able to add a certain amount of noise to augment the dataset. Recent approaches like [31–34] augment image data by using Generative Adversarial Networks (GANs) to generate new data that is based on the given data distribution. However, augmenting text is still a weakly researched area as it is a complex problem. There

is no Gaussian noise, no rotation or translation that can be made to augment the text. Instead, [35] use a thesaurus to enhance the dataset by replacing words of the training text with synonyms. They report that they receive the best results when using the thesaurus data augmentation.

In this work, we show how we can create a big enough dataset using standard NLP tools and augmentation techniques. Then we demonstrate how the combination of traditional sentiment and emotion mining techniques with modern neural network architectures can help accurately predicting the distribution of reactions on Facebook posts.

3 Dataset Construction

Our dataset consists out of Facebook posts on the customer service page of 12 US/UK big supermarket/retail chains, namely Tesco, Sainsbury, Walmart, Aldi UK, The Home Depot, Target, Walgreens, Amazon, Best Buy, Safeway, Macys and publix. The vast majority of these posts are initiated by customers of these supermarkets. In addition to the written text of the posts, we also fetch the Facebook's reaction matrix[1] as well as the comments attached to this post made by other users. Such reactions only belong to the initial post, and not to replies to the post since the feature to post a reaction on a reply has only been introduced very recently (May 2017) and would result in either a very small dataset or an incomplete dataset. These reactions include *like*, *love*, *wow*, *haha*, *sad*, *angry* as shown in Fig. 1. This form of communication was introduced by Facebook on February 24th, 2016 and allows users to express an 'emotion' towards the posted content.

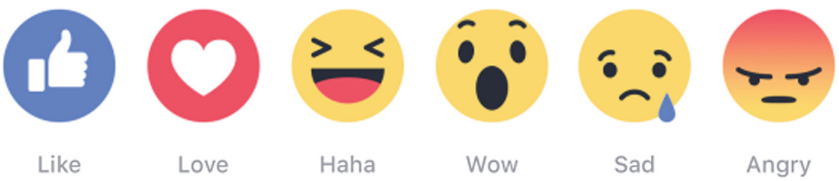

Fig. 1. The Facebook reaction icons that users are able to select for an original post [5].

In total, there were more than 70,000 posts without any reaction. Apart from this problem, people are using the 'like' reaction not only to show that they like what they see/read but also to simply tell others that they have seen this post or to show sympathy. This results in a way too often used 'like'-reaction which is why likes could be ignored in the constructed dataset. So, instead of using all crawled data, the developed models will be trained on posts that have at least one other reaction than likes. After applying this threshold the size of the training

[1] http://newsroom.fb.com/news/2016/02/reactions-now-available-globally/.

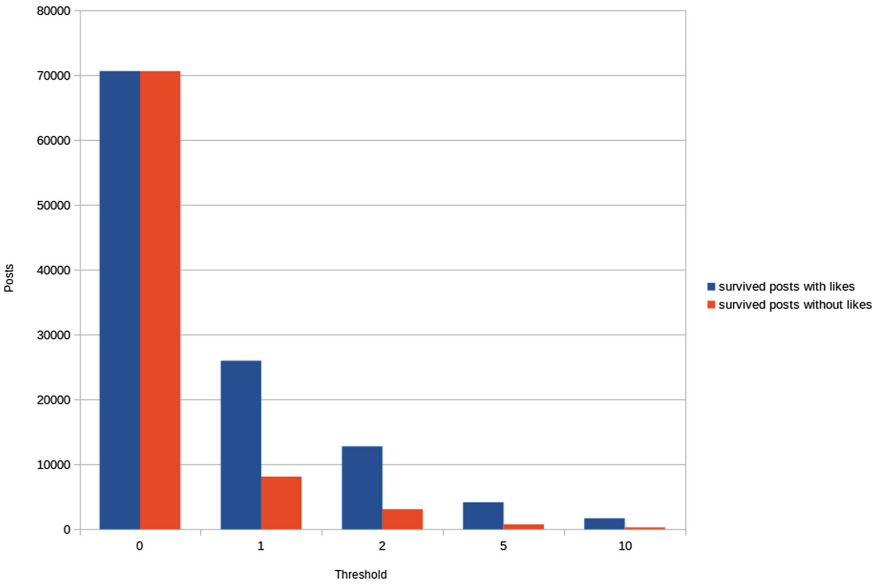

Fig. 2. Amount of survived posts for different thresholds including/excluding likes [5].

set reduced from 70,649 to 25,969. The threshold of 1 is still not optimal since it leaves much space for noise in the data (e.g. miss-clicked reactions) but using a higher threshold will lead to extreme loss of data. Statistics on the dataset and on how many posts 'survive' by using different thresholds can be seen in Fig. 2.

Exploratory analysis on the dataset shows that people tend to agree in the reactions they have to Facebook posts (which is consistent for building a prediction system), i.e. whenever there are more than one types of reactions they seem to be the same in a great degree (over 80%) as can be seen in Fig. 3. In addition, Fig. 4 shows that even by excluding the *like* reaction, which seems to dominate all posts, the distribution of the reactions remains the same, even if the threshold of minimum reactions increases. Using all previous insights and the fact that there are 25,969 posts with at least one reaction and since the *like* reaction dominates the posts, we chose to include posts with at least one reaction which is not a *like*, leading to finally 8,103 posts. Full dataset is available[2].

3.1 Pre-processing

Pre-processing on the dataset is carried out using the Stanford CoreNLP parser [36] and includes the following steps:

- Convert everything to lower case
- Replace URLs with "__URL__" as a generic token
- Replace user/profile links with "__AT_USER__" as a generic token

[2] https://github.com/jerryspan/FacebookR.

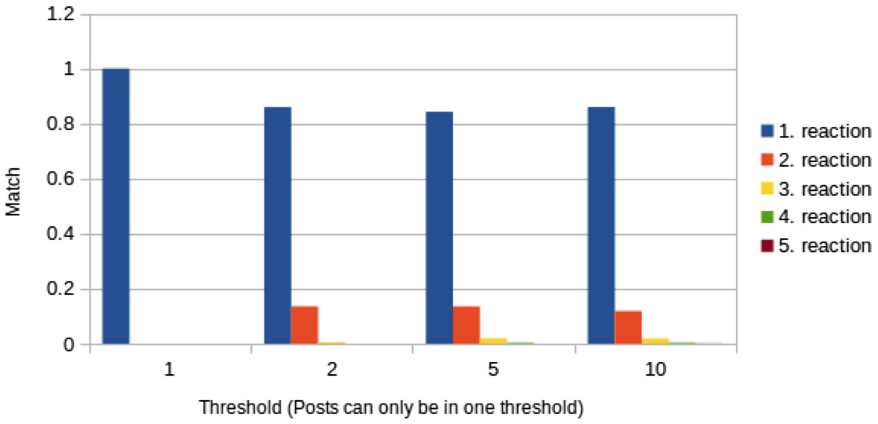

Fig. 3. Reaction match when there is more than one type [5].

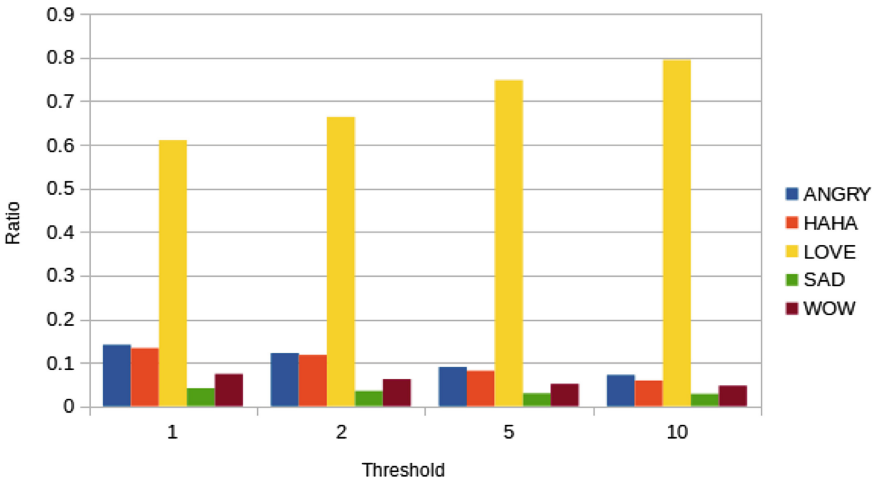

Fig. 4. Distribution of reactions with different minimum thresholds [5].

– Remove the hash from a hashtag reference (e.g. #hashtag becomes "hashtag")
– Replace three or more occurrences of one character in a row with the character itself (e.g. "looooove" becomes "love")
– Remove sequences containing numbers (e.g. "gr34t").

Afterwards, each post is split using a tokenizer based on spaces and after some stop-word filtering the final list of different tokens is derived. Since preprocessing on short text has attracted much attention recently [37], we also demonstrate the effect of it on the developed models in the Experiments section.

3.2 Data Augmentation

As mentioned in the previous subsection, our final dataset consists of 8,103 relevant posts. However, this is a relatively small amount of data which might lead to unsatisfying results or to overfitting depending on the network architecture. We noticed that the networks are overfitting after several epochs when using such a small dataset and this raised the need for performing data augmentation. We used the same approach like [35], namely a thesaurus data augmentation. Similar to image data augmentation, thesaurus data augmentation tries to change the form but not the underlying features of textual sentences. This is achieved by replacing words that have the same meaning (synonyms). Therefore, the underlying feature, namely the semantics, is not changed. The augmentation pipeline is as follows:

1. Collect all posts that at least have one reaction except 'likes'.
2. Annotate each word within each post with a Part-of-Speech (POS) tag and split sentences using the CoreNLP Server.
3. Use NLTK [38] with WordNet [39] to request the closest synonym of each noun, adjective and verb.
4. Build new posts by replacing original words with their closest synonyms.
5. Finally, save the new posts with the same reactions as the original post in the database.

The data augmentation process increased the number of relevant posts from 8,103 to 486,471, which of course is a massive increase of the volume. We are aware of the fact, that there might be some error cases (due to e.g. language properties or when the closest synonym does not make sense). In Sect. 5, we evaluate the difference between training with original data and with augmented data and how it affects the results.

4 Reaction Distribution Prediction System Pipeline

In this Section, the complete prediction system is described. There are three core components: emotion mining applied to Facebook comments, artificial neural networks that predict the distribution of the reactions for a Facebook post and a combination of the two in the final prediction of the distribution of reactions.

4.1 Emotion Mining

The overall pipeline of the emotion miner can be found in Fig. 5. The first step is to split posts into sentences and pre-process the sentences by using CoreNLP. The processed sentences are then tokenized to be fit into a word-based emotion classifier, that allows one to annotate an emotion set to each word and therefore, to each sentence. However, the word-based classifier will not be able to annotate all words. The last step is a Support Vector Machine that predicts the emotions of the left non-annotated sentences. In the end, we are annotated the complete data set by processing through the emotion pipeline.

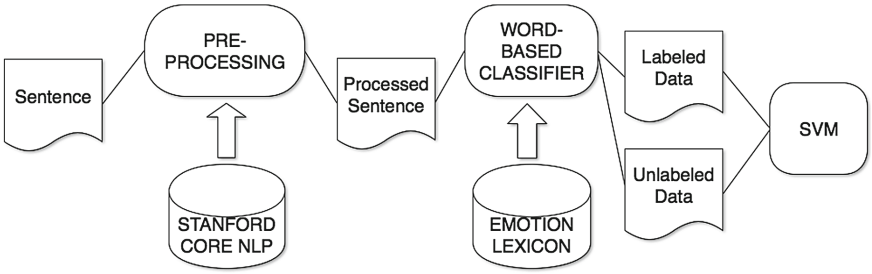

Fig. 5. Emotion miner pipeline [5].

The emotion lexicon that we utilize is created by [16] and is called NRC Emotion Lexicon (EmoLex). This lexicon consists of 14,181 words with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) associated with each word in the lexicon. It is possible that a single word is associated with more than one emotion. An example can be seen in Table 1. Annotations were manually performed by crowd-sourcing.

Table 1. Examples from EmoLex showing the emotion association to the words 'abuse' and 'shopping' [5].

	Anger	Anticipation	Disgust	Fear	Joy	Sadness	Surprise	Trust
abuse	**1**	0	**1**	**1**	0	**1**	0	0
shopping	0	**1**	0	0	**1**	0	**1**	**1**

Inspired by the approach of [17], EmoLex is extended by using WordNet [39]: for every synonym found, new entries are introduced in EmoLex having the same emotion vector as the original words. By applying this technique the original database has increased in size from 14,181 to 31,485 words that are related to an emotion vector. The lexicon can then be used to determine the emotion of the comments to a Facebook post. For each sentence in a comment, the emotion is determined by looking up all words in the emotion database and the found emotion vectors are added to the sentence emotion vector. By merging and normalizing all emotion vectors, the final emotion distribution for a particular Facebook post, based on the equivalent comments, can be computed. However, this naive approach yielded poor results, thus several enhancements were considered, implemented and described in Subsect. 4.1.

Negation Handling. The first technique that was used to improve the quality of the mined emotions is negation handling. By detecting negations in a sentence, the ability to 'turn' this sentiment or emotion is provided. In this paper, only basic negation handling is applied since the majority of the dataset contains only small sentences and this is sufficient for our goal. The following list of negations and pre- and suffixes are used for detection (based on work of [40]) (Table 2):

Table 2. Negation patterns [5] that we use for the negation handling. Those are standard negations, prefixes and suffixes used in the normal life.

Negations	no, not, rather, wont, never, none, nobody, nothing, neither, nor, nowhere, cannot, without, n't
Prefixes	a, de, dis, il, im, in, ir, mis, non, un
Suffixes	less

The following two rules are applied:

1. The first rule is used when a negation word is instantly followed by an emotion-word (which is present in our emotion database).
2. The second rule tries to handle adverbs and past particle verbs (POS tags: RB, VBN). If a negation word is followed by one or more of these POS-tags and a following emotion-word, the emotion-word's value will be negated. For example this rule would apply to 'not very happy'.

There are two ways to obtain the emotions of a negated word:

1. Look up all combinations of negation pre- and suffixes together with the word in our emotion lexicon.
2. If there is no match in the lexicon a manually created mapping is used between the emotions and their negations. This mapping is shown in Table 3.

Table 3. Mapping between emotion and negated emotions [5].

	Anger	Anticipation	Disgust	Fear	Joy	Sadness	Surprise	Trust
Anger	0	0	0	0	**1**	0	0	0
Anticipation	0	0	0	0	**1**	0	**1**	0
Disgust	0	0	0	0	**1**	0	0	**1**
Fear	0	0	0	0	**1**	0	0	**1**
Joy	**1**	0	**1**	**1**	0	**1**	0	0
Sadness	0	0	0	**1**	0	0	0	0
Surprise	0	**1**	0	0	0	0	0	**1**
Trust	0	0	**1**	0	0	0	**1**	0

Sentence Similarity Measures. [17]'s approach is using word vectors [18] in order to calculate similarities between sentences and further annotate sentences. In the context of this paper, a more recent approach was attempted [41], together with an averaging word vector approach for comparison. [41] creates a

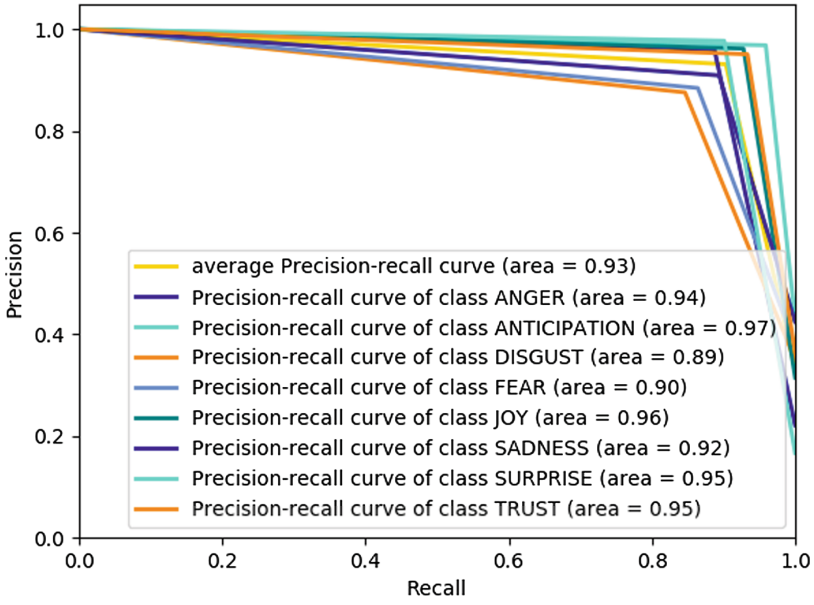

Fig. 6. Precision-Recall (ROC) curve using a linear SVM in an one-versus-all classifier [5]. The average precision recall has an area of about 0.93. The best precision-recall is achieved by the anticipation class as it has the most significant samples in the training set. The worst score is surprisingly reached by the disgust class. One would think that the disgust class has striking word representations and patterns.

representation for a whole sentence instead of only for one word as word2vec. The average word vector approach is summing up the word vector of each word and then taking the mean of this sum. To find a similarity between two sentences, one then uses the cosine similarity. Surprisingly, both approaches return comparable similarity scores. One main problem which occurred here is that two sentences with different emotions but with the same structure are measured as 'similar'. This problem is exemplified with an example:

```
Sentence 1: "I really love your car."
Sentence 2: "I really hate your car."
Sentence2Vec similarity: 0.9278
Avg vector similarity: 0.9269
```

This high similarity is problematic since the emotions of the two sentences are completely different. Also, one can see that the two models output almost the same result and that there is no advantage by using the approach of [41] over the simple average word vector approach. Hence, the sentence similarity measure method to annotate more sentences is not suited for this emotion mining task because one would annotate positive emotions to a negative sentence. Therefore, sentence similarity measurement is not used for our pipeline.

Classification of Not Annotated Sentences. If after executing these enhancement steps any non-emotion-annotated sentences remain, then a Support Vector Machine (SVM) is used to estimate the emotions of these sentences based on the existing annotations. The SVM is trained as a one-versus-all classifier with a linear kernel (8 models are trained, one for each emotion of EmoLex) and the TF-IDF model [42] is used for providing the input features. The input consists of a single sentence as data (transformed using the TF-IDF model) and an array of 8 values representing the emotions as a label. With a training/test-split of 80%/20%, the average precision-recall is about 0.93. Full results of the SVM training can be seen in Fig. 6 together with the precision-recall curve for all emotions. The result in this case was judged to be sufficient in order to utilize it for the next step, which is the reaction prediction and is used as presented here.

4.2 Reaction Distribution Predictor

In order to predict the distribution of the post reactions, neural networks are built and trained using TensorFlow [43]. Two networks were tested, based on literature research: a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) that uses LSTMs.

Both networks start with a word embedding layer. Since the analyzed posts were written in English, the GloVe [44] pre-trained embeddings (with 50 as a vector dimension) were used. Moreover, posts are short texts and informal language is expected, thus we opted for using embeddings previously trained on Twitter data instead of the Wikipedia versions.

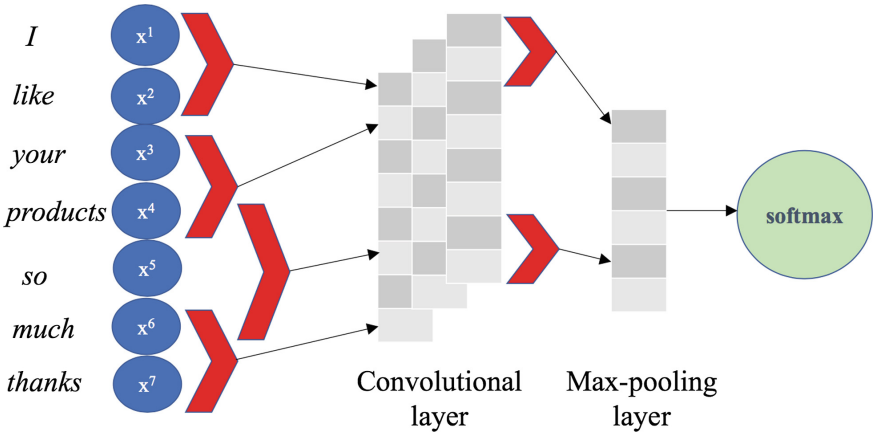

Fig. 7. A convolutional network architecture example [5]. The text input is vectorized with a pre-trained word embedding and then fed into the network. The convolutions extract meaningful features and the softmax activation forces the output to be a distribution output. (Color figure online)

CNN. The CNN model is based on existing successful architectures (see [7]) but is adapted to give a distribution of reactions as an output. An overview of the used architecture is provided in Fig. 7.

First issue to be handled with CNNs is that since they deal with variable length input sentences, padding is needed so as to ensure that all posts have the same length. In our case, we padded all posts to the maximum post length which also allows efficient batching of the data. In the example of Fig. 7 the length of the sentence is seven and each word x_i is represented by the equivalent word vector (of dimension 50).

The convolutional layer is the core building block of a CNN. Common patterns in the training data are extracted by applying the convolution operation which in our case is limited into 1 dimension: we adjust the height of the filter, i.e. the number of adjacent rows (words) that are considered together (see also red arrows in Fig. 7). These patterns are then fed to a pooling layer. The primary role of the pooling layer is to reduce the spatial dimensions of the learned representations (that's why this layer is also known to perform down sampling). This is beneficial, since it controls for over-fitting but also allows for faster computations. Finally, the output of the pooling layer is fed to a fully-connected layer (with dropout) which has a softmax as output and each node corresponds to each predicted reaction (thus we have six nodes initially). However, due to discarding *like* reaction later on in the research stage, the effective number of output nodes was decreased to five (see Experiments). The softmax classifier computes a probability distribution over all possible reactions, thus provides a probabilistic and intuitive interpretation.

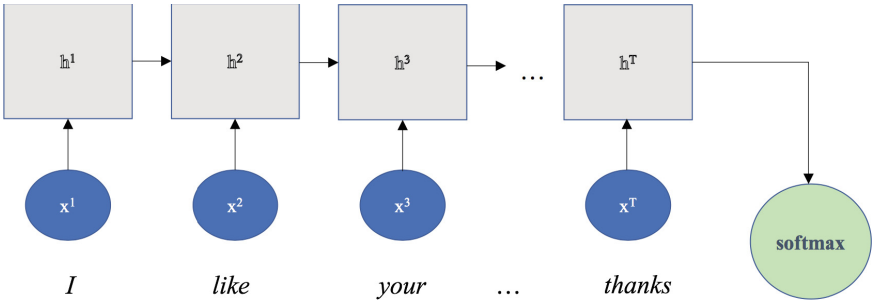

Fig. 8. Recurrent network architecture example [5]. As in the CNN example, the input text is converted to word embeddings. The words are then inserted step per step until the last layer that outputs the distribution with a softmax function.

RNN. Long short-term memory networks (LSTM) were proposed by [8] in order to address the issue of learning long-term dependencies. The LSTM maintains a separate memory cell inside it that updates and exposes its content only when deemed necessary, thus making it possible to capture content as needed. The implementation used here is inspired by [45] and an overview is provided in Fig. 8.

An LSTM unit (at each time step t) is defined as a collection of vectors: the input gate (i_t), the forget gate (f_t), the output gate (o_t), a memory cell (c_t) and a hidden state (h_t). Input is provided sequentially in terms of word vectors (x_t) and for each time step t the previous time step information is used as input. Intuitively, the forget gate controls the amount of which each unit of the memory cell is replaced by new info, the input gate controls how much each unit is updated, and the output gate controls the exposure of the internal memory state.

In our case, the RNN model utilizes one recurrent layer (which has 50 LSTM cells) and the rest of the parameters are chosen based on current default working architectures. The output then comes from a weighted fully connected 5-class softmax layer. Figure 8 explains the idea of recurrent architecture based on an input sequence of words.

4.3 Prediction Ensemble

The final reaction ratio prediction is carried out by a combination of the neural networks and the mined emotions on the post/comments. For a given post, both networks provide an estimation of the distributions, which are then averaged and normalized. Next, emotions from the post and the comments are extracted following the process described in Sect. 4.1. The ratio of estimations and emotions are combined into a single vector which is then computed through a simple regression model, which re-estimates the predicted reaction ratios. The whole pipeline combining the emotion miner and the neural networks can be seen in Fig. 9 and experimental results are presented in the next Section.

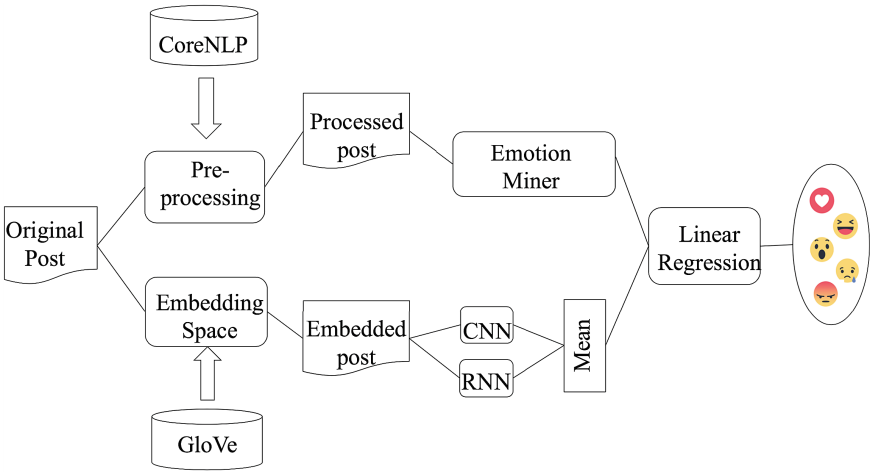

Fig. 9. Complete pipeline for the final prediction of the reaction distributions [5].

5 Experiments

Several experiments were conducted in order to assess different effects on the reaction distribution prediction. Firstly, the effect of pre-processing on posts is examined in Subsect. 5.1. Since Facebook reactions were not introduced too long ago, a lot of posts in the dataset still contain primarily *like* reactions. This might lead to uninteresting results as described in the Dataset Section and in Subsect. 5.2. Finally, Subsect. 5.3 discusses the training with respect to the mean squared error (MSE) for CNN and RNN models, as well as the effect of the combined approach.

As mentioned before, both networks utilized the GloVe pre-trained embeddings (with size 50). Batch size was set to 16 for the CNN and 100 for the RNN/LSTM.

CNN used 40 filters for the convolution (with varying height sizes from 3 to 5), stride was set to 1 and padding to the maximum post length was used. Rectified Linear Unit (ReLU) [46] activation function was used.

Learning rate was set to 0.001 and dropout was applied to both networks and performance was measured by the cross entropy loss with scores and labels with L2-regularization [47]. Mean Squared Error (MSE) is used in order to assess successful classifications (which effectively means that every squared error will be a 0) and in the end MSE is just the misclassification rate of predictions.

5.1 Raw Vs Pre-processed Input

In order to assess the effect of pre-processing on the quality of the trained models, two versions for each neural network were trained. One instance was trained without pre-processing the dataset and the other instance was trained with the pre-processed dataset. Results are cross-validated and here the average values are reported. Figure 10 indicates that overall the error was decreasing or being close to equal (which is applicable for both CNN and RNN). The x-axis represents the minimum number of 'non-like' reactions in order to be included in the dataset. It should be noted that these models were trained on the basis of having 6 outputs (one for each reaction), thus the result might be affected by the skewed distribution over many 'like' reactions. This is the reason that the pre-processed version of CNN performs very well for posts with 5 minimum reactions and very bad for posts with 10 minimum reactions In addition, the variance for the different cross-validation results was high. In the next subsection we explore what happens after the removal of 'like' reactions.

5.2 Exclusion of Like Reactions

Early results showed that including the original *like* reaction in the models would lead to meaningless results. The huge imbalanced dataset led to predicting a 100% ratio for the *like* reaction. In order to tackle this issue, the *like* reactions are not fed into the models during the training phase (moreover the *love* reaction can be used for equivalent purposes, since they express similar emotions). Figure 11

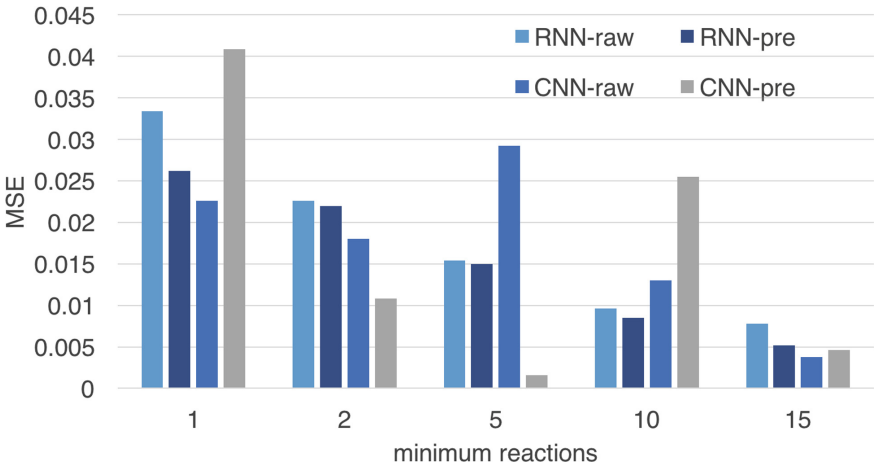

Fig. 10. Effect of pre-processing on different models [5].

shows an increase of the error when the likes are ignored. The explanation for this increase is related to heavily unbalanced distribution of *like* reactions: Although there is an increase in the error, predictions now are more meaningful than always predicting a like ratio close to 100%. After all, it is the relative reaction distribution that we are interested in predicting.

5.3 Ensemble Performance

Table 4 summarizes the testing error for the CNN and RNN with respect to the same split dataset and by also taking the validation error into account. One can see that RNN performs better than CNN, although it requires additional training time. Results are cross-validated on 10 different runs and variances are presented in the Table as well.

Table 4. RNN and CNN comparison after cross-validation [5].

	MSE	# Epochs
CNN	0.186 (+- 0.023)	81
RNN	0.159 (+- 0.017)	111

Combined results for either of the networks and the emotion miner can be seen in Fig. 12. The networks themselves have the worst results but an average combination of both is able to achieve a better result. Optimal result is achieved by the emotions + cnn combination, although this difference is not significantly better than other combinations. These results can be boosted by optimizing the

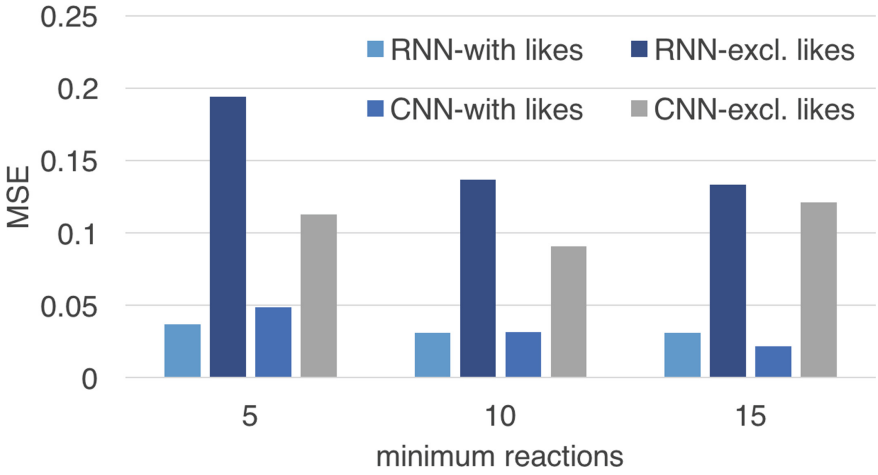

Fig. 11. Effect of inclusion/exclusion of likes on different models [5].

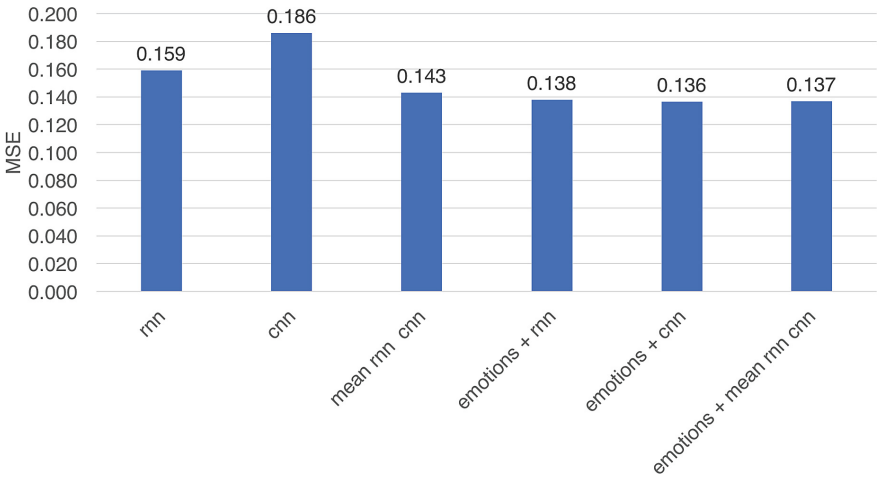

Fig. 12. Performance results for different combinations of the neural networks and emotions [5].

hyperparameters of the networks and also by varying different amount of posts. As a conclusion one can say that using emotions to combine them with neural network output improves the results of prediction.

5.4 Initial Data Vs Augmented Data

In this section we compare the initial dataset with the augmented dataset. The augmentation method is described in Sect. 3.2.

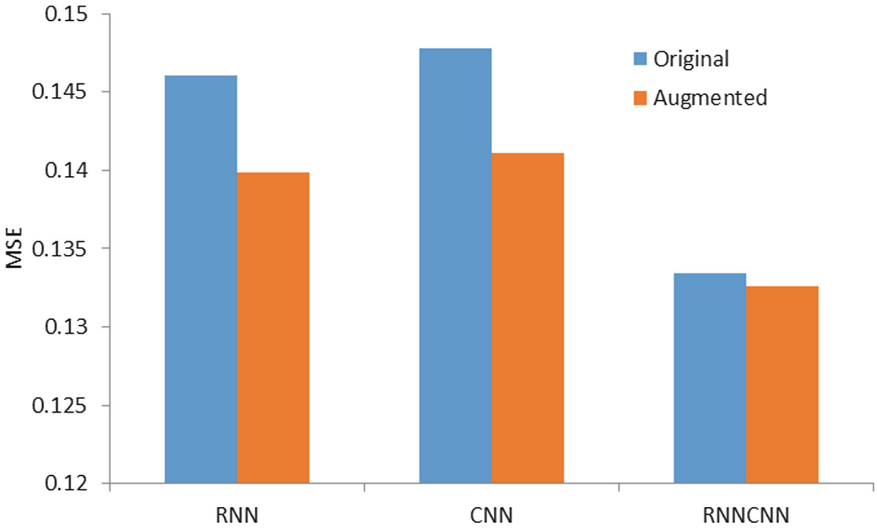

Fig. 13. Mean squared error of the three different models with and without augmented data.

Figure 13 shows that using augmented data improves the result of all three networks. But even the CNN, on which the data augmentation had the highest impact, could decrease its error by only 4.5% while the RNN changed by 4.2%. The combined network, which was the best performing model already, barely changed at all (0.6%). Even though the data augmentation improved our results the changes were not as significant as we hoped them to be. The reason could be that even though we achieved much more training data most of the posts still have very few reactions and hence are weak against noise. Those noisy posts are even multiplied due to data augmentation and leads to a higher error. Another reason might be that each original post has been augmented multiple times. In each of these copies we replaced a single word with the best matching synonym and copied reactions and emotions of the original post. This leads to a high number of similar posts that each have the exact same label. This could be reduced by adding some small Gaussian noise to the labels of each augmented copy. Since using both, the neural network's predictions and mined emotions, proofed to be a successful combination earlier (see Fig. 12) we also tried to do that but it seems that the emotion miner does not work well on the augmented dataset. The results were much worse and especially the mean squared error of the RNN increased drastically as shown in Fig. 14.

5.5 Visualization

Finally, we present a simple, yet effective visualization environment which highlights the results of the current paper, that can be found in Fig. 15. In this figure, one can see at the input field of the Facebook post on the top and then four

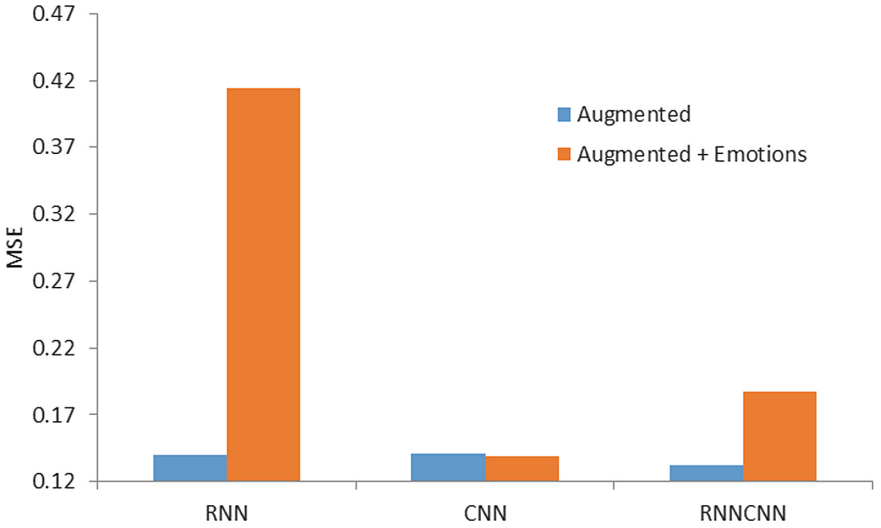

Fig. 14. Mean squared error of the three different models using the augmented dataset and the mined emotions.

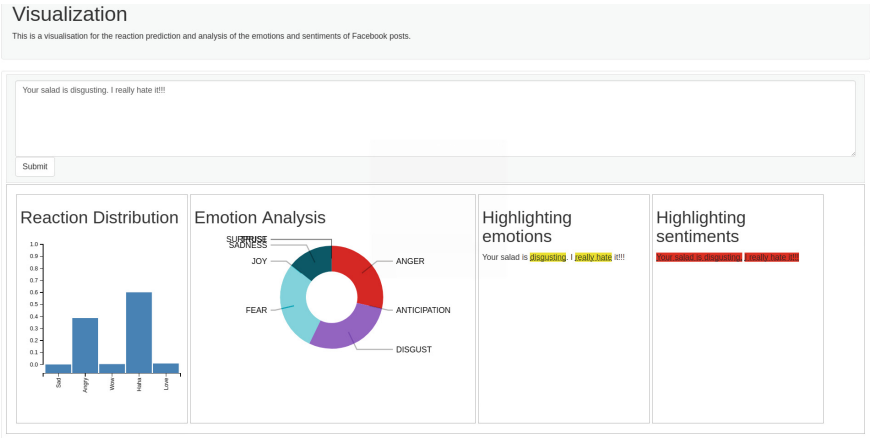

Fig. 15. Example visualization containing all components of SEMTec [5]. The user is able to write a post and to see the predicted reactions, an emotions pie diagram and an emotion and sentiment word highlighting of the input.

different result panels: the first one shows the reaction distribution, the second panel shows the proportions of the eight emotions, the third panel highlights the emotions (and by hovering, one can see the total shows the overall distribution (vector of eight)) and the fourth panel shows the highlighting of the sentiments. The visualization enabled us to analyze the network in a more interactive way and to point out where it fails and where it works. Also, it gives a good overview of the different components that this paper uses.

6 Conclusion

In this paper we presented and shared a dataset containing Facebook posts (with their reactions and comments) taken from public pages of several big supermarket chains. A framework for predicting the post reaction distribution was described and the effect of using the initial (small) dataset with an augmented (large) dataset was presented. We were able to show that data augmentation improved the quality of our neural networks by reducing the overall error for all three models. Our experiments demonstrate that combining a traditional emotion/sentiment mining technique with the output of modern neural network architectures can effectively predict the Facebook reactions. Furthermore, since most research works focus on sentiment analysis, our paper also contributes towards putting emotion analysis to the spotlight (especially in a social media context). Finally, the scrapped dataset developed during this work is available for other researchers and can be also used as a baseline for performing further experiments. It is in our future goals to further curate the dataset and more accurately evaluate the emotion mining part by using the MPQA corpus [48].

Despite recent distrust towards social media, it is clear that a transparent system which utilizes Facebook reaction predictions can enhance customer experience analytics. Identifying the emotion/reaction prediction of a post in almost real-time can be used to provide effective and useful feedback to customers and improve their experience. As long as page owners provide accurate recommendations and answers to complaints that are transparent and clear to the users, such a machine learning system remains extremely useful.

For future work, we plan to incorporate information that is currently not used in the dataset. That includes the reaction of the page owner and could contain useful information on how the post was addressed (or could be addressed). Another direction would be to combine the images that users attach to their posts with the actual text content. This synergy can reveal more about the sentiment/emotion of the user and can potentially highlight new directions in the language/vision domain. Since we are dealing with the social media domain, using external parameters (e.g. popularity of the post/poster, inclusion of other media external links, etc.), would also be a promising direction.

Finally, once there are enough posts (with potential comments) and resolutions from the page owner, we could build an automated reply system that based on the content of the post and the subsequent emotion/sentiment would be able to provide feedback to the customers in a minimal amount of time with minimal human intervention.

References

1. Ortigosa, A., Martín, J.M., Carro, R.M.: Sentiment analysis in Facebook and its application to e-learning. Comput. Hum. Behav. **31**, 527–541 (2014)
2. Feldman, R.: Techniques and applications for sentiment analysis. Commun. ACM **56**, 82–89 (2013)

3. Troussas, C., Virvou, M., Espinosa, K.J., Llaguno, K., Caro, J.: Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning. In: 2013 Fourth International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1–6. IEEE (2013)
4. Fan, W., Gordon, M.D.: The power of social media analytics. Commun. ACM **57**, 74–81 (2014)
5. Krebs, F., Lubascher, B., Moers, T., Schaap, P., Spanakis, G.: Social emotion mining techniques for Facebook posts reaction prediction. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence, pp. 211–220 (2018)
6. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
7. Kim, Y.: Convolutional neural networks for sentence classification. CoRR abs/1408.5882 (2014)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
9. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
10. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertainty Fuzziness Knowl.-Based Syst. **6**, 107–116 (1998)
11. Wang, G., Sun, J., Ma, J., Xu, K., Gu, J.: Sentiment classification: the contribution of ensemble learning. Decis. Support Syst. **57**, 77–93 (2014)
12. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in Twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1031–1040. ACM (2011)
13. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: the good the bad and the omg!. Icwsm **11**, 164 (2011)
14. Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of Twitter. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012. LNCS, vol. 7649, pp. 508–524. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35176-1_32
15. Sarlan, A., Nadam, C., Basri, S.: Twitter sentiment analysis. In: 2014 International Conference on Information Technology and Multimedia (ICIMU), pp. 212–216. IEEE (2014)
16. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon. Comput. Intell. **29**, 436–465 (2013)
17. Canales, L., Strapparava, C., Boldrini, E., Martínez-Barco, P.: Exploiting a bootstrapping approach for automatic annotation of emotions in texts. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 726–734 (2016)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
19. Tian, Y., Galery, T., Dulcinati, G., Molimpakis, E., Sun, C.: Facebook sentiment: Reactions and emojis. In: SocialNLP 2017, p. 11 (2017)
20. Pool, C., Nissim, M.: Distant supervision for emotion detection using Facebook reactions. arXiv preprint arXiv:1611.02988 (2016)
21. 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, vol. 10, pp. 79–86. Association for Computational Linguistics (2002)

22. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 513–520 (2011)
23. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
24. Yang, C., Lin, K.H.Y., Chen, H.H.: Emotion classification using web blog corpora. In: IEEE/WIC/ACM International Conference on Web Intelligence, pp. 275–278. IEEE (2007)
25. Wen, S., Wan, X.: Emotion classification in microblog texts using class sequential rules. In: AAAI, pp. 187–193 (2014)
26. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREc, vol. 10 (2010)
27. Yang, Z., Fang, X.: Online service quality dimensions and their relationships with satisfaction: a content analysis of customer reviews of securities brokerage services. Int. J. Serv. Indus. Manag. **15**, 302–326 (2004)
28. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
29. Cambria, E., Schuller, B., Xia, Y., Havasi, C.: New avenues in opinion mining and sentiment analysis. IEEE Intell. Syst. **28**, 15–21 (2013)
30. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning (2017)
31. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, C.: BAGAN: Data Augmentation with Balancing GAN. ArXiv e-prints (2018)
32. Antoniou, A., Storkey, A., Edwards, H.: Data Augmentation Generative Adversarial Networks. ArXiv e-prints (2017)
33. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Synthetic data augmentation using GAN for improved liver lesion classification. CoRR abs/1801.02385 (2018)
34. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification. ArXiv e-prints (2018)
35. Zhang, X., LeCun, Y.: Text understanding from scratch (2015) cite arxiv:1502.01710 Comment: This technical report is superseded by a paper entitled "Character-level Convolutional Networks for Text Classification", arXiv:1509.01626. It has considerably more experimental results and a rewritten introduction
36. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations, pp. 55–60 (2014)
37. Singh, T., Kumari, M.: Role of text pre-processing in Twitter sentiment analysis. Procedia Comput. Sci. **89**, 549–554 (2016)
38. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python, 1st edn. O'Reilly Media, Inc., Sebastopol (2009)
39. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
40. Farooq, U., Nongaillard, A., Ouzrout, Y., Qadir, M.A.: Negation handling in sentiment analysis at sentence level. In: International Conference on Information Management, London, United Kingdom (2016)
41. Arora, S., Yingyu Liang, T.M.: A simple but tough-to-beat baseline for sentence embeddings (2017)

42. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**, 513–523 (1988)
43. Abadi, M., et al.: Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
44. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
45. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
46. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011)
47. Masnadi-Shirazi, H., Vasconcelos, N.: On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In: Advances in Neural Information Processing Systems, pp. 1049–1056 (2009)
48. Deng, L., Wiebe, J.: MPQA 3.0: an entity/event-level sentiment corpus. In: Mihalcea, R., Chai, J.Y., Sarkar, A. (eds.) HLT-NAACL, The Association for Computational Linguistics, pp. 1323–1328 (2015)