Exploring Self Organizing Maps for Brand oriented Twitter Sentiment Analysis

Akshay Kapoor, Veni Jindal
Electrical & Computers Engineering
a43kapoor@uwaterloo.ca, v2jindal@uwaterloo.ca
University of Waterloo, Canada

***Abstract*—Nowadays, the social media presence of a business brand contributes to a brand's success enormously. In this paper we present an end to end tool to assess the brand's social media standing by analysing sentiments on tweets associated with the brand. Supervised and unsupervised language processing techniques have been used to understand the polarity of sentiments in these tweets. The paper focuses on use of Kohonen's self organizing maps to reduce dimensionality and to visualize the polarity of sentiments present in tweets. The performance of unsupervised KSOMs based models is compared with embedding based recurrent neural and transformer models like BERT etc. The experiments show promising results with KSOM's unsupervised approaches, with performance of BERT preprocessed RNN based supervised learning models resulting in best accuracy scores on test dataset.**

***Index Terms*—KSOM, sentiment analysis, word2vec, RNN, BERT.**

I. Introduction

In the present day world, the socio-economic growth of a company depends heavily on its presence on social media platforms. Leveraging these platforms help businesses to build their brand and promote the products/services. The social sentiment of customers associated with these brands can be measured by analyzing and classifying positive or negative polarity based on customer interactions with these businesses. With the development of natural language processing and various text processing models this task can be carried out using supervised and unsupervised text classification techniques. In addition to this, a comparison analysis of different competitor brands can be carried out and embedded with business intelligence tools.

Twitter is considered as one of the most effective social media platforms, where people convey views and opinions about a particular topic or a product. People engage in communicative transactions with brands via tweets and express how they feel about issues/products related to the brands. Hence, Twitter is chosen as the social media platform for carrying out social media sentiment analysis for a given brand. The positive or negative polarity of texts in tweets for a given brand reflects a high-level sentiment associated with them and can be very useful for understanding how customers perceive the products/initiatives made by these brands. Moreover, visualization of polarity in sentiments can also provide a comparative view that can complement sentiment labels to describe the degree of polarity (positive or negative) associated with a tweet.

Various traditional machine learning techniques such as naive bayes, logistic regression, and SVM have been used in the past to classify texts into corresponding positive and negative classes. Moreover the preprocessing for texts has mostly been done by handpicking features such as Part of Speech (POS) tagging and Term Frequency Inverse Document Frequency(TF-IDF) techniques. As the text content of tweets features very short length with diverse topics using above mentioned traditional techniques becomes very challenging. Artificial Neural Network (ANN) based models have shown promising results in the past for text classification cases. The supervised models such as recurrent neural networks (RNN) with Long Short Term Memory (LSTMs) and one-dimensional convolutional networks have emerged as go-to options for sentiment classification tasks. With the use of Word2vec, doc2vec and BERT based approaches to get the text feature representations, the ability of neural networks to classify text into relevant classes has been improved enormously. Despite high-performance accuracy for sentiment tasks, supervised machine learning techniques still face issues of high computational costs and lack of interpretability. Unsupervised ANNs such as Kohonen's Self Organizing (KSOM) as introduced in [1], can be used to tackle these issues faced by supervised ANNs. In this paper, we experiment with KSOM models with deep-learning bases text preprocessing methods such as Bidirectional Encoder Represenatations from Transformers (BERT), word2vec, doc2vec, etc. The performance of KSOM models with various text preprocessing techniques is visualized and compared with corresponding RNN based supervised implementations. The best performing model from the above-described methods is then used for prediction on recent tweets for a given brand. The predictions obtained are processed with interactive dashboards to help brands with strategizing the business decisions.

The organization of this paper is as follows: Section I gives a brief introduction of the work, section II explains related work and covers reviews of literature that describes the techniques used in the past to solve the related problems. Section III describes the methodology followed and the various experiments carried out using deep-learning-based architectures. Section IV shows the results and discussion of the experiments performed in section III. Section V summaries the work by presenting concluding remarks on the work.

II. Related Work

A great deal of work has been done on sentiment analysis and opinion mining. Most of the work done in twitter sentiment analysis focuses on the supervised techniques to predict the sentiment/emotion expressed in the tweet by the consumer. Techniques like Naive Bayes, Decision Tree, Support Vector Machine, and Artificial Neural Networks (ANN) utilizing recurrent networks like LSTMs (Long Short Term Memory units) have been suggested by much past research works. The main challenge of dealing with the unstructured text is to convert the documents or words to numerical embeddings which computers can understand and work with. Many different techniques have been proposed to get the document embeddings like TF-IDF, word2vec, paragraph to vector (Doc2vec), and another state of the art solutions like BERT to get the sentence encoding which will be discussed in this section. Apart from the supervised techniques, unsupervised ANNs like Kohonen's Self Organising maps have been proposed by several researchers for text classification with the benefits of better visualization, dimensionality reduction, and interpretability.

Work on text-based sentiment classification had been carried out by many researchers with interesting NLP based techniques. In paper [2], the authors design a model for real-time sentiment analysis of tweets where the tweets are extracted using Twitter API and then build the visualizations based on the designed model using histograms and pie-charts. The tweets which are streaming live were being collected, then preprocessed involving the tokenization, stop-word removal, stemming (getting the root word), and removing hashtags & URLs. The bag of words (BoW) model was used for classification of the tweets based on the two files in which positive and negative words were stored respectively.

Various feature extraction techniques as introduced in [3] discuss the issue of lack of features for classification due to the short text of tweets. The deep neural networks were used to extract the semantics of the texts. The tweets were first converted into word vectors using word2vec skip-gram model. After the conversion of words into vectors, the word sequence was then fed to LSTMs for the classification tasks. The deep learning methods showed better results given enough data as they can understand the context better than the bag to words model.

Another interesting development was proposed by Le and Mikolov in paper [4] as doc2vec, which constructs a paragraph vector to represent the text of variable lengths. This vector representation was proposed to overcome the major two weaknesses of the bag of words model which includes losing the order of words in the text as well as ignoring semantics of words. It is based on an unsupervised algorithm that tries to predict the word in the documents. The paragraph vector framework concatenates the paragraph and word vectors to predict the next word in the sequence instead of just word vectors as in the case of word2vec. The paragraph vector can be considered as a memory that provides the missing context. The paragraph vector is shared for the contexts of the same paragraph but not across other paragraphs, however, the word vectors are shared across paragraphs. It was considered a state of the art solution at that time for text classification and sentiment analysis. The embeddings created from doc2vec have been used to convert the tweets into vectors in our analysis which will be discussed in later sections.

The revolutionary ideas discussed by authors for [5], in which they fine-tuned pre-trained BERT (Bidirectional Encoder Representations from Transformers) for sentiment analysis which resulted to outperform other network architectures. BERT was proposed by Devlin et al. [6] which learns the bidirectional deep representations using the attention-based mechanism on transformer architecture. It has become a recent state of the art for NLP tasks as it allows a high degree of parallelism. Reimers and Gurevych proposed the Sentence BERT(SBERT) which uses the Siamese and triplet network on the top of pre-trained BERT to construct useful meaningful embeddings of the sentences. It helps to find the most similar sentences in a much more efficient way than the original BERT. SBERT achieved significant improvements in the field of designing sentence embeddings [7]. These techniques can be used to improve the performance of supervised and specifically of unsupervised ANN models for text classification purposes.

The unsupervised self-organizing maps (SOM) [1] architecture has been appreciated a lot for the task of document classification. SOM has the property of spatially arranging the inputs with similar internal feature representations together. This can prove to be very useful for finding the semantic similarity between documents. The paper [8] validates the use of Self Organising Maps in sentiment analysis. The work compares unsupervised SOMs with the supervised learning technique called Learning Vector Quantization (LVQ). The results obtained showed that unsupervised SOMs were well suited for the visualization as well as a sentiment classification task. Dieter Merkl [9] in his work discusses several ways of using SOM for classification tasks. This includes designing a hierarchical network built from several independent SOM to get a true scheme of classification. The major advantage of this structure was the reduction in training time and better visualization.

We used the aforementioned techniques to convert the tweets into fixed vector representations which were then fed to SOM as input vectors and designed visualizations for analyzing the sentiment polarity. We have also designed the LSTM based neural network architecture with the use of word2vec embeddings. Additionally, pre-trained BERT has also been fine-tuned to perform sentiment classification tasks yielding acceptable results The methodology to carry out end to end customer sentiment analysis with explorations of KSOM ANN models is explained in the section below.

III. METHODOLOGY

The methodology for the complete sentiment analysis consists of multiple steps. The flowchart for sentiment classification and analysis problem is shown in Fig 1. It shows the flow of steps that are being used to solve the task of sentiment analysis associated with brand tweets.

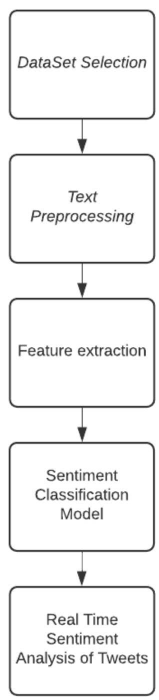

Fig. 1. Sentiment Analysis Methodology Flow Chart

A. Dataset Selection

The ANN-based data-driven models require large datasets to effectively learn the abstractions from the text data. Since our approach solves the sentiment classification problem by analyzing tweets associated with brands, a dataset featuring a large corpus of diverse tweets seemed like a good choice to select as a training dataset. Sentiment140 [10] is chosen as the dataset for which multiple variants of supervised and unsupervised models are trained. The dataset features 1.6 million tweets with balanced labels of positive and negative polarities. Since the size of the dataset is very large which in turn makes the training process computationally expensive, the experiments for ANN models have been performed on a sampled data dataset featuring only 50k tweets (25k with positive polarity and 25k with negative polarity). The model that performs best is used for making predictions and providing sentiment analysis on extracted tweets associated with brands. This sampled dataset is further split into training, validation, and test set in the ratio of 80:10:10 with samples of 40k, 5k, and 5k tweets respectively.

B. Text Preprocessing

The text data obtained from sampled tweets is cleaned before being fed to ANNs. These preprocessing steps are carried out to ensure that only relevant words from the text are used. The following types of preprocessing steps have been applied for text cleaning in the dataset.

1) Twitter specific cleaning: The HTML tags and twitter handle names are removed from texts as they do not add substantial meaning to the overall polarity of a tweet. Furthermore, hashtags and web-based links are also removed, as these links just act as a medium to take users to another page of the world wide web.

2) Stopwords removal: The frequently occurring words which do not have a significant contribution to the meaning of the sentence are removed as well. The list of stopwords is deliberately structured to not to remove any negation words as they can play a crucial role in determining the polarity of sentiment related to a given text.

3) Tokenization: Tokenization is a basic preprocessing step for any NLP task which involves splitting the sentences/ documents in the corpus to smaller units referred as tokens. Here, after removing the stopwords, hashtags, html tags from the tweets, they are then tokenized into words. The word sequence obtained is used to train the word2vec model.

C. Feature extraction/vector representation

For understanding the sentiment associated with tweets, a numerical high dimensional representation of text is required. Such representation of a given text can be done in two ways as described underneath.

1) Word based representation: The word-based representations such as TF-IDF [11] and word2vec [12, 13] are used to get the embeddings of a sentence in terms of its word representations. The procedure for these two approaches is explained below:

a) TF-IDF: It is a technique that can reflect how much a word is important in a document or corpus. The TF-IDF value is proportional to the frequency of the word in a document (term frequency) and offset by the number of times it appears across the document (inverse document frequency). The clean text tweets are transformed into the term frequency-inverse document frequency matrix to get the relevance of the words in a given tweet from dataset text corpus. The TF-IDF weight for the word ‘x’ in document ‘y’ can be calculated as follows:

$W_{x,y} = tf_{x,y} * log(N/df_x)$ where N is the total number of documents in the corpus, $tf_{x,y}$ is the frequency of word x in the corresponding document y and df_x is the number of documents in the corpus containing word x.

The generated features for a given tweet text are very high dimensional corresponding to the size of the vocabulary of the training data. Feature extraction technique; truncated singular value decomposition has been used to reduce the dimensionality of the formed tweet text feature sparse matrix. The experiments have been conducted by reducing the number of features to 100 and 300 with the results being shown in the later sections.

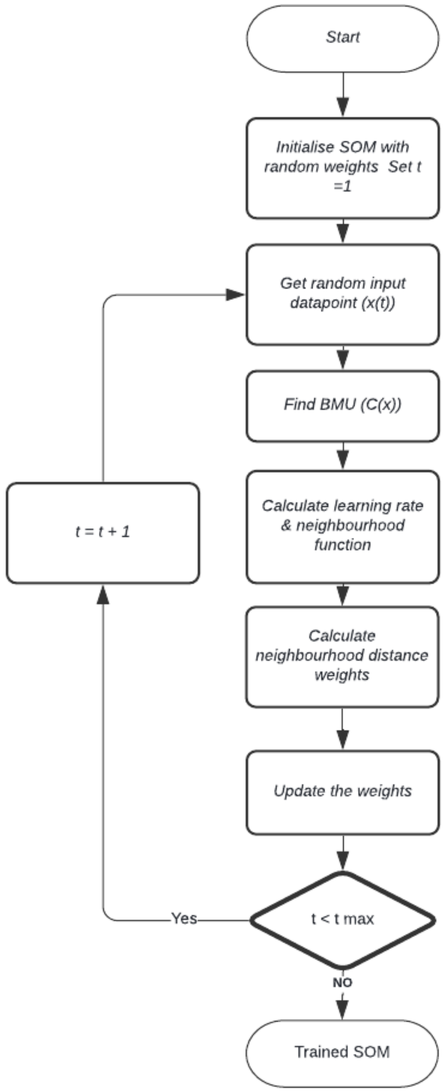

Fig. 2. Flowchart for Unsupervised SOM algorithm

b) Word2vec: Word2vec is another technique of vectorizing the words which take into account the context of the word, rather than just considering the frequency of the word as in TF-IDF. It is an unsupervised neural network-based algorithm trained on a large corpus. The basic idea is that the words which occur in similar contexts will have similar vectors in the embedding space. Every word in the word2vec representation has a fixed length embedding using an unsupervised approach which can be obtained by using either Continuous Bag of Words (CBOW) or Skip-gram model. The former tries to predict the target word given the context while the latter predicts the context based on the given word. Skip-gram model has been used in the work presented here to convert the words to vectors of size 100 and 200 respectively. The average of all the word vectors in the document is taken and considered as the document vector which is then fed to KSOM for dimensionality reduction. The results obtained from the embeddings of size 100 and 200 have been compared for the unsupervised KSOM approach in the results section. Furthermore, a word embedding of dimension 300 has been used for feeding the word sequence to the LSTM layer and CNN network in the supervised approach.

2) Sentence based representation: The word-based techniques described above provide the meaning of a sentence based on individual contributions from words. On the other hand sentence based representations render a higher-level meaning of the vector representation of a sentence. The sentence based representation used in this paper are explained below.

a) Doc2vec: The doc2vec model as introduced in 2014 [4] tries to generalize the word vector representation produced by word2vec. Here, we have used the Paragraph Vector: Distributed Memory (PVDM) model which is similar to the CBOW model of word2vec with an additional memory vector that tries to capture the topic of the paragraph. The 'doc2vec' algorithm using the 'gensim' library has been used to produce vectors from the tweets. The size of the document vectors produced from the processed tweet tweets using doc2vec is varied between 100 and 500 which are then fed to KSOM for clustering and dimensionality reduction.

b) Sentence BERT: Sentence BERT (SBERT) as described in [7] is used to transform text tweets into respective vector representations. We have used the sentence-transformers library in python to convert the cleaned tweet to a 768-dimensional vector. BERT pre-trained model 'bert-base-nli-mean-tokens' has been used to embed the sentences and save the embeddings corresponding to each tweet in a CSV file which is later used to provide input to the SOM model.

D. Sentiment Classification Model

Supervised and unsupervised ANN models are trained with vector representation of sentences obtained from techniques such as TF-IDF, word2vec, Doc2vec, and BERT. The focus of this paper is kept on unsupervised methods featuring KSOM as these methods provide good visualizations and interpretability. Moreover, these methods reduce the dimensionality of feature space by forming a two-dimensional feature map, and the position of the best matching unit on the feature map can also be used to form clusters or to train supervised models on the top. Supervised RNN based models have been trained to compare the performance with unsupervised models. Variants of implemented unsupervised models and supervised models are explained below:

1) Unsupervised ANN model: Kohonen's self-organizing maps [1] are used to classify the processed text- vector embeddings. SOM is an ANN technique which is based on competitive and cooperative learning methods. The training algorithm of KSOM models is shown in Fig. 2. A typical SOM architecture is made of input layer nodes, an array of nodes as an output map along with a matrix of connection between each input and output layer units. The input data is received at the input node and is propagated through the weight connection matrix to form a set of output nodes arranged in a map topology (the two-dimensional grid is used) also known as Kohonen's layer. The spatial location of an output node can be related to the particular feature of the input data pattern as described by Kohonen's principle of topographic map formation(Merkl, 1998) [9].

Each output node j related with weight vector w_j (having the same dimension as input data vector) and is associated with a position in the feature map. The mapping of an input data vector to a feature map is done by finding the spatially closest weight vector also known as the best matching unit (BMU). The selection of BMU is defined by equation (1) and the weights associated with the neighborhood of BMU are updated as per the equation defined by (2). The neighborhood function for cooperative learning is given by equation (3) where we can control the spread of the neighborhood function. The spread parameter for the neighborhood (σ) and learning rate (α) are exponentially decayed after each iteration as described by (4) and (5) respectively. These algorithm steps are explained in the flowchart shown in Fig 2.

$$C = \|x - w_c\| = min_{ij}\|x - w_{ij}\| \tag{1}$$

$$w_i(t+1) = w_i(t) + (t).N_{ij}(t).(x(t) - w_i(t)) \tag{2}$$

$$N_{ij}(t) = exp(-d_{ij}^2/2\sigma(t)) \tag{3}$$

$$\alpha(t) = \alpha(0).exp^{-c_1 t/T} \tag{4}$$

$$\sigma(t) = \sigma(0).exp^{-c_2 t/T} \tag{5}$$

The implementation of SOM ANN models is carried out using SimpSOM [14] libarary in python. Apart from unsupervised techniques like SOM, we have also implemented and compared several supervised algorithms where labels are being provided with the input vectors designed by several techniques described in the feature extraction section. Several supervised algorithms like CNNs, LSTMs, and state of the art solutions like BERT have been described in the section below.

2) Supervised ANN model: Supervised sentiment analysis techniques have been widely used to detect customer opinions about the brand. Detection of semantics from these short tweets requires some deeper neural network architecture. First, the words are converted into the embeddings using the 'word2vec' library from the 'gensim' package in python. The word2vec model is trained using a vector size of 300 with a context window of 7 for a total of 32 epochs. The text sequences are then converted to dense representations using the embedding layer initialized with the embeddings generated from the trained word2vec model. RNN based state of the art models featuring convolutional layers (One dimensional), Long short term memory units, and BERT based classification is also carried out for the task of sentiment classification. These models are described in the section below.

As the bag of words or TF-IDF representation of the document vectors is very high dimensional due to large vocabulary size and does not take into account the context, here we have used word2vec model to generate the word vector representations of 100, 200 & 300 dimensions and then fed these word vectors as a sequence to LSTM layer. The output is then used to classify the sentiment polarity in the tweet. The architectures involving the use of unidirectional LSTM as well as Bidirectional LSTMs (BiLSTM) have been used for predicting the sentiment. The Bi-directional extension of traditional LSTMs generally results in improvement of the performance of the classification model as it trains on two sequences instead of one including the reversed word sequence. This results in providing an additional context to our model. The typical model architecture used includes two Bidirectional LSTM layers followed by a dense layer. The output layer consists of a single neuron with a sigmoid activation function as there are only 2 classes i.e positive and negative. However, the RNNs are generally slower because the input string is processed sequentially. In contrast, CNN's can result in the improvement of speed as they can process all the elements simultaneously.

CNN's are very commonly used in computer vision with the filters sliding over the local patches of the images. Recently, these models are also been used in text classification which involves sliding the filters over every row of the matrix representing the documents. Each row in this matrix corresponds to a fixed dimensional word vector of the token in the sentence. The main feature of the filters used for the NLP task is that the width of the filter is the same as the width of the document matrix. A typical CNN architecture used for text classification involves 1-Dimensional convolutional filters of different window sizes applied to the input embedding layer. These layers are followed by the max-pooling layers to reduce the dimensionality of the receptive fields. This produces a final sentence vector which can be then used for classification purposes. The CNN model used for twitter sentiment analysis here consists of four one dimensional convolutional layers, each followed by the max-pooling layers and a dropout layer(to prevent overfitting to the training data). The output from the convolutional layers is then flattened and converted into a dense vector that is used for classification using the sigmoid activation function.

The major limitation of traditional NLP models like word2vec is that it does not perform well when the context is changed. The LSTM or CNN models built based on word2vec embeddings are not able to capture the meaning of the sentence. To overcome, these shortcomings of a deep neural network involving LSTMs, several language models like Embeddings from Language Models(ELMo) [15], Universal language model fine-tuning for text classification(ULMFiT) [16] have been proposed which integrate the use of BiLSTM but are not able to look at both the directions simultaneously. BERT on the other hand can learn from the position of words in the sentence and can capture the meaning of the sentence. It has been trained on a large corpus including Wikipedia articles and Google books in an unsupervised manner. Here, we have used BERT to predict the sentiment of the tweet in 'TensorFlow' using the 'TensorFlow hub' module. Firstly, the input text is formatted in the form that BERT understands. Next, we

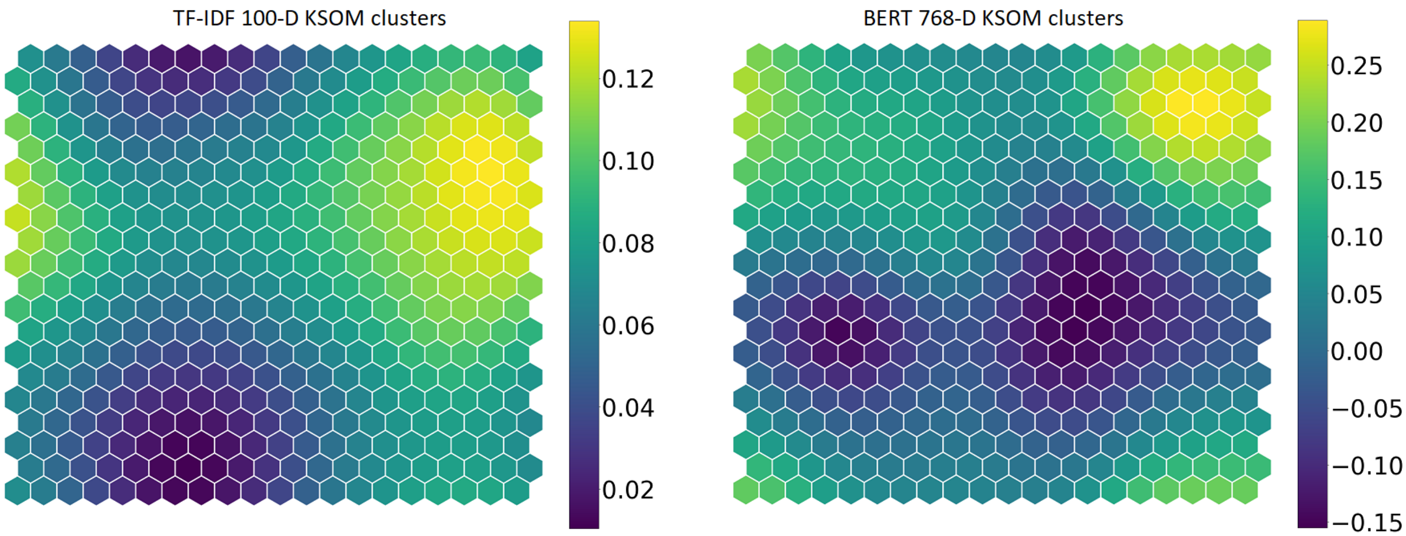

Fig. 3. Colormap obtained from TF-IDF based SOM (on the Left) and SBERT based SOM after 10k epochs(on the right)

created a BERT tokenizer which accepts the lowercase text to convert text to features. The model is fine-tuned by loading a pre-trained BERT model with the addition of a new layer that will be used for sentiment classification.

After designing all the above models described and training them on the training set, the model is then used to make predictions on the test set. The tweets in the test set are preprocessed in the same way as applied to the training set. The output from the aforementioned models is then evaluated with the labels of the test data in the experiments. The results obtained are discussed and compared in the next section.

E. Real Time Sentiment Analysis of Tweets

The best performing ANN model is used for obtaining customer sentiment insights from the recent tweets, the steps to do so are explained below.

1) Extracting recent Tweets: The recent tweets associated with a given brand and their competitor are extracted using ”Tweepy” python API. These tweet texts are then preprocessed similar to the data cleaning technique used for training ANN models.

2) Prediction on tweets: The predictions on the above tweets are made based on the best performing model. The probabilistic outputs are converted into crisp values by using 0.5 as a threshold value. Moreover, the probabilistic polarity score is also stored in a data frame to reflect the degree of customer sentiment reflected in a given tweet.

3) Dashboard Visualizations: To understand the proportion of customer sentiments associated with the recent social media status for a given company. The percentage of tweets identified as positive and negative are visualized using pie charts. Moreover, a categorical bar chart representation is also presented to visualize comparison of competitor brands. The visualization dashboard has been designed in Tableau and looks similar to the Fig. 5.

The next section shows the result obtained from training supervised and unsupervised method with multiple variants of prepossessing steps.

IV. Results and Discussion

In order to evaluate the performance of each approach and network architecture, we have conducted several experiments with varying dimensions of word/document vectors. Input features have been selected using different feature extraction techniques discussed in the previous section and compared against each other.

TABLE I
Unsupervised KSOM Results

S.No.	Model Description	Test Accuracy
1	KSOM with TF-IDF 100-D representation	59.42%
2	KSOM with TF-IDF 300-D representation	59.7%
3	KSOM with word2vec 100-D representation	57.36%
4	KSOM with word2vec 200-D representation	59.48%
5	KSOM with doc2vec 100-D representation	66.84%
6	KSOM with doc2vec 300-D representation	65.2%
7	KSOM with doc2vec 500-D representation	64.3%
8	KSOM with SBERT 768-D representation	71.02%

Table I shows the performance of SOM based models on the sampled dataset of 50k tweets. The size of the feature map is kept as 20x20 with initial learning rate value is kept fixed as 0.01 and initial value of spread parameter is kept as 100. The learning rate and spread parameter are exponentially decayed after each iteration as described by equations (4) and (5). From Table I, it can be observed that the performance of TF-IDF (test set accuracy of 59.7% with 300 dimensions) and word2vec (test set accuracy of 59.48% with 200 dimensions) based methods is inferior compared to the sentence based preprocessing applied using doc2vec and SBERT methods. This can be attributed to the fact that word based representations are focused on representing the sentence

TABLE II
SUPERVISED ALGORITHM RESULTS

S.No.	Model Description	Training Accuracy	Test Accuracy
1	LSTM Unidirectional model with 161k parameters	78.33%	75.2%
2	LSTM Based Unidirectional model with 27k parameters	76.72%	75.12%
3	LSTM Based Bidirectional model with 27.5k parameters	77%	75.46%
4	LSTM Based Bidirectional model with 55k parameters	78.78%	74.84%
5	One Dimensional Convolutional layer model with 264k parameters	77.3%	74.98%
6	One Dimensional Convolutional layer model with 114k parameters	75.75%	74.8%
7	One Dimensional Convolutional layer model with 208k parameters	78.02%	75.08%
8	BERT based model	98.8%	82.68%

with respect to individual words whereas the sentence based technique takes association of words into consideration to form a vector representation of the sentence. Moreover the performance of SBERT based KSOM model (test set accuracy of 71.02%) is better than doc2vec (test set accuracy of 66.84%) based model, this performance behavior reflects that SBERT is better at capturing semantic as well as syntactic relation of text documents. It can also be attributed to SBERT's dynamic ability to consider polysemy in text data.

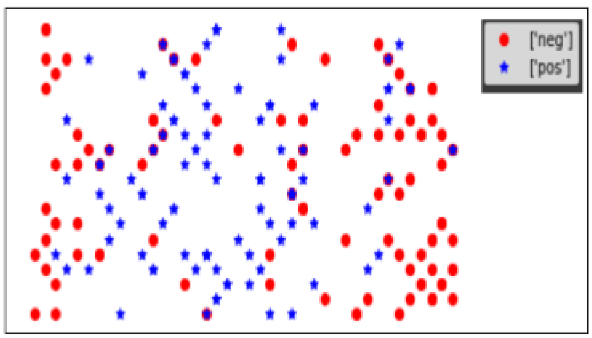

Fig. 4. KSOM with 768-D SBERT features

The two dimensional feature map obatined after training SOMs for 10000 epochs with TF-IDF (100 dimensional representation) and SBERT (768 dimesional representation) is shown in the Fig. 3. It can be observed from the fig .. that BERT based SOM (hexagon geometry) implementation forms more cohesive clusters (appearing in dark shades of blue colors and light green- yellow color shades) compared to TF-IDF based KSOM implementation. This highlights the importance of feature extraction for unsupervised ANN networks.

A typical representation of the two dimensional feature map obtained from the BERT based SOM mode with 768 dimensional vector representation on 200 sample points (100 samples featuring positive and 100 samples featuring negative sentiment) is shown in Fig. 4. The positive samples shown as blue stars and negative samples are shown as red dots. It can be observed that the majority of these positive and negative samples are forming respective clusters. The accuracy for the dataset having reduced dimensionality (here dimensionality is kept as 2) is been computed using traditional machine learning algorithm such as SVM, logistic regression, KNN and shallow neural networks The best performing accuracies are reported in Table I.

The performance of ANN based with 1024 batch size and 32 epochs on supervised models (model is trained till overfitting is observed) are shown in table II. It can be observed that performance on test set of unidirectional (best test set accuracy of 75.20%) and bidirectional LSTMs (best test set accuracy of 75.46%) is similar to one dimensional convolutional layer based network (best test set accuracy of 75.08%). However the observed training speed for CNN is faster than the LSTM network.The preprocessing for all supervised ANN models except the BERT based model is carried out by training 300 dimensional word2vec models.

Test set performance of BERT based supervised model is best among all supervised implemented ANNs with a test set accuracy 82.68%. Looking at the table II results it can be said that on a given text data, BERT preprocessing based model is robust in extracting meaningful features compared to word2vec based preprocessing with supervised CNN/LSTM based networks.

On comparing table I and table II observations, it can be said that the BERT based supervised and unsupervised model has performed best among the carried out ANN implementations. Moreover the performance of SBERT based unsupervised KSOM method (test set performance 71%) significantly lower than supervised BERT ANN model (test set performance of 82.68%).

In order to make predictions on real-time prediction, BERT based supervised model is trained on 50k tweets data(for better results the it can be trained on 1.6 million tweets). The training and testing accuracy of this model is achieved as 98.8% and 82.68% respectively. Recent tweets for a given brand and its competitor brands are extracted using 'tweepy' API in python. The sentiment classification on these tweets are used to create dashboards for summarizing customer sentiments.

Typical dashboard results are shown in Fig. 5. A pie chart for a given company e.g. Amazon is shown, along with the probabilistic size based bubble charts representing the degree of positive polarity associated with the tweets. In addition to this, a comparative illustration featuring bar chart comparison of competitive brands; Ebay, Alibaba and Shpoify has been shown. It can be observed that Ebay has more positive sentiments associated with it compared to Amazon, Alibaba and Shopify. Moreover the probabilistic range (0 being highly

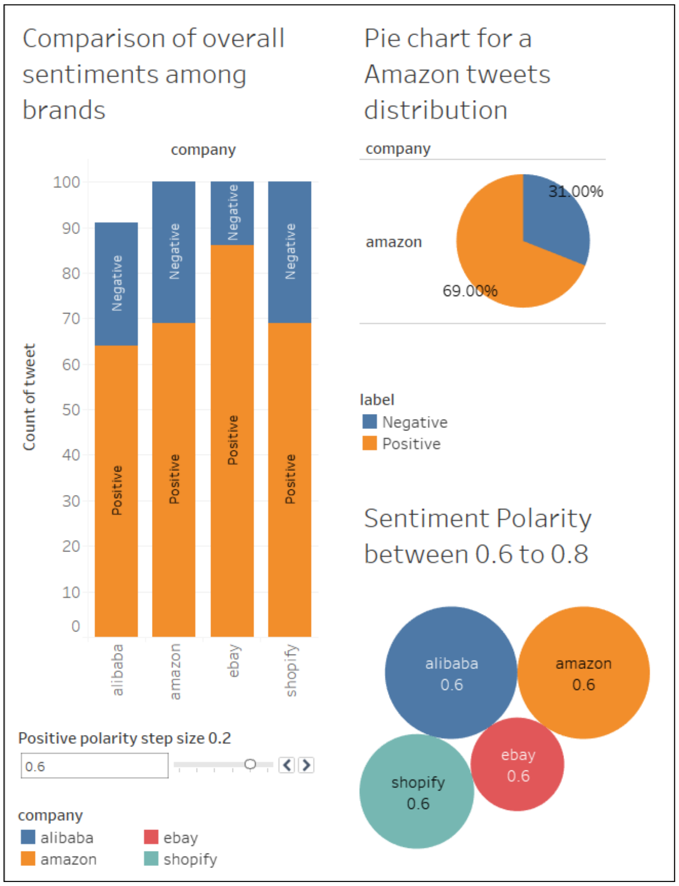

Fig. 5. Typical sentiment analysis dashboard for Amazon

negative sentiment and 1 being very positive sentiment) is distributed in bins of size 0.2 and has been used to compare the degree of customer sentiment polarity associated with each brand. The size of spherical bubble in the image represents the number of tweets associated with a given brand that lie in given positive polarity range (the example shows the range from 0.6-0.8). Analysis similar to this can be used to assist in the strategic planning to maintain a desired social media standing.

V. Conclusion

In this paper we built a customer sentiment analysis tool for business brands based on associated customer tweets. ANN based unsupervised models have been majorly explored for sentiment classification/clustering tasks. Various ANN based preprocessing steps have been experimented such as word2vec, doc2vec and several variants of BERT, and compared with traditional techniques such as TF-IDFs. SBERT based Kohohnen's self organizing maps have performed significantly better on sampled dataset with an accuracy of 71%. Moreover, the performance of both unsupervised as well as supervised models used on the top of BERT based preprocessing has been the best among all the experiments. Real time analysis of brand tweets has been performed applying BERT based supervised ANN model trained on sampled sentiment140 dataset. Dashboards for customer sentiments related to a given brand are presented with analytical graphs and charts to assist brands for making strategic decisions.

In the future, we would like to form clusters of similar competitive brands by extracting data from the web and use clustering algorithms to automatically find brand competitors. Moreover, we would like to train our model on larger corpus along with tuning hyperparameters for robust and efficient ANN based models. We would also like to work upon automating this pipeline to directly create analytical insights by integrating graphical media platforms directly with python code.

References

[1] T. Kohonen, “The Self-Organizing Map,” Proc. IEEE, 1990, doi: 10.1109/5.58325.
[2] Prakruthi V, Sindhu D, Dr S Anupama Kumar, “Real Time Sentiment Analysis Of Twitter Posts,” 3rd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions 2018.
[3] J. H. Wang, T. W. Liu, X. Luo, and L. Wang, “An LSTM approach to short text sentiment classification with word embeddings,” in Proceedings of the 30th Conference on Computational Linguistics and Speech Processing, ROCLING 2018, 2018.
[4] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” 2014.
[5] M. Munikar, S. Shakya, and A. Shrestha, “Fine-grained Sentiment Classification using BERT,” 2019, doi: 10.1109/AITB48515.2019.8947435.
[6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2019.
[7] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” in EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, 2020, pp. 3982–3992.
[8] A. Sharma and S. Dey, “Using self-organizing maps for sentiment analysis,” in Proceedings - KIM 2013, Knowledge and Information Management Conference: Sustainable Quality, 2013.
[9] D. Merkl, “Text classification with self-organizing maps: Some lessons learned,” Neurocomputing, 1998.
[10] http://help.sentiment140.com/for-students accessed on 10^{th} July 2020.
[11] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Inf. Process. Manag., 1988.
[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013.
[13] Mikolov, Tomas (2013). ”Distributed representations of words and phrases and their compositionality”. Advances in neural information processing systems.
[14] Federico Comitani, 2019. fcomitani/SimpSOM: v1.3.4. doi:10.5281/zenodo.2621560
[15] M. E. Peters et al., “Deep contextualized word representations,” in Proc. of NAACL, 2018.
[16] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 2018.