University of Guilan

Computational Sciences and Engineering

Journal homepage: https://cse.guilan.ac.ir/

Weighted Bi-directional GRU Capsule Ensemble Approach for Multi-Domain Sentiment Analysis

Vahid Mottaghi [a,*1] , Hamed Afshar Farnia [a]

[a] Department of Computer Engineering, Technical and Vocational University, (TVU), Tehran, Iran

ARTICLE INFO

Article history:
Received 13 March 2022
Received in revised form 26 May 2022
Accepted 26 May 2022
Available online 26 May 2022

Keywords:
Multi-domain sentiment analysis
Deep learning
Weighted neural network
Natural language processing
Ensemble method

ABSTRACT

With the advent of the Web today, users' opinions can be incorporated into a variety of applications. Automated methods have been developed to derive users' general sense from these textual comments, often known as sentiment analysis, and aim to determine the polarity of a text relative to a subject. One of the challenges is the inability to use one domain of data to analysis sentiment in another domain and the lack of sufficient labelled data in a particular domain. To address these challenges, multi-domain sentiment analysis systems have been developed. This paper propose Bi-GRU Capsule ensemble approaches for multi-domain sentiment classification to address the mentioned issues. Using a weighted score of Term-Frequency and Inverse Document Frequency degree and the initial polarity of the sample test data on each domain, a new aggregated score of final polarity is obtained. The DRANZIERA protocol is used for evaluation of the proposed model. The outcomes demonstrate the effectiveness of the proposed approach and also set a plausible starting point for future work.

1. Introduction

With the spread of the Internet and electronic commerce, many users, producers and service providers use their online stores for their business. One of the most prominent features of online stores is that users can be informed about the quality of the products and services by reading reviews of other consumers to make a better decision. This vast source of information is not only useful to the customers, but also helps the providers to increase the quality of their products or services based on customers' requirements. However, reading a large number of users' opinions (or more technically sentiments) is time-consuming, tedious, and in most cases impossible. Due to importance of analyzing this amount of data, several researchers have motivated to find automatic techniques to extract the sentiments from these textual repositories. Sentiment Analysis (SA), also

[*1] Corresponding author, Email: mvahid500@gmail.com

https://dx.doi.org/10.22124/cse.2022.21884.1027

known as opinion mining, review mining, and polarity classification [1], is a computational study of opinions which uses Natural Language Processing (NLP), computational techniques, and text analysis for extracting the polarity of unstructured documents or textual reviews [2]. The primary purpose of the sentiment analysis is to automatically detect the polarity of a review in terms of being positive or negative [3].

One of the most critical issues when working on multi-domain data ("domain" is a set of documents about a similar topic) is that a term in different domains may have different credentials. For example, consider two following sentences:

1. "The weather is cold."
2. "I like cold weather."

In the first sentence, the polarity of the "cold" is "negative", while in the second sentence the adjective "cold" is "positive”. Furthermore, in some domains, some words have "positive" or "negative" orientations that make no sense in another domain. For instance, "short battery life" indicates negative sense in the electronic domain, while does not convey any sense in the book domain [4]. This problem is known as domain dependency in the literature which is an inherent problem of the sentiment analysis [5]. Since one of the basic methods in sentiment analysis is using classifiers, one of the main reasons for reducing performance is lack of learning previously unseen sentiment word on particular domain when the classifier is trained on other domains [4].

This paper deals with the mentioned problem of multi-domain SA. The proposed Weighted Neural Network Ensemble (WNNE) approach includes the following general steps:

1. Embedding words of the raw reviews using existing pre-trained word embeddings,
2. Training a neural network (CNN, LSTM, or Bi-GRUCapsule) for each domain in the dataset.
3. Calculating the domain-belonging degree as a weighting criterion.
4. Combining the outputs of the networks and the domain-belonging degrees to reaching the final polarity.

The paper is structured as follows. In Section 2 a survey of machine learning and deep learning approaches in sentiment analysis and multi-domain SA has been presented. Section 3 describes the proposed WNNE method. The experiment results on the method are presented in Section 4. Section 5, covers the error analysis of the proposed approach as well as some points for improvements in the future. Finally, Section 6 concludes the article.

2. RELATED WORKS

In this section, we summarize the methods of SA, the categorization of methods, and the weaknesses and strengths of each of these methods. We will also briefly mention the multi-domain SA. At the end, we focus on deep learning techniques and their application in NLP and SA.

2.1. Sentiment Analysis and its techniques

The topic of SA has been studied in the literature[6, 7] and several different techniques have been suggested for it. Among the proposed approaches, machine learning based approaches are more common, which are divided into two main categories: Supervised and Unsupervised learning approaches.

- Supervised learning approaches: which are based on labelled dataset. The success of these models is heavily dependent on extracted features that are used to discover sentiments. For instance, Z.Zhang et al [8] used the Support Vector Machine(SVM) and Naive Bayes(NB) approaches to classify movie reviews. They used unigrams, bigrams, and trigrams as features to train their classifiers. B.Pang et al [9] used the SVM, NB, and Maximum Entropy(ME) model with unigram, bigram, and position of adjective to words to classify the movie reviews. in [10], the authors used SVM, NB, and character-based N-gram for classifying the reviews about travel destination by using unigram frequency. In addition, in [11] several approaches mainly, SVM and rule-based classifiers with POS-tag and n-gram features are evaluated. They have been used to classify the movie reviews, product reviews, and space comments. SVM has also been used in [12, 13] for classification of movie reviews and MPQA2 Various features including the unigram, bigram, extraction pattern feature, adjective word frequency and percentage of appraisal groups have been used in these works, to train the classifier.
- Unsupervised learning approaches: Unfortunately, supervised methods require labelled data. It is time-consuming to collect these data for users. So, unsupervised approaches are used when no labelled data is available.
  Clustering is a class of algorithms used in unsupervised learning. Obviously, for the purpose of sentiment analysis, sentiment words or phrases are the main indices for clustering. Several approaches have been proposed for this purpose. Turney [14] has presented a simple unsupervised learning algorithm to classify if a review is recommended or not. He uses Pointwise Mutual Information (PMI) to indicate whether a word is positive or negative. PMI, has also been used in this work to indicate how strong is a word in terms of positiveness or negativeness. For each word, it calculates co-occurrence of the word with positive seed word(“excellent”) and negative seed word(“poor”) as follow:

  $$PMI(word_1, word_2) = \log_2\left(\frac{p(word_1 \& word_2)}{p(word_1)p(word_2)}\right) \quad (1)$$

  $word_2$ can be "excellent" or "poor". This value is called the semantic orientation (SO), which is used to classify the reviews. Harb et al. [15] used an unsupervised approach for extracting opinions from blogs. That performs blog classification by starting with the two sets of positive words which include: "good", "nice", "excellent", "positive", "fortunate", "correct", and "superior" and negative words which include: "bad", "nasty", "poor", "negative", "unfortunately", "wrong", and "inferior" as seeds. These words have the semantic orientation (SO) as in [14] and Google's search engine has also been used to create association rules that find more related words.

[2] Available at: http://www.cs.pitt.edu/mpqa/databaserelease

Lexicon-based approaches are also common for SA. These methods use the dictionary of emotional words along with the sentimental score for each word for the classification of data. These approaches are divided into two categories in the literature [7], Dictionary-based and Corpus-based approaches:

- Dictionary-based Approaches: These approaches are based on lexicon dataset and do not require the large corpora to build sentiment lexicon.
  In these approaches, firstly a small set of opinion words is collected by hand [7], which then grows by searching for the synonyms and antonyms of the opinion words in WordNet [16] or any other similar thesaurus. For instance, in [17] a simple dictionary-based approach is proposed for identifying the useful parts of sentiment in the sentence using two positive and negative word lists that were manually collected. WordNet is used to find the synonyms and antonyms that are added to the seed list.
- Corpus-Based Approaches: These approaches try to find new sentiment words in a large corpus using a syntactic or co-occurrence pattern and also a seed list of opinion words. So, a large corpus is required in these methods to reach to a good coverage. [18] is an import turning point among the corpus-based approaches. A seed opinion adjective list and several language constraints are used to determine the orientation of words in this work. The most significant constraints are "AND" and "But". "AND" indicates that the adjectives have the same orientation and "But" indicates that the adjectives have non-identical orientation. Based on these constraints a relation graph is created, in which the vertices indicate the words and the edges indicate the relation between words. Then a clustering algorithm is applied on the graph to categorize the words into two positive and negative categories.

In addition, SA has been studied at three levels, which are document, sentence and entity (and aspect) levels [3].

- Document Level: At this level, the propose is classifying an opinion document in term of positive, negative, or neutral polarity.
- Sentence Level: At this level the polarity, positive, negative, or neutral (no opinion) is determined for each sentence. Every sentence in this level considered as a short document which can be subjective or objective. This level is closely related to subjectivity classification which distinguishes objective sentences (factual information) from subjective sentences (express opinion) [3].
- Entity and Aspect Level: At this level, the purpose is to extract sentimental information on the aspect of items, which is divided into two sub-tacks of aspect extraction and aspect sentiment classification. The goal of aspect extraction is to identify the aspect that have been evaluated, and in aspect sentiment classification, the goal is to indicate polarity of opinion (positive, negative, or neutral) on the various aspects of items.

2.2. Multi-domain Sentiment Analysis

Domain is the collection of documents about a similar topic [19]. Therefore, documents about DVD, book, and electronic can be called DVD domain, book domain, and electronic domain respectively. In spite of the availability of huge number of opinions on any subject, researchers are seeking to create domain adaptation model. In these models, a classifier is trained in several domains and then is used to classify domains that have not used in making the model. Several researches have been performed on multi-domain SA. Ohana et al [20] proposed a case-based approach for cross domain sentiment classification, that includes two main parts: case description and case solution. The case description is a document signature used for later retrieval. A n-dimensional feature vector is used for the case description, which includes extracted features per document. For the case solution, they record all of the sentiment lexicons during the training on the document represented by the case that made a correct prediction. Then they used k-nearest cases to predict the polarity of a document.

In [21] authors used fuzzy membership function for multi-domain SA. Their approach include four basic steps:

- Feature extraction step: In this step, the Stanford NLP Parser is used to extract linguistic features from documents.
- Preliminary learning step: In this step for each extracted feature from previous phase which is called "PL" phase, have bean the two value of preliminary polarity and domain belonging degree are computed. The following fuzzy function is used to compute preliminary polarity:

$$p_i^E(C) = \frac{K_C^i}{S_C^i} \in [-1, 1] \; \forall i = 1, \dots, n \tag{2}$$

  and TF-IDF measure is used in computation of document belong degree.
- Information refinement step: Unbalanced data set may affect the quality of the polarity of the features in the sentiment model. To avoid its impact, they combined the result of the "PL" phase with the two linguistic source General Inquirer[3] and SenticNet[4], which it affects the core of final fuzzy membership function.
- Polarity aggregation step: In this step, the information obtained for each feature from "PL" phase and the "information refinement" phase, combined using a fuzzy membership function to determine the final polarity of the documents.

M.Dragoni et al [22], proposed a neural word embedding approach for multi-domain SA. They initially created an embedding vector for all words in the documents using Skip-gram model. Then they created a deep neural network on these word embeddings with two output layers. The domain identification layer, that identifies probability of the input documents to belong to one of the domains in dataset, and the polarity identification layer that identifies the polarity of the documents in the domains. At the end, they aggregated the results of two layers to compute the final polarity of the documents.

2.3. Deep learning in natural language processing and Sentiment Analysis

Recently the multi-layer neural network has gained special significance in NLP. These methods are capable of delivering advanced and acceptable results at a higher pace. This success is due to the availability of large amount of data and emergence of Graphical Processing Units (GPU). Deep learning (DL) has emerged as a highly impressive technique in machine learning to perform text mining, which consists of various tasks like text classification, sentiment analysis, question answering systems, semantic analysis, etc [23].

One of the most import issues that should be taken into account when working with the neural network on the text, is that we cannot directly feed the raw text to the neural network, as the neural

[3] http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm
[4] http://sentic.net/.

network receives D-dimensional feature vectors. One-hot encoding is not suitable due to the length of the dictionary size. So, there is a need to embed each feature into a D-dimensional space and represent it as a dense vector in the space. The solution is called word embeddings, which creates a single D-dimensional dense vector for each feature. The vectors are very flexible and help avoiding the curse of dimensionality. Some of the most import deep learning methods that are used in NLP, are described in the following sections:

2.3.1. Convolution Neural Networks (CNNs)

The CNNs are one of the most popular deep learning architectures that were first used in computer vision. CNNs are a special type of the feed-forward neural network that has properties such as: (i)convolution layer, (ii)sparse connectivity, (iii)parameter sharing, (iv)pooling [24]. The CNNs have three main layers [25]:

- Convolution layers: These layers are feature selection layers, that try to make various feature maps by utilising various kernels to convolve the whole data and the intermediate features map.
- Pooling layers: The task of these layers are to reduce the number of network parameters, and hence the problem of the overfitting, is controlled.
- Fully connected layers: These layers are calculate the rating of the categories (in the SA the categories can be positive, negative, or neutral). Moreover, they contain the largest number of network parameters to be learned.

The performance of these three layers is discussed in more detail, in Section 4.

In recent years the CNNs have gained great efficiency for various task such as image recognition [26], speech recognition [27], and NLP [28]. For example, in [29] authors proposed a single convolution neural network that was capable of performing various NLP tasks include: part-of-speech tagging, chunking, named entity recognition, and semantic role labelling.

2.3.2. Recurrent Neural Networks (RNNs)

RNNs are a type of neural networks that have direct cycles between their units. These cycles allow the neural network to make the internal state of the network and add the concept of time to the model, that allows the network to display dynamic behaviour. There may also be a feed-forward neural network without cycle between its components.

At time t each network node receives two input from the current inputs x^t and the hidden node values h^{t-1} .Based on these inputs, the output of the hidden layer h^t is obtained from the following equation [30]:

$$h^{(t)} = \sigma(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h) \tag{3}$$

Here W^{hx} is the matrix of weight between the input and the hidden layer, W^{hh} is the matrix of recurrent weights between the hidden layer and itself at adjacent time steps. σ is activation function, b_h is bias parameter, $x^{(t)}$ is current input data, and $h^{(t-1)}$ is previously hidden layer output. The network output is also obtained from the following equation:

$$y^{(t)} = softmax(\ W^{yh}h^{(t)} + b_y\ ) \tag{4}$$

Here W^{yh} is the matrix of weight between the input and the output layer, $softmax$ is a nonlinearity function that used for multi-class classification, $h^{(t)}$ is the output of the hidden layer is obtained by the equation (3), and b_y is bias parameter. The bias allows each node to learn an offset.

2.3.3. Long-Short-Term-Memory (LSTM)

RNNs suffer from vanishing gradient problem. This problem is difficulty found in certain Artificial Neural Networks (ANN) with gradient-based methods (e.g. Back Propagation). RNNs cannot capture long-term dependencies due to vanishing gradients during back-propagation. LSTM is a type of RNN architecture that addresses the vanishing gradient problem and allows learning of long-term dependencies. The primary idea of LSTMs was proposed by German researchers Hochreiter and Schmidhuber [31], to avoid the long-term dependency problem. These networks keep more information than the recurrent network in a cell. The information inside this cell can be read, written, and stored like computer memory. Each cell has four gates, namely the input gate i, the output gate o, the forget gate f, and the cell update gate g. At each time step, there are the following values for each gate [32]:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{5}$$

$$f_t = \sigma\big(W_f.[h_{t-1}, x_t] + b_f\big) \tag{6}$$

$$g_t = tanh\big(W_g.\big[h_{g-1}, x_t\big] + b_g\big) \tag{7}$$

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{8}$$

where $\left[W_i, W_f, W_g, W_o, b_i, b_f, b_g, b_o\right]$ are the set of parameters to be learned.

Several deep learning approaches used to address the SA problem. For instance, in [33] authors have used a CNN for sentence-level sentiment classification. Moreover, in [34] another solution has been proposed for SA based on CNN on twitter data. KS. Tai et al [35] proposed tree structure of LSTMs, that is called "Tree-LSTM" for two tasks of predicting the semantic relatedness of two sentences and sentiment classification.

In general, CNN is suitable for text classification but in sequenced tasks cannot play a useful role. However, the LSTM is more appropriate for sequenced tasks.

3.material

3.1. The Dranziera Dataset

The Dranziera protocol [36] is used to evaluate the proposed model. This protocol includes 1 million shopping reviews, which is compiled from the Amazon website for the various product. These reviews are about 20 different domains, which is called in-domain models (IDMs) and includes the items in Table 1.

Table 1. 20 different domains of Dranziera dataset

Amazon Instant Video	Automotive
Baby	Beauty
Books	Clothing Accessories
Electronics	Health
Home Kitchen	Movies TV
Music	Office Products
Patio	Pet Supplies
Shoes	Software
Sports Outdoors	Tools Home Improvement
Toys Games	Video Games

For each domain, 25000 positive reviews and 25000 negative reviews are collected. One of the most import advantages of this protocol is its balanced data, means equal number of positive and negative reviews for each domain. Dividing each domain into 5 folds allows distinguishing between the review that are used for training and testing the model.

In addition to the IDMs in the Dranziera, there are seven other domains that are used only for testing the model and have not been used in the training phase. This will enable us to provide a more general model. It can also work well on domains that are not used to build the model. These domains have been called out-model domains (OMDs) and are shown in the Table 2.

Table 2. out-model domains

Cell Phones Accessories	Gourmet Foods
Industrial Scientific	Jewelry
Kindle Store	Musical Instruments
Watches	

for each of these domains 5000 positive review and 5000 negative reviews were collected.

3.2. Deep Learning Library

The Keras [37] is a high-level library for creating neural network tools written in python and capable of running on TensorFlow[5], CNTK[6] and Theano[7] Keras includes several implementations of neural network structure blocks, such as layer, objection, activation function, and optimizer as well as numerous tools for images and text data. This API is compatible with python version 2.7-3.6 and can be implemented seamlessly on CPU and GPU. Keras focuses on user-friendliness, modularity, and flexibility. Keras can be integrated with low-level languages such as TensorFlow, which gives a high flexibility it. Keras uses its own graph data structure to build a computational graph, that does not rely on the back-end framework. As a result, there is no need to learn to program the back-end framework.

The Keras has been used in current work for developing WCNNE model described in section 4.

[5] https://www.tensorflow.org/
[6] https://cntk.ai/pythondocs/
[7] https://pypi.org/project/Theano/

3.3. Word Embeddings

Generally, for creating dense vectors the word embedding methods are trained on a large volume dataset. The pre-trained word2vec-GoogleNews-vectors word embedding has been used in our approach. It trained on 100 billion words from Google News producing a vocabulary of 3 million words that are available here[8].

3.4. Evaluation Measures

The efficiency of a sentiment classifier is determined by applying it to test data. Various measures can be used for evaluation of the binary classifier. In this article, four measures have been used to evaluate our proposed sentiment classifier.

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall \; = \frac{TP}{TP+FN} \quad 10)$$

$$F_1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (11)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

where true positive (TP) is the number of positive documents that are classified as positive, true negative (TN) is the number of negative documents that are classified as negative, false positive (FP) is the number of positive documents that are classified as negative, and false negative (FN) is the number of negative documents that are classified as positive.

4. Method

First, before applying and describing the ensemble models, we trained the models on in-domain data that described in section 4.1 as 80% training and 20% testing. Three basic models, CNN-Multi channel [33], LSTM [38], and Bi-GRUCapsule [39], were used in this analysis. The results of these basic models are presented in Table 3. The Bi-GRUCapsule model has a higher accuracy than other models.

Table 3. The experiment result on many baseline methods

Model	Accuracy		Precision		Recall	
-	Train	Test	Train	Test	Train	Test
CNN-Multi channel[33]	0.8873	0.8875	0.8834	0.8627	0.9331	0.9207
LSTM [38]	0.9360	0.9283	**0.9390**	0.9334	0.9322	0.9218
Bi-GRUCapsule [39]	**0.9423**	**0.9345**	0.9231	**0.9347**	**0.9432**	**0.9336**

[8] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

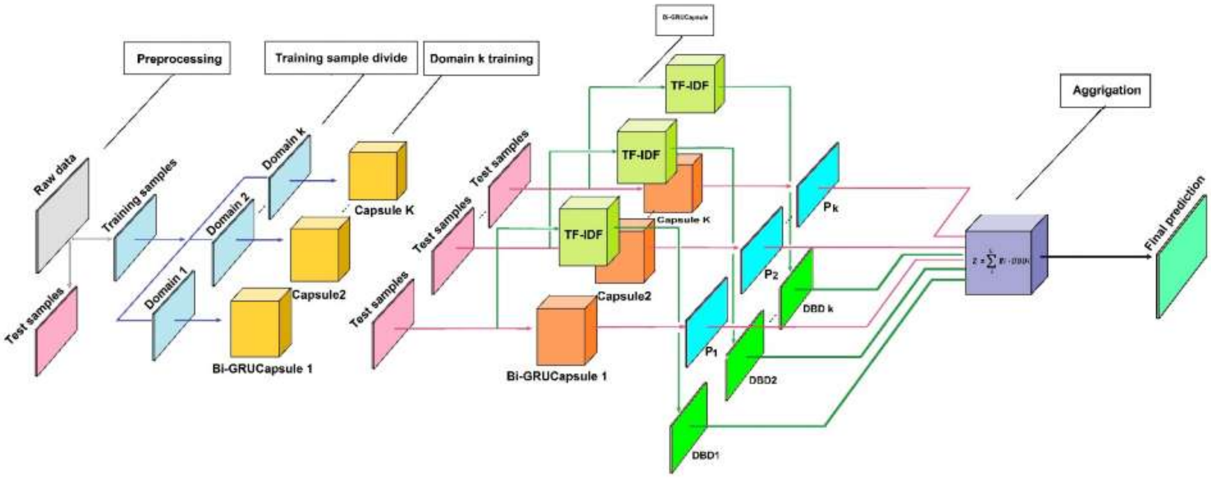


Figure 1. Architecture of the proposed Weighted Bi-Directional GRU Capsule Ensemble (WBi-GRUCapsuleE).

An illustration of the proposed architecture is presented in Figure 1. The architecture consists of four main phases: data preparation and word embeddings, training Bi-GRUCapsule networks, calculating domain belonging degree, and results aggregation. The details of each phase are described below.

4.1. Data preparation and word embeddings

At the beginning of this phase, each document (review) is considered as a sequence of words. Afterward, all stop-words (commonly used words such as "the") and punctuations are removed from these words' sequences. Removing these items affects the accuracy of the classifier and make it learn essential features. Next, the input sequence is mapped to a sequence of low-dimensional distributed representations by looking up word embeddings table, T^{emb}. After mapping, the sequence of the words can be displayed as $s = [X^1, X^2, \dots, X^n]$ that $X^i \in \mathbb{R}^K$ is a k-dimensional word vector corresponding to the i-th word in the sequence.

One of the limitations of Bi-GRUCapsule is requiring a fixed dimension for the input vectors, while the input vectors have different lengths. In order to solve this problem, in general, a threshold is used and all the inputs are padded by zero when the length of sequence is less than the threshold, or cropped in the case of the length is greater. To address this limitation in our method, the maximum sequence length in the dataset is considered as the threshold to prevent from cropping, because some useful data may be lost.

Each input sequence is converted into a $Maxreviewlength * K$ dimensional matrix which the rows are equal to words and the columns are the word embeddings vectors. A sequence of length n is represented as:

$$X^{1:n} = X^1 \oplus X^2 \oplus X^3 \oplus \dots \oplus X^n \tag{13}$$

where $\oplus$ is the concatenation operator.

4.2. Training Bi-GRUCapsule networks

In this phase, a Bi-GRUCapsule is trained for each domain. First, various configuration hyper parameters of the networks, like the depth and the number of filters, have been investigated to create the Bi-GRUCapsule.

The Bi-GRU Capsule network consists of four layers:

1. Words embedding layers:
   in this layer, each of the documents is converted to dense vectors according to the pattern described in previous step, which derived from Word2vec words embedding. The output of this layers for each document is equal to:

   $$Out_{embed} = M * E \tag{14}$$

   Where M is the maximum document length in the entire dataset and E is the size of the dense vector, which equal to 300.
2. Bidirectional GRU layer:
   Our GRU inputs are Out_{embed}that derived from the embedding layer. If this matrix is represented by $Out_{embed} = [X_1, X_2, \dots., X_n]$, then GRU's input in step t is equal to $X_t \in R^{300}$. Based on these inputs the $h^t = [h_1, h_2, \dots, h_t]$ that represent the hidden vectors sequence in GRU is calculated by the following equations:

   $$z_t = \sigma(w_z x_t + U_z h_{t-1} + b_z \tag{15}$$

   $$r_t = \sigma(w_r x_t + U_r h_{t-1} + b_r \tag{16}$$

   $$h'_t = \sigma(w_h x_t + U_n(r_t \odot h_{t-1}) + b_h \tag{17}$$

   $$h_t = (1 - z)h_{t-1} + z_t h'_t \tag{18}$$

   Where z_t is update gate, r_t is rest gate, h'_t is candidate gate, and h_t is output activation. $[W_Z, W_R, W_N, U_Z, U_R, U_N]$ learnable matrixes, $[b_n, b_z, b_r]$ learnable biases, σ sigmoid activation function, and $\odot$ an elementwise multiplication. GRUs in normal model will simulate input in one direction. In this step we used the bi-direction mode.
3. Capsule layer:

   Bi-GRU's encoded features are given to a CapsuleNet. This network includes a set of capsules. The capsule layer covert the scalar features extracted by the Bi-GRU layer into vector-valued capsules to capture the input sequence features. If Bi-GRU output is h_i, and w is a weighted matrix, then $\hat{v}_{i|j}$, which represents the predictor vector, is obtained from the following equation.

   $$\hat{v}_{i|j} = w_{ij} h_i \tag{19}$$

The set of inputs to a capsule s_j is a weighting set of all prediction vectors $\hat{v}_{i|j}$, which is computed according to the equation (20).

$$S_j = \sum_i c_{ij}\,\hat{v}_{i|j} \tag{20}$$

Where c_{ij} is the coupling coefficient, which is repeatedly adjusted by Dynamic Routing algorithm [12].

The “squash” is used as a non-linear function for mapping the values of S_j vectors to [0-1]. This function is applied to S_j according to the following equation.

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \cdot \frac{s_j}{||s_j||} \tag{21}$$

The output of a capsule is a vector and it can be selected to which one of the higher-level capsules to send. In the proposed architecture, Dynamic Routing [Sara Sab] was used for the routing mechanism.

4. Classification Layer:

The flattened outputs of the capsule layer, represented by F, are given to a fully connected layer of 2 neurons.

$$P = W_{dense} * F \tag{22}$$

The output of P should be such that it represents the probability of each of the 2 class. For this purpose, we use the $Sigmoid$ function, which is calculated for each $f_i \in F$ as follows:

$$p_i = \frac{1}{1 - e^{-f_i}} \tag{23}$$

At this point, each of the twenty networks is individually trained on each domain (20 domains of the in-domain). The output of any network is the probability that the document will be positive or negative in that domain. Due to better differentiation in the results aggregation step, the output of the sigmoid functions (P) produced by the Bi-GRUCapsule models, originally in the range of $[0, 1]$, are normalized based on the equation 1 to range of $[-0.5, 0.5]$:

$$P_i = \begin{cases} P_i - 0.5, & if\ round(P_i) = 1 \\ -P_i, & if\ round(P_i) = 0 \end{cases} \tag{24}$$

4.3. Calculating TF-IDF (DBD)

Domain Belonging Degree (DBD) value is obtained from the Equation (25) based on TF and IDF factors:

$$DBD(T, d_i) = TF(T, d_i).IDF(T, d_i) \tag{25}$$

where TF of the term T in the domain d_i, $TF(T, d_i)$, is obtained from the equation 3 and the IDF for term T is computed by equation 27:

$$TF(T, d_i) = \frac{n_{T_i}}{N_i} \tag{26}$$

$$IDF(T, d_i) = \frac{n_{T_i}}{\sum_{j=1}^{M} n_{T_j}} \tag{27}$$

where n_{T_i} is the number of times that T occurs in the domain d_i and N_i is the sum of occurrence number of all the terms within the domain d_i. Also, M is the number of domains in the training set (in our case $M = 20$), and n_{T_j} is the number of occurrences of term T in the domain j.

4.4. Results aggregation

In this phase, the final polarity of each document obtained by integrating the results from the Bi-GRUCapsule networks in the second phase and the DBD values gained in the third phase. For this, the following equation is used:

$$Z = \sum_{i}^{K} P_i . DBD_i$$
$$Polarity(Z) = \begin{cases} 1 \quad positive, & if\ Z \geq 0. \\ -1 \quad negative, & otherwise. \end{cases} \tag{28}$$

where P_i is the predicted polarity for the input document by the Bi-GRUCapsule model i.

5. Experiments

5.1. Implementation and results

The accuracy of the weighted ensemble models on the DRANZIERA dataset has been compared with two other state-of-the-art systems, NeuroSent [22] and DAP (Domain Aggregation Polarity) [21], The results are presented in Table 4 and Table 5 for the in-domain and the out-domain data respectively. As shown in the Table 4, the weighted ensemble models outperform the NeuroSent for all 20 in-domains with the maximum average accuracy improvement of 0.0303. Likewise, Table 4 represents the maximum average accuracy improvement of 0.0261 on out-domains data in comparison with the NeuroSent. Unfortunately, the detailed accuracy results are not available for DAP [21] for each domain in the both tables. We also compared the WBi-GRUCapsuleE model with the three basic models to investigate the effect of the DBD weight factor. We performed comparisons for all three baseline models in equal weighted and non-weighted data conditions.

Table 4. Accuracy on in-domain data

Domain	SVM	NB	ME	DBP	DDP	IRMUDO	NeuroSent	Bi-GRUCapsuleE	WBi-GRUCapsuleE
Amazon Instant Video	0.7017	0.6544	0.7026	0.7230	0.7147	0.7751	0.8017	0.8535	**0.8605**
Automotive	0.7166	0.7172	0.7172	0.7202	0.6943	0.7412	0.8537	0.8769	**0.8801**
Baby	0.6885	0.6929	0.7155	0.7088	0.6938	0.7652	0.8518	0.8698	**0.8808**
Beauty	0.6982	0.7023	0.7230	0.7481	0.7341	0.7797	0.8550	**0.8890**	0.8886
Books	0.6923	0.6873	0.6887	0.6957	0.6926	0.7315	0.7966	**0.8324**	0.8202
Clothing Accessories	0.6988	0.6904	0.7224	0.8038	0.7856	0.8462	0.8696	0.8783	**0.8871**
Electronics	0.6851	0.6880	0.6988	0.7309	0.7035	0.7492	0.8641	0.8829	0.8846
Health	0.6717	0.7205	0.6629	0.6887	0.6867	0.7527	0.8611	**0.8793**	**0.8793**
Home Kitchen	0.7217	0.7178	0.6900	0.7137	0.6929	0.7683	0.8686	0.9021	**0.9073**
Movies TV	0.7354	0.6915	0.7160	0.7030	0.7122	0.7743	0.8090	**0.8634**	0.8531

Music	0.6936	0.6701	0.6542	0.7171	0.7216	0.7834	0.8083	0.8023	0.8278
Office Products	0.7321	0.6910	0.7314	0.7298	0.7017	0.7523	0.8730	0.8907	**0.8964**
Patio	0.6875	0.6923	0.7142	0.7024	0.6926	0.7459	0.8564	**0.8900**	0.8852
Pet Supplies	0.6817	0.7078	0.7302	0.6680	0.6626	0.7195	0.8361	0.8446	**0.8698**
Shoes	0.6705	0.7164	0.7276	0.8324	0.8115	0.8434	0.8655	0.8587	**0.8941**
Software	0.7395	0.6762	0.6872	0.7196	0.7151	0.7462	0.8479	0.8662	**0.8756**
Sports Outdoors	0.6685	0.7050	0.7314	0.7084	0.7129	0.7927	0.8669	**0.8945**	0.8939
Tools Home Improvement	0.7325	0.6896	0.7356	0.6842	0.6887	0.7438	0.8518	0.8642	**0.8850**
Toys Games	0.6636	0.6664	0.6948	0.7383	0.7108	0.8018	0.8624	0.8930	**0.8983**
Video Games	0.6954	0.6808	0.7038	0.6999	0.7012	0.7590	0.8206	0.8534	**0.8578**
Average	0.6987	0.6929	0.7074	0.7218	0.7115	0.7686	0.8460	0.8692	**0.8763**

Proposed ensemble approaches in the out-domain data also achieve acceptable results. Among these approaches, WBi-GRUCapsuleE had the most accuracy on all domains. This approach increases the generalization power of the model for out-domain data (data not used in training) because of considering features as feature vectors.

Table 5. Accuracy on out-domain data

Domain	SVM	NB	ME	DBP	DDP	IRMUDO	NeuroSent	Bi-GRUCapsuleE	WBi-GRUCapsuleE
Cell Phones Accessories	0.6671	0.6209	0.6904	-	0.6675	0.7032	0.8431	**0.8635**	0.8632
Gourmet Foods	0.6376	0.6257	0.6384	-	0.6738	0.7638	0.8227	0.8608	**0.8625**
Industrial Scientific	0.6175	0.6234	0.6301	-	0.6392	0.6821	0.8155	0.8159	**0.8404**
Jewelry	0.6003	0.6191	0.6423	-	0.6628	0.7826	0.8712	0.8745	**0.8969**
Kindle Store	0.6877	0.6102	0.6337	-	0.7105	0.7560	0.8054	0.8112	**0.8176**
Musical Instruments	0.6949	0.6140	0.6827	-	0.6938	0.7811	0.8597	0.8747	**0.8864**
Watches	0.6944	0.6682	0.6330	-	0.6889	0.7867	0.8567	0.8693	**0.8898**
Average	0.6571	0.6260	0.6501	-	0.6766	0.7508	0.8392	0.8528	**0.8653**

In addition, some more detailed results of the other evaluation scores, i.e., precision, recall, and F1-measure, are shown in the Table 6 and Table 7 for all systems, which respectively indicate our approaches got better results on the both in-domain and out-domain data in some scores. As shown in Table 6 and Table 7, the average scores of the WBi-GRUCapsuleE and Bi-GRUCapsuleE method on the in-domain data is better than the other approach. Table 6 also reflects that WBi-GRUCapsuleE is gained better result on average precision in comparison with the DAP approach. Similar result obtained for the out-domain data as shown in the Table 7. The average precision of the WBi-GRUCapsuleE is greater than NeuroSent and DAP by 0.0186 and 0.112 respectively and the average F1-measure of the WBi-GRUCapsuleE shows 0.022 and 0.005 improvements in comparison to the NeuroSent and DAP, however in average recall the DAP approach is better than our approaches.

Table 6- Results for the in-domain data

Approach	Avg. Precision	Avg. Recall	Avg. F1
SVM	0.6890	0.7097	0.6987
NB	0.6956	0.6915	0.6929
ME	0.7073	0.7085	0.7074
DBP	0.7108	0.7331	0.7218
DDP	0.6731	0.7546	0.7115
IRMUDO	0.7410	0.7984	0.7686
GWE	0.8008	0.7921	0.7964
NeuroSent	0.8515	0.8407	0.8460
Bi-GRUCapsuleE	0.8696	0.8702	**0.8680**

WBi-GRUCapsuleE	**0.8785**	**0.8766**	0.8611

Table 7- Results for the out-domain data.

Approach	Avg. Precision	Avg. Recall	Avg. F1
SVM	0.6507	0.6437	0.6571
NB	0.6435	0.6459	0.6260
ME	0.6524	0.6475	0.6501
DBP	- - -	- - -	- - -
DDP	0.6609	0.6931	0.6766
IRMUDO	0.7848	0.8120	0.7848
GWE	0.7930	0.7864	0.7896
NeuroSent	0.8442	0.8343	0.8392
Bi-GRUCapsuleE	0.8539	0.8607	0.8578
WBi-GRUCapsuleE	**0.8628**	**0.8611**	**0.8610**

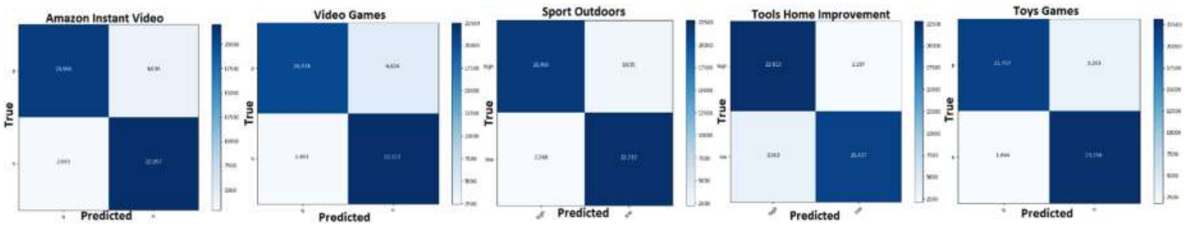
6. Error analysis and future works

Figure 2 shows the number of TP, FP, FN, and TN obtained from the proposed approach WBi-GRUCapsuleE on the in-domain data. As shown, the Music domain has the highest FP value, and the Software domain has highest FN value. Similarly, Figure 3 demonstrates the number of TP, FP, FN and TN obtained from the proposed approach on the out-domain data. As shown in the figure, Kindle Store has highest FP value and Cell Phones Accessories has the highest FN value.

By manual and statistical study of 10000 misclassified reviews, we found out the detection of the negation scope has led to the incorrect classification of these reviews for the most part. Therefore, as a part of future work, our next goal is to apply negation scope as a manual feature along with the features selected automatically by the WBi-GRUCapsuleE. In [2], the author used 19 negative words with 90 patterns to detect the negation scope which led to increase the accuracy of their classifier. Similar patterns can be used to improve the proposed approach.

We suggest more sub-tasks to improve the proposed approach. These sub-tasks are promising ways to improve the WBi-GRUCapsuleE:

1. Detection of sarcasm with a new algorithm.
2. Applying some more pre-processing steps for noise reduction from the original reviews, like replacing some irregular forms of words with their correct forms.
3. Generally, one of the problems with using pre-trained word embeddings is that calculated word-vectors do not contain sentimental information. Authors in [40], proposed Improved Word Vector (IWV) approach to address this problem. By combining this algorithm with the WCNNE, we hope to get more improvements in our results.



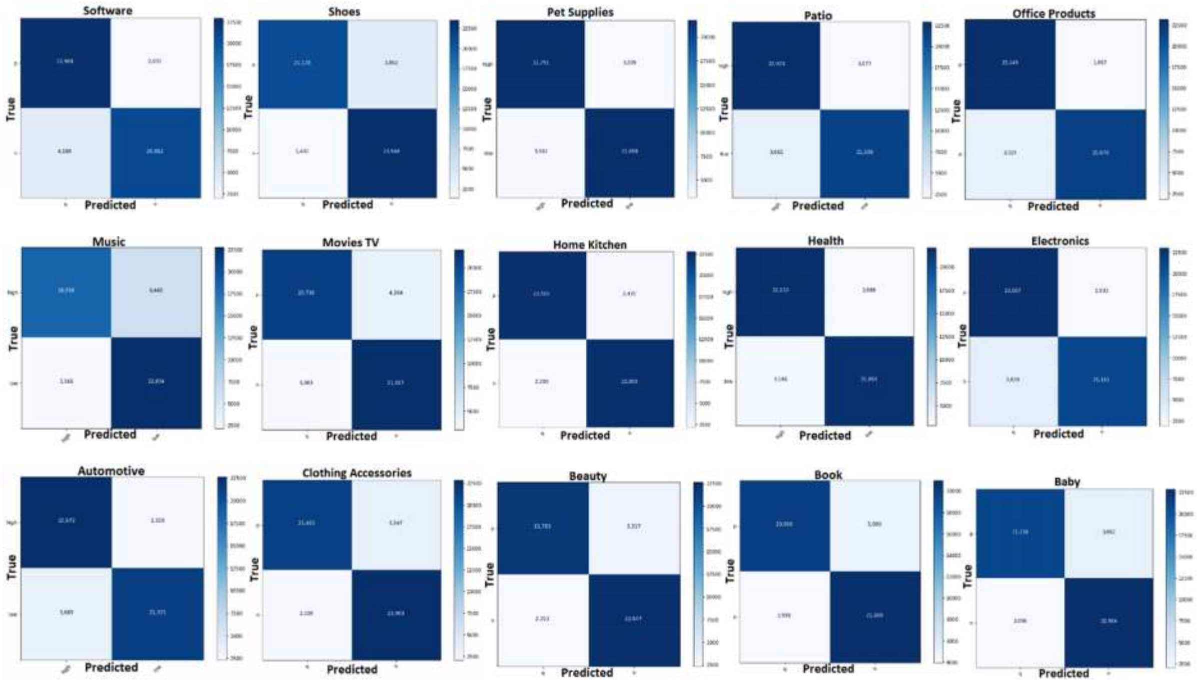


Figure 2. Number of $\boldsymbol{TP}$, $\boldsymbol{FP}$, $\boldsymbol{FN}$, and $\boldsymbol{TN}$ samples Obtained by WBi-GRUCapsuleE approach on the in-domain data

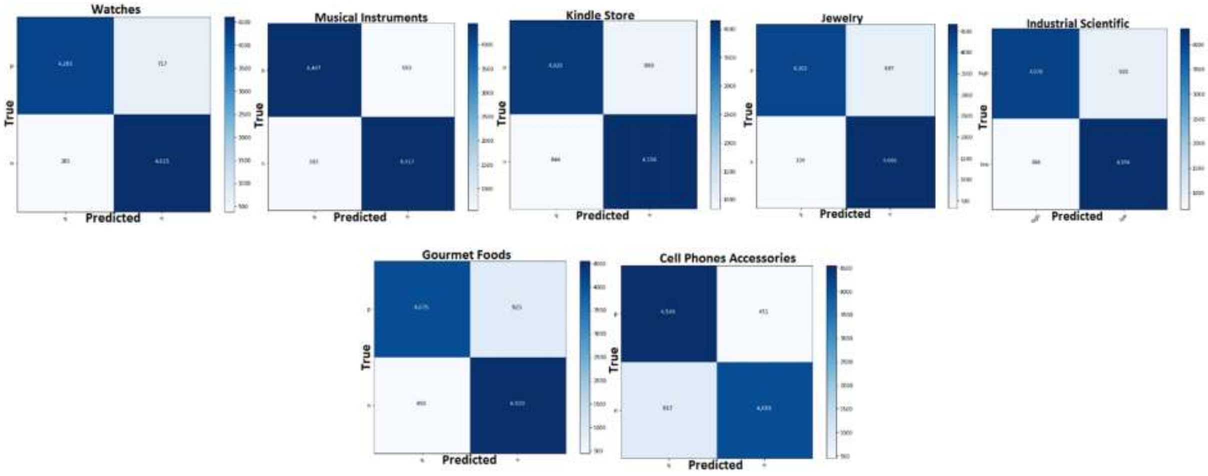


Figure 2. Number of $\boldsymbol{TP}$,$\boldsymbol{FP}$,$\boldsymbol{FN}$ and $\boldsymbol{TN}$ samples Obtained by WBi-GRUCapsuleE approach on the out-domain data

7. Conclusion

In this paper, we proposed novel methods based on the weighted mechanism for multi-domain SA exploiting linguistic overlaps between domains for inferring document polarity. The suggested approaches use a combination of trained CNN, LSTM, and Bi-GRUCapsule models on specific domains and DBD to estimate final polarity. We used DBD for the weighing mechanism, which for each document takes into account the domain belonging degree. The efficiency of the WBi-GRUCapsuleE approach was evaluated by the DRANZIERA protocol and the results demonstrate the success of the proposed approach in comparison with the related state-of the-art systems.

References

[1] Routray, P., Swain, C. K., & Mishra, S. P. (2013). A survey on sentiment analysis. *International Journal of Computer Applications*, 76(10).

[2] Torabian, B. (2016). Sentiment classification with case-base approach.

[3] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1-167.

[4] Bollegala, D., Weir, D., & Carroll, J. (2012). Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE transactions on knowledge and data engineering*, 25(8), 1719-1731.

[5] Gindl, S., Weichselbraun, A., & Scharl, A. (2010). Cross-domain contextualisation of sentiment lexicons.

[6] Zhang, H., Gan, W., & Jiang, B. (2014, September). Machine learning and lexicon based methods for sentiment classification: A survey. In *2014 11th web information system and application conference* (pp. 262-265). IEEE.

[7] Shepelenko, O. (2017). Opinion mining and sentiment analysis using Bayesian and neural networks approaches (*Doctoral dissertation, Master thesis, University of Tartu, Institute of Computer Science*).

[8] Zhang, Z., Ye, Q., Zhang, Z., & Li, Y. (2011). Sentiment classification of Internet restaurant reviews written in Cantonese. *Expert Systems with Applications*, *38*(6), 7674-7682.

[9] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. arXiv preprint cs/0205070.

[10] Ye, Q., Zhang, Z., & Law, R. (2009). Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert systems with applications, 36(3), 6527-6535.

[11] Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetric*, *3*(2), 143-157.

[12] Riloff, E., Patwardhan, S., & Wiebe, J. (2006, July). Feature subsumption for opinion analysis. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 440-448).

[13] Whitelaw, C., Garg, N., & Argamon, S. (2005, October). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 625-631).

[14] Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *arXiv preprint cs/0212032*.

[15] Harb, A., Plantié, M., Dray, G., Roche, M., Trousset, F., & Poncelet, P. (2008, October). Web Opinion Mining: How to extract opinions from blogs?. In *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology* (pp. 211-217).

[16] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, *3*(4), 235-244.

[17] Kim, S. M., & Hovy, E. (2004). Determining the sentiment of opinions. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 1367-1373).

[18] Hatzivassiloglou, V., & McKeown, K. (1997, July). Predicting the semantic orientation of adjectives. In *35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics* (pp. 174-181).

[19] Blitzer, J., Dredze, M., & Pereira, F. (2007, June). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 440-447).

[20] Ohana, B., Delany, S. J., & Tierney, B. (2012, September). A case-based approach to cross domain sentiment classification. In *International Conference on Case-Based Reasoning* (pp. 284-296). Springer, Berlin, Heidelberg.

[21] Dragoni, M., & Petrucci, G. (2018). A fuzzy-based strategy for multi-domain sentiment analysis. *International Journal of Approximate Reasoning*, *93*, 59-73.

[22] Dragoni, M., & Petrucci, G. (2017). A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Transactions on Affective Computing*, *8*(4), 457-470.

[23] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, *19*(1), 22-36.

[24] Rojas-Barahona, L. M. (2016). Deep learning for sentiment analysis language and linguistics. Compass 10: 701–719.

[25] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27-48.

[26] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[27] Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, *22*(10), 1533-1545.

[28] Moreno Lopez, M., & Kalita, J. (2017). Deep Learning applied to NLP. *arXiv e-prints*, arXiv-1703.

[29] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, *12*(ARTICLE), 2493-2537.

[30] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

[31] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[32] Olah, C. (2015). Understanding lstm networks.

[33] Kim, Y. (2014). Convolutional neural networks for sentence classification. CoRR abs/1408.5882. *arXiv preprint arXiv:1408.5882*.

[34] Severyn, A., & Moschitti, A. (2015, August). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 959-962).

[35] Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

[36] Dragoni, M., Tettamanzi, A. G., & da Costa Pereira, C. (2016, May). Dranziera: an evaluation protocol for multi-domain opinion mining. In *Tenth International Conference on Language Resources and Evaluation (LREC 2016)* (pp. 267-272). European Language Resources Association (ELRA).

[37] Chollet, F. (2015). keras, GitHub. GitHub repository.

[38] Yogatama, D., Dyer, C., Ling, W., & Blunsom, P. (2017). Generative and discriminative text classification with recurrent neural networks (2017). *arXiv preprint arXiv:1703.01898*.

[39] Kim, J., Jang, S., Park, E., & Choi, S. (2020). Text classification using capsules. *Neurocomputing*, *376*, 214-221.

[40] Rezaeinia, S. M., Ghodsi, A., & Rahmani, R. (2017). Improving the accuracy of pre-trained word embeddings for sentiment analysis. *arXiv preprint arXiv:1711.08609*.