Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Weighted Focal Loss Function for Evaluating Effectiveness of Word Embeddings on Suggestion Mining from Opinion Reviews

Naveen Kumar Laskari[1], Suresh Kumar Sanampudi[2]

[1]Assistant Professor, IT, BVRIT HYDERABAD.

[2]Sr. Assistant Professor & Head, IT, JNTUHCEJ.

Abstract

Opinion reviews are significant in the purchase and decision-making process. Opinion Mining(OM) approaches are used to detect the sentiment at varying levels of granularity. It is worth noting that suggestions appear in opinion reviews, and mining these suggestions is regarded as suggestion mining. Suggestion mining seems to be a text classification technique that includes a variety of methodologies, ranging from traditional machine learning methods to deep models. For any of these models to work, the input text must be represented as a vector. Word embeddings are the process for encoding the input into a vector representation. This study proposes the effect of several widely used pre-trained word vectors combined with neural network architectures to utilize the weighted focus loss function. With the unique loss function established after examining the dependability of numerous models for the job, FasText and Glove embedding approaches fared substantially better with CNN and multi-layer LSTM architectures. The FastText embedding grabbed the information more effectively attributed to the sub-word information-based method, and the 1-D convolution operation captured the sequential information more effectively. Consequently, the CNN model can learn faster and achieve the best outcome. The models are implemented and tested using the datasets given by the SemEval-2019 organizers.

Keywords Opinion Reviews, Suggestion Mining, Neural Networks, Word Embeddings, and Loss function.

1.0 Introduction

Opinion reviews have become central and vital in any decision-making process in the web 2.0 era(Lin et al., 2016). Vast volumes of opinion reviews are generated on social platforms such as micro-blogs, e-commerce portals, and third-party web portals(Baby & B, 2020). These platforms are becoming the first place to express the users' voices and opinions about various things and the primary source for obtaining the opinionated text(Lin et al., 2016). The study and analysis of information from these

sources are considered sentiment analysis or opinion mining(Baby & B, 2020; Lin et al., 2016). In conventional times, opinion mining is applied towards extracting the sentiment expressed by the users as positive, neutral, or negative. Later this has been extended to mine sentiment towards each aspect mentioned in the opinionated text(B. Liu, 2016)(Laskari & Sanampudi, 2016). In due time, it is identified that the information expressed on these social platforms also consists of suggestions towards various aspects of entities and suggestions towards quality or service improvement(S. Negi & Buitelaar, 2017; Sapna Negi et al., 2018). In general, a suggestion can be defined as a tip or advice, or recommendation towards a specific activity for further improvement(S. Negi & Buitelaar, 2017; Sapna Negi et al., 2018). Suggestions help fellow customers utilize the services in a better way(S. Negi & Buitelaar, 2017; Sapna Negi & Buitelaar, 2015). The extraction of these suggestions from opinion reviews is termed suggestion mining.

Machine learning techniques are widely applied for most Natural Language Processing (NLP) tasks. In conventional times, Naïve Bayes(Krishna et al., 2019), Support Vector Machines (SVM)(Krishna et al., 2019), Random Forest (RF)(Krishna et al., 2019), etc. kind of approaches is effectively used with the combination of manual feature extraction methods. However, the due increase in the availability of data, computing ability, and the invention of new algorithms, training techniques, and neural network approaches gave noticeable results in various tasks of NLP. As a result, neural network architectures such as multi-Layer Perceptions (MLP), Recurrent Neural Networks (RNN)(Bengio et al., 2015), Convolutional Neural Networks (CNN)(Z. Liu et al., 2020)(Kim, 2014), and advanced encoder-decoder models(Xia et al., 2018; Yu et al., 2019) are the kinds of techniques effectively used for NLP tasks in the literature. The loss function in a neural network is used to evaluate the error between the actual output and the output value predicted by the model. With an optimization algorithm, the network's parameters are adjusted based on the computed loss value. For binary classification, binary cross-entropy is the popularly used loss function in neural networks(Baby & B, 2020). On the other hand, focal loss is the function of dealing with the unbalanced dataset(Mukhoti et al., 2020; Pasupa et al., 2020; Sun et al., 2019). Since the suggestion mining dataset is unbalanced and biased towards the non-suggestion class, the modified version of focal loss shows promising results for suggestion mining.

Suggestion mining can be defined as extracting sentences containing suggestions from the opinionated text. A standard problem definition is given in different publications on the topic of suggestion mining(S. Negi & Buitelaar, 2017; Sapna Negi et al., 2018; Viswanathan et al., 2011) as

Given a sentence s, predict a label for s where s € {suggestion, non-suggestion}

The above definition clarifies that suggestion mining can be considered a text classification problem. However, identifying and detecting the suggested reviews on social platforms is quite challenging. For example, consider a review from the hotel domain "**don't forget to tip**" such a small review gives a good suggestion to a customer about giving tips to staff. The suggestion intended sentences may express the suggestion in *implicit* or *explicit* mode, suggestion towards the *manufacturer* or a *peer-customer*. Nevertheless, using computational methods for detecting these without manual intervention is highly challenging. With consideration of the complex nature of the task, various approaches have been implemented so far by the research community, including linguistic techniques,

rule-based(Markov & De La Clergerie, n.d.; Sapna Negi et al., 2018), traditional machine learning, and deep learning methods(Markov & De La Clergerie, 2019; Sapna Negi et al., 2018).

The applications of suggestion mining are widespread. In earlier times, organizations used to seek suggestions from employees, customers, and known people for further improvement on various aspects. Individuals also seek suggestions from friends and relatives before purchasing or travel decisions. In industry 4.0, people tend to express and ask for suggestions on various social platforms. The automated suggestion mining mechanisms help product quality improvement(Viswanathan et al., 2011), product feature enhancement, customer-to-customer suggestion(S. Negi & Buitelaar, 2017; Sapna Negi & Buitelaar, 2015), summarizing suggestions for a particular product or entity, and recommender systems building.

Word embeddings map the discrete tokens into real-valued vector space of higher dimensions(Almeida & Xex, 2015; Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013; *What Are Word Embeddings for Text?*, 2015). The pioneering approaches used to represent the tokens were like one-hot encoding and term frequency-based, which does not capture any semantics. The major problem with the earlier methods was the nature of sparsity and the lack of semantics. Slowly the research moved towards representing the words or tokens in dense space by incorporating semantics(Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). Finally, a very effective dense representation by capturing semantics was developed with the advent of neural network-based mechanisms. In section 3, a more elaborated explanation is presented about word embedding and its types.

The rest of the paper is organized as follows: section-2 summarizes the literature study about suggestion mining. The methodology has been presented in section 3, along with the datasets, word embedding techniques, and neural network architectures. Section-4 illustrates the results obtained and the elaborated discussion on the results. Finally, the conclusion and future scope are presented in section-5.

2.0 Related Work

Suggestion mining is a relatively new problem in NLP and Sentiment Analysis. Most of the work made progress on suggestion mining, classifying the opinionated sentences into suggestion or non-suggestion(S. Negi & Buitelaar, 2017; Sapna Negi et al., 2018). The first time in the literature concept of suggestion mining was brought into the research aspect by Viswanathan, Amar et al. (Viswanathan et al., 2011). The authors extracted information from reviews collected from mouthshut.com using rule-based approaches. In (Sapna Negi et al., 2017); they attempted to detect the suggestion expression sentences in customer reviews using linguistic features, n-gram features, and POS tag information. In addition, the authors considered the hotel and electronics reviews from trip advisor and yelp reviews. Until 2016, the formal definition of suggestion mining was missing in the literature. Negi, S. (Sapna Negi et al., 2018)formulated a problem definition and made the labeled datasets available by collecting data across the various hotel and electronic reviews, Microsoft tweets, and software forum discussions.

In, Sapna Negi et al(Sapna Negi et al., 2018; Sapna Negi & Buitelaar, 2015). They have collected additional reviews from travel portals and Twitter with suggestion hashtags. The authors attempted to classify the reviews as suggestions or non-suggestion using rule-based approaches and deep learning-based methods such as LSTM and CNN. LSTM with Glove embeddings of 50- and 100-dimension representation are used in deep learning approaches and reported that LSTM performed better. In (Sapna Negi et al., 2018), authors developed a hybrid system to identify the review sentences that carry the suggestion intention. The authors built a semi-supervised learning approach towards identifying the customer-to-customer suggestions from opinion reviews.

A pilot shared task has been organized as part of the Semantic Evaluation workshop in 2019 (SemEval-2019) (*2019 - SemEval-2019 Task 9 Suggestion Mining from Online Reviews and Forums*, 2019) by Sapna Negi and Paul Buitelaar. The shared task consisted of two subtasks. Subtask-A mainly focuses on evaluating the system with the same domain dataset, whereas in subtask-B, and evaluated on a different domain dataset(*2019 - SemEval-2019 Task 9 Suggestion Mining from Online Reviews and Forums*, 2019). In(Fatyanosa et al., 2019), used core machine learning approaches such as random forest, logistic regression, support vector machines, and Naïve Bayes and genetic algorithm-based methods and reported results. In (Alexandros Potamias et al., n.d.; Oostdijk & Van Halteren, n.d.), rule-based handcrafted features are used as input to Bi-LSTM models and reported results. Most of the other submissions under these two subtasks focused on transfer learning approaches such as BERT(J. Liu et al., 2019; Park et al., 2019; Yamamoto & Sekiya, 2019; Yue et al., 2019; Zhou et al., 2019), ULTFiT(Anand et al., 2019), and other deep neural network models. The BERT-based models stood in top positions in the subtasks of suggestion mining. However, due to the transfer learning approaches and black box nature of work, it is not easy to understand the inherent learning and evaluate the system. Considering suggestion mining as a binary classification problem, all the models built so far used binary cross-entropy standard loss function(*2019 - MIDAS at SemEval-2019 Task 9 Suggestion Mining from Online Reviews*, 2019; Park et al., 2019; Prasanna & Seelan, 2019; Yamamoto & Sekiya, 2019). Therefore, to deal with the imbalanced dataset and improve the model's performance, the following research objectives are defined in this paper.

- We are building various deep learning models combined with several types of pre-trained word embeddings.
- The weighted focal loss function is proposed to overcome the class imbalance problem for suggestion mining.
- We utilize the weighted focal loss function to train the deep learning models for suggestion mining.

3.0 Methodology

The methodology for the suggestion mining tasks started with linguistic, rule-based, and manually handcrafted featuring approaches. The traditional supervised machine learning method, support vector machines (SVM) popularly used in this category with manually crafted features. The deep neural network models have slowly shown better results(Sapna Negi and Buitelaar 2017; Pecar, Simko, and Bielikova 2019) and transfer learning models(Anand et al., 2019; Yue et al., 2019) popularly experimented in submission to SemEval 2019. Though the advanced mechanisms are used

and presented in the literature, the research gaps identified are the nature of various word embeddings' performance not evaluated with a combination of kinds of neural networks. In this paper, we try to bridge the gap identified in the suggestion mining task and improve the models' performance with the help of a novel weighted focal loss approach for computing the error during training.

3.1 Datasets

The dataset shared by shared task organizers in SemEval -2019(*2019 - SemEval-2019 Task 9 Suggestion Mining from Online Reviews and Forums*, 2019) has been collected from various social platforms like third-party web portals and e-commerce sites related to multiple domain categories for suggestion mining tasks(Sapna Negi et al., 2018). For example, for hotel reviews, the data was collected from TripAdvisor.com. Electronics reviews data collected from Amazon.com. Travel forum the data collected from InsightVacations and Fodors. Microsoft tweets dataset collected from Twitter using Twitter API with keywords Windows Phone 7. Complete details of the datasets are summarized in Table-1. For the implementation, we have considered the dataset shared by SemEval-2019 organizers.

Table 1: Summary of Suggestion Mining datasets

S.No	Dataset Name	Authors information	Review category or dataset information	Dataset Size
1	Tweet Dataset about Microsoft phones	Dong et al., 2013	Twitter dataset about windows phones about product improvement with keyword search Windows Phone 7.	3000
2	Travel Advice dataset	Wicaksono and Myaeng, 2013	Review sentences from discussion threads, which are labeled as advice.	5199
3	Electronic and hotel review dataset	Negi and Buitelaar, 2015b	Prepared from social networks, the sentences which convey suggestions to the fellow customers	7534
4	Travel advice -2 dataset, Suggestion Forum	Negi and Buitelaar, 2015	Customer posts have been crawled and labeled a subset of them as suggestion mining tasks.	5724
5	Tweets with hashtags, suggestion, advice, recommendation	Negi and Buitelaar, 2015	Tweet dataset with various hashtags	4099

3.2 Word Embeddings

Word representation has been a vital component in all kinds of NLP tasks such as text classification, question-answering, chatbots, information extraction, and more. Conventionally, in most approaches, words are represented using the one-hot technique; due to the simple nature of use. Nevertheless, it does not capture any semantically related information, and the vector becomes sparse. Although the other popular strategies, such as TF-IDF and word co-occurrence methods, are used in the literature, these methods suffer from sparsity. The dense representation of words had become a critical research task and made remarkable progress and better results for all NLP tasks. Word2vec(Mikolov, Sutskever, et al., 2013; Rong, 2014), Glove(Pennington et al., 2019), and FastText(Bojanowski et al., 2017) are the popularly used industry-standard dense representation of word vectors.

Table 2: Summary of Word Embeddings

S.No	Word Embedding Approach	The primary idea in the model	Embedding Dimensions available	The dataset / Corpus used for training / Vocabulary size	Captures sub-word information	Captures polysemy words	Learning Strategy / Neural Network Model used
1	Word2Vec [2013](Bojanowski et al., 2017)	Capturing distributed semantics, predicting the target word	300-D	3 million words/ Google News dataset	No	Cannot capture the semantics of polysemy	The shallow neural network to learn embeddings
2	Glove [2014](Pennington et al., n.d.)	Word-word co-occurrence matrix	50-D, 100-D, 200-D, 300-D	Wikipedia-2014, Common crawl / 840 B / 2.2 M vocab	No	Cannot capture the semantics of polysemy	The shallow neural network to learn embeddings
3	FastText [2016](Bojanowski et al., 2017)	Character n-grams	300-D		Yes	Cannot capture the semantics of polysemy	The shallow neural network to learn embeddings

Word2vec is one of the most popular word embedding learning techniques designed with a two-layered, fully connected neural network(Bojanowski et al., 2017). It learns the vector representation by taking a large corpus of text as input and producing vector space of hundreds of dimensions. The vector representations learned by word2vec models have been shown to carry semantic meanings and are helpful in various NLP downstream tasks. The vectors in the Word2Vec model are obtained using a continuous bag of words (CBOW) or a skip-gram model. The above table presents the summary of various word embedding approaches popularly used.

The continuous bag of words (CBOW) model predicts the center word based on the context. The model trained to update the model parameters iteratively through the context-target word pairs generated from a training corpus. On the other hand, in the skip-gram model, given the center word, predict the surrounding words(Mikolov, Sutskever, et al., 2013; Rong, 2014). Global Vectors for representation (Glove)(Pennington et al., 2014) is an unsupervised learning approach that generates word embedding. The resulting embeddings show interesting linear substructures of sthe word in vector space. The glove method outputs vector space with meaningful substructures by training only on non-zero elements in a word-word global co-occurrence matrix. FastText(Bojanowski et al., 2017) is the word embedding model that learns the vector representation by leveraging the word's internal structure. This learning approach captures different morphological forms of words, the meaning of shorter words with the help of prefixes and suffixes.

3.3 Neural Network Architectures

Neural network-based models give promising results across all the NLP tasks such as sentiment analysis, question answering, machine translation, and chatbots. Neural network architectures can automatically extract features from the data and produce state-of-the-art results(Lecun et al., 2015). For example, if we have input data of speech, text, and video. Based on the properties, text data can be treated as a sequence. The best-suited networks for the sequence or temporal data are recurrent structured. Therefore, the recurrent neural networks

and variants of recurrent neural networks such as Long Short-Term Memory networks (LSTM) and Gated Recurrent Units (GRU)(Mohammadi & Shaverizade, 2021; Xing et al., 2019) are used for most (NLP) applications(Jing, 2019; G. Liu & Guo, 2019; Xue & Li, 2018).

Recurrent Neural Networks (RNN) is a famous network architecture for sequence prediction tasks. RNNs are very powerful and use various sequence learning and language modeling problems(Olah, 2015). The powerfulness of RNN comes with distributed hidden state, which allows storing much information about the past and updating its hidden state. The recurrent structure in RNN by connecting the output of the previous time-stamp as input makes it suitable for processing sequential information more effectively. The hidden states such as h_0, h_1, h_2, ……. h_t are the hidden state information passed as input to RNN cell and x_0, x_1, x_2, x_3…….. x_t to predict the output y_1, y_2, ……. y_t. The weights of the network are updated for each time step with backpropagation.

The Long Short-Term Memory networks (LSTM)s are variants of recurrent neural networks. LSTM is proposed in the literature to overcome traditional RNN such as vanishing gradient and exploding gradient(Olah, 2015). The LSTM architecture consists of recurrently connected sub-networks known to be memory blocks. This sub-network comprises three gates internally to implement and achieve long-term dependency. The gates are Input Gate, Forget Gate, and Output Gate. In LSTM, the first step is to decide what all information should be forgotten from the previous cell state. A Sigmoid activation function is used to implement forget-gate. The forget gate takes h_{t-1}, $x_{t,}$ and outputs value [0,1], identifying the quantity to remember or forget. The input gate helps in adding how much new information should be added to the cell state. Sigmoid and tanh activation functions can achieve this. The output gate helps in generating the modified version of the cell state. In output-gate, the sigmoid and tanh activation functions help alter specific parts of input and scale the cell state(Olah, 2015).

Gated Recurrent Unit (GRU) is the recent variant of recurrent neural network and modified version of LSTM with reduced gates. The Gated Recurrent Unit network consists of the update gate and the reset gate. GRU combines the functionality of the forget-gate and input gate of LSTM into a single update gate. As a result, GRU is computationally more efficient compared with LSTM.

The uni-directional nature of recurrent networks does not capture the future information in the process of learning. Therefore, a bi-directional variant of LSTM (Bi-LSTM) and GRU (Bi-GRU) has been used extensively in the literature for most NLP tasks to capture future and past information in the input. The bi-directional variant model consists of two LSTMs or GRUs, one taking input in the forward direction and the other in the backward direction. With bi-directional input availability, the overall context available to the algorithm gets increased. As a result, the model's performance has boosted up and reported better results for various NLP tasks.

Convolutional Neural Networks (CNN) are popular architectures for processing data in multiple arrays. CNN models are produced state of the art results in all computer vision tasks. The typical architecture of the CNN model to process or learn from image data consists of majorly two types of layers, such as convolutional layers and pooling layers. The convolutional neural networks have also been used to process sequence data such as text(Z. Liu et al., 2020)(Kim, 2014). For example, in the sequence learning task, given a sequence of words w_1, w_2, w_3,…… w_n, where each word has been associated with an embedding vector of a specific dimension. A 1-D convolution of width-k results in moving a sliding window of size k over the vector and applying a convolution filter to each window in the sequence. The operations are a dot product between the concatenation of the embedding vectors in the sliding window and a weight vector *u*, followed by a nonlinear activation function.

For suggestion mining, a multi-channeled CNN- LSTM (Z. Liu et al., 2020) consists of multiple versions of the standard CNN and LSTM models parallel with different embedding initialization. This allows processing each review with additional features at the same time. In this paper, for the multi-channeled model, we experimented with CNN-LSTM layers along with word2vec, Glove, and FastText embeddings with 300 dimensions. The following table demonstrates the summary of various neural network architectures

Table 3: Summary of Neural Network Architectures

S.No	Neural Network Models	Major Feature of Model	Kinds of applications built using the model	Merits	Demerits
1	Recurrent Neural Network (RNN)	Captures patterns from Sequences	Sentiment Analysis (SA), Text classification, Text summarization, Machine Translation	More suitable for sequential data, stores previous time-stamps computations	Suffers from Vanishing gradient problem
2	Long Short-Term Memory Network (LSTM)	Gating Mechanism to capture the long-term dependency	Analysis (SA), Text classification, Text summarization, Machine Translation	Pay attention to the selected sequence of words and store past computations	Cannot capture the future information and computationally heavy
3	Gated Recurrent Unit (GRU)	The fewer number of gates, to capture long term dependency	Analysis (SA), Text classification, Text summarization, Machine Translation	Pay attention to the selected sequence of words and store past computations	No Memory and computationally heavy
4	Bi-Directional LSTM	Gating Mechanism to capture the long-term dependency	Analysis (SA), Text classification, Text summarization, Machine Translation	Captures the information from both the directions	Computationally heavier
5	Bi-Directional GRU	The fewer number of gates, to capture long term dependency	Analysis (SA), Text classification, Text summarization, Machine Translation	Captures the information from both the directions	No Memory and computationally heavy
6	Convolutional Neural Network (CNN)	Captures features from any parts of words	Text classification	Fast and computationally less expensive, compared with other models	Mis-spelled words can be learned
7	Multi-Channel Neural Network	Hybrid model, integrate LSTM and CNN models	Text Classification	Captures features from CNN and LSTM, both methods	Computationally heavy

3.4 Weighted focal loss function

Loss functions measure how far an estimated value is from its actual value. For example, binary cross-entropy is the most commonly used loss function for binary text classification applications(Do et al., 2019)(Huang et al., 2019)(Golchha et al., 2018). In this, it compares each of the predicted probabilities with the actual output and calculates the score that penalizes the probabilities based on the distance from the expected value. In

addition, the focal loss function(Mukhoti et al., 2020; Pasupa et al., 2020; Sun et al., 2019) has been used to deal with models trained with imbalanced datasets. A weighted focal loss function is defined by considering the factor of focal loss and binary cross-entropy loss. For the predicted and actual output values, both the loss values are computed and used to optimize model learning.

Loss_1 = Focal Loss (P_t)=$-(1-P_t)^{\gamma} \log(P_t)$ ---------- (1)

Loss_2 = Binary Cross Entropy (BCE) = $-\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(y_i^{prob}) + (1-y_i)\log(1-y_i)\right]$ ---- (2)

Weighted Focal Loss = α * Loss_1 + (1- α) * Loss_2 ---------- (3)

Where α is the loss weightage parameter, if the dataset is more- imbalanced, more weightage is given for Focal Loss with α = 0.8. On the other hand, if the dataset is balanced, equal weightage is given for both loss functions as α = 0.5. All the experiments were conducted using α = 0.8. Multiple experiments were undertaken to fix the value of α, ranging from 0.2 to 0.9, for the suggestion mining task due to the class imbalance, at the importance of 0.8 it is giving promising results.

4 Evaluation Measures, Results, and Discussion

4.1 Evaluation Metrics

Suggestion mining is considered a binary text classification task; the performance of any binary classifier can be evaluated in terms of Precision, Recall, Accuracy, and F-score. The binary classifier for suggestion mining classifies all the data instances, either suggestion or non-suggestion. The classifier produces four different kinds of outcomes, in which two are correct classification or true, two are incorrect classification or false. The correct category is True Positives and True Negatives, and the wrong type is False Positives and False Negatives. Precision can be defined as the number of true positives divided by total instances labeled as the positive class (True positives or False Positives). Recall can be defined as the number of true positives divided by the number of cases belonging to the positive class (True positives and False negatives). Accuracy is another widely used measure for classification performance, and it is defined as the ratio between the correctly classified instances to the total number of cases. An effort that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

4.2 Results and Discussion

The experimental results of various models combined with several word embedding initialization methods and loss functions are summarized in Table-4. Firstly, four different word vector initialization approaches, Random initialization, Word2Vec, Glove, and FastText, combined with two loss functions such as binary cross-entropy and weighted focal loss function and seven neural network architectures modeled. We notice that the LSTM model with any embedding method performs the least out of all the models implemented for the task. Based on the observation, we claim that the sequential features captured by the LSTM model are not well suited for the Suggestion Mining task. The performance of all the models was evaluated using metrics such as accuracy, precision, recall, and f1-score. Tables 4(a)(b)(c)(d) depict the accuracy, precision, recall, and f1-score, respectively, with two different loss functions for suggestion mining. Being suggestion mining dataset is an imbalanced dataset and biased towards the non-suggestion class. Furthermore, the traditionally used binary cross-entropy loss function could not compute the error propagated in the optimization process. The modified version of the focal loss function combination with binary cross-entropy performed better in the majority of the cases for classifying the opinion reviews into suggestion and non-suggestion classes.

As per the accuracy measure, CNN and multi-layered LSTM perform better without considering embedding or loss functions. On the other hand, if we consider word embedding initialization, Glove and FastText have very similar performances with a bit of difference. In consideration of loss functions, the weighted loss function proposed in this paper performs better for a few instances with Glove and FastText embedding initialization. With precision as a measure, most models give better results combined with Word2Vec, Glove, and FastText with the proposed weighted focal loss function. With Recall and F1-scores also as measures, the proposed weighted loss function was performing better for the majority of the cases. Thus, we can summarise that CNN, Multi-layered LSTM, and Multi-channeled model with Glove and FastText embedding initialization with proposed weighted focal loss function gives better results for suggestion mining tasks.

The 1(a)(b) graphs show the accuracy measure with two different loss functions for the six different neural network architectures with four-word vector initialization methods. The interpretation from the chart is that the CNN with FastText outperformed. Chart 1(c)(d) depicts the precision measure for the two-loss functions and seven neural network architecture with several word embeddings initializations. Again, the results show that the CNN model with Word2Vec and FastText are giving better results. Next, 2(a) and (b) depict recall measurements of the same neural network models and word embeddings with two loss functions. Finally, 2(c) and (d) depict the f1-score of the models with the same neural networks, word embedding initialization, and loss functions. At the outset, the CNN model with Glove and FastText performing better.

5.0 Conclusion and Future Scope

This paper presented the effect of various word embeddings combined with multiple neural network architectures with two different loss functions for Suggestion mining. First, pre-trained embeddings such as Word2Vec, Glove, and Fast text are initialized for the embedding layer of seven neural network models: ANN, CNN, LSTM, GRU, Bi-LSTM, and multi-channel network. Second, a weighted focal loss is proposed to make the model to learn better from the data, optimize the learning process, and deal with class imbalance problems. In all the models concerning the performance evaluation, the proposed weighted focal loss performs better in combination with CNN, Multilayer LSTM with initialization of Glove, and FastText embeddings. In continuation to this, we would like to experiment with attention models contextual embedding behavior for the same task. We would also like to experiment with transformer-based, encoder-decoder models and information extraction approaches to extract aspects and identify the suggestions towards specific elements.

Table 4(a): Performance of different neural networks with embeddings and Loss functions - Accuracy

Model /Embedding	WOEmb	Word2Vec	Glove	FastText	WOEmb	Word2Vec	Glove	FastText
	With binary_cross_entropy_loss				Weighted_focal_loss [our loss]			
	Accuracy				Accuracy			
ANN	0.837	**0.845**	0.834	0.842	0.831	0.832	0.832	**0.841**
CNN	0.873	0.874	0.882	**0.887**	0.863	**0.882**	0.88	0.847
LSTM	0.787	0.752	**0.837**	0.805	0.792	0.778	**0.852**	0.801
Multi-layer LSTM	0.847	0.836	**0.878**	0.855	0.857	0.862	0.864	**0.861**
Bi-GRU	0.859	**0.872**	0.87	0.865	**0.864**	0.863	0.859	0.857
Bi-LSTM	0.854	0.851	**0.874**	0.858	0.852	0.843	0.858	**0.862**
Multi-Channel	0.841	0.851	0.854	**0.863**	0.843	0.849	**0.855**	0.86

Table 4(b): Performance of different neural networks with embeddings and Loss functions - Precision

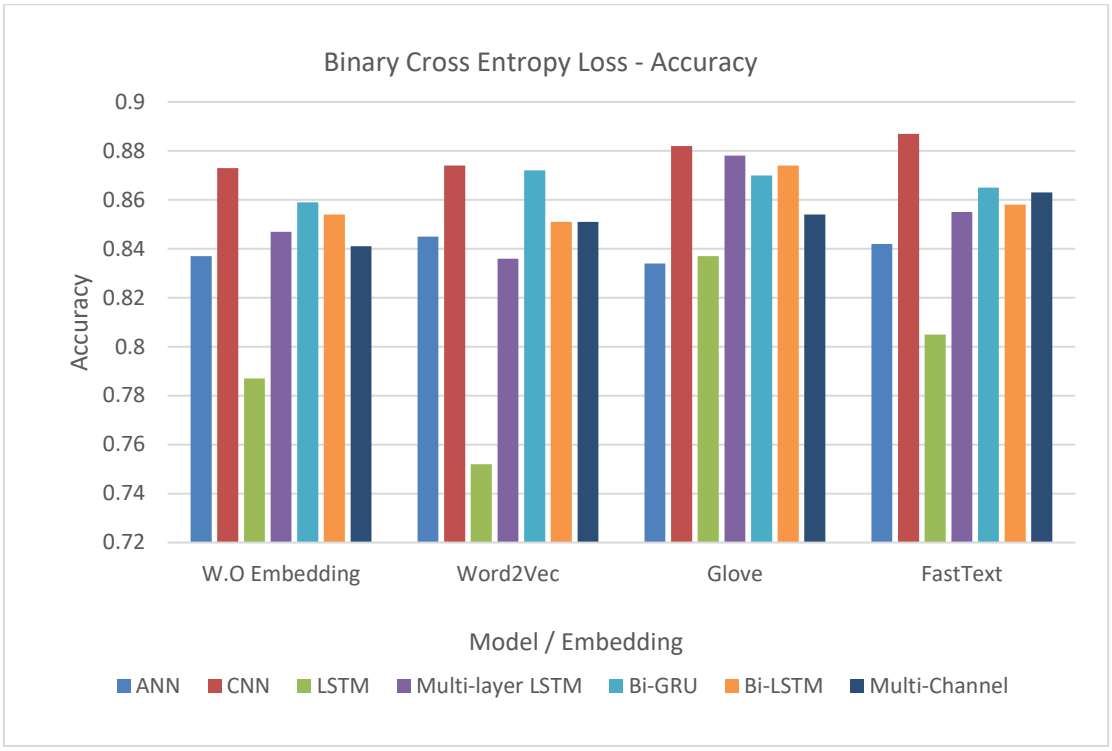
Model /Embedding	WOEmb	Word2Vec	Glove	FastText	WOEmb	Word2Vec	Glove	FastText
	With binary_cross_entropy_loss				Weighted_focal_loss [our loss]			
	Precision				Precision			
ANN	0.73	0.714	0.713	**0.767**	0.72	0.73	**0.749**	0.676
CNN	0.792	**0.854**	0.835	0.819	0.815	0.808	0.76	0.816
LSTM	0.583	0.505	**0.714**	0.604	0.562	0.51	**0.725**	0.612
Multi-layer LSTM	0.754	0.64	**0.786**	0.745	0.787	0.76	0.758	**0.801**
Bi-GRU	0.749	**0.808**	0.769	0.789	**0.786**	0.768	0.772	0.777
Bi-LSTM	0.734	0.743	0.776	**0.778**	0.584	0.745	0.759	**0.769**
Multi-Channel	0.731	0.715	0.717	**0.807**	0.745	0.722	0.724	**0.817**

Table 4(c): Performance of different neural networks with embeddings and Loss functions -- recall

Model /Embedding	WOEmb	Word2Vec	Glove	FastText	WOEmb	Word2Vec	Glove	FastText
	With binary_cross_entropy_loss				Weighted_focal_loss [our loss]			
	Recall				Recall			
ANN	0.549	**0.636**	0.563	0.528	0.518	0.523	0.492	**0.7**
CNN	0.667	0.596	0.66	**0.705**	0.584	0.695	**0.761**	0.502
LSTM	0.511	0.422	0.577	**0.634**	0.523	0.452	0.564	**0.623**
Multi-layer LSTM	0.573	**0.785**	0.702	0.641	0.584	0.658	**0.667**	0.591
Bi-GRU	0.655	0.639	**0.686**	0.629	0.625	**0.648**	0.617	0.601
Bi-LSTM	0.65	0.615	**0.695**	0.606	0.599	0.566	0.634	**0.639**
Multi-Channel	0.573	0.669	**0.683**	0.594	0.554	0.672	**0.685**	0.574

Table 4(d): Performance of different neural networks with embeddings and Loss functions – F1-Score

Model /Embedding	WOEmb	Word2Vec	Glove	FastText	WOEmb	Word2Vec	Glove	FastText
	With binary_cross_entropy_loss				Weighted_focal_loss [our loss]			
	F1-Score				F1-Score			
ANN	0.627	0.673	0.627	**0.673**	0.627	0.673	0.627	**0.673**
CNN	0.724	0.702	**0.724**	0.702	0.724	0.702	**0.724**	0.702
LSTM	0.545	0.46	**0.545**	0.46	0.545	0.46	**0.545**	0.46
Multi-layer LSTM	0.651	0.705	0.651	**0.705**	0.651	0.705	0.651	**0.705**
Bi-GRU	0.699	0.714	0.699	**0.714**	0.699	0.714	0.699	**0.714**
Bi-LSTM	0.69	0.673	**0.69**	0.673	0.69	0.673	**0.69**	0.673
Multi-Channel	0.642	0.691	0.642	**0.691**	0.642	0.691	0.642	**0.691**



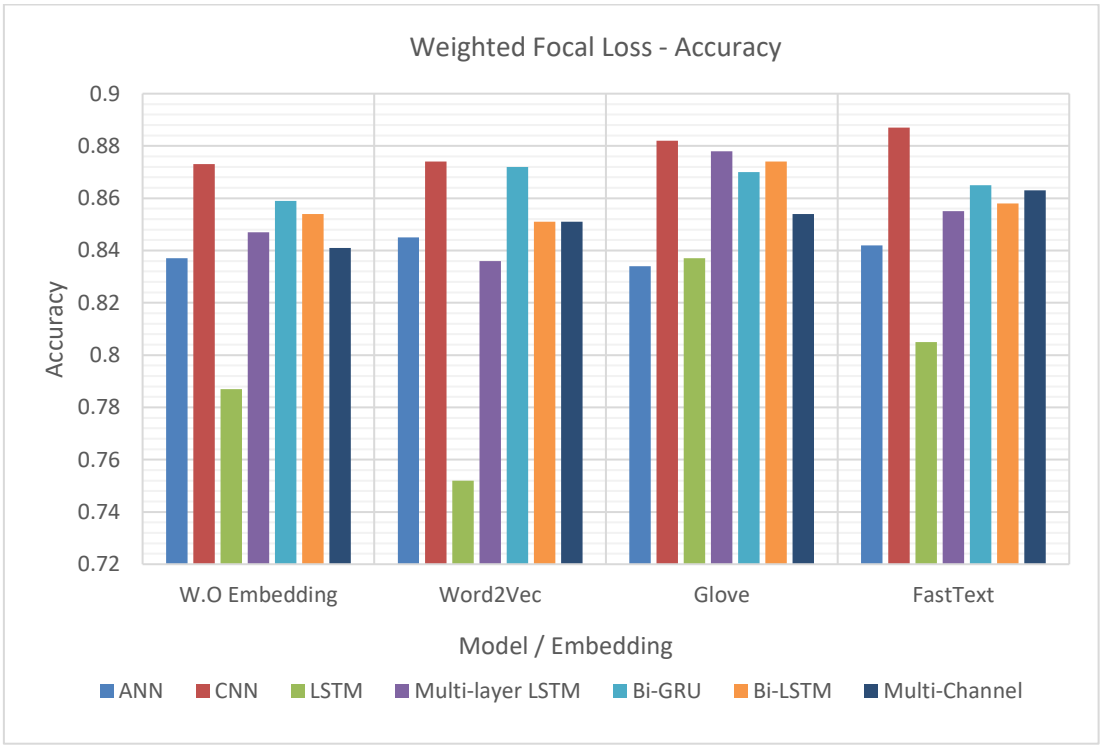
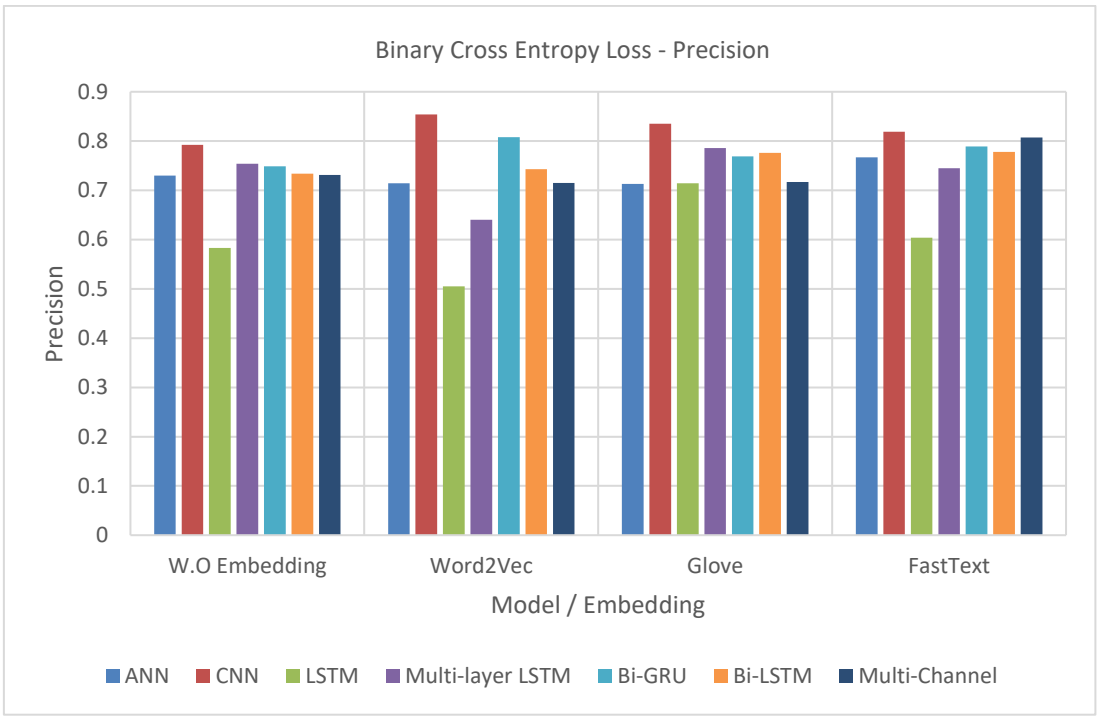


Figure 1(a)(b) - Accuracy Measure



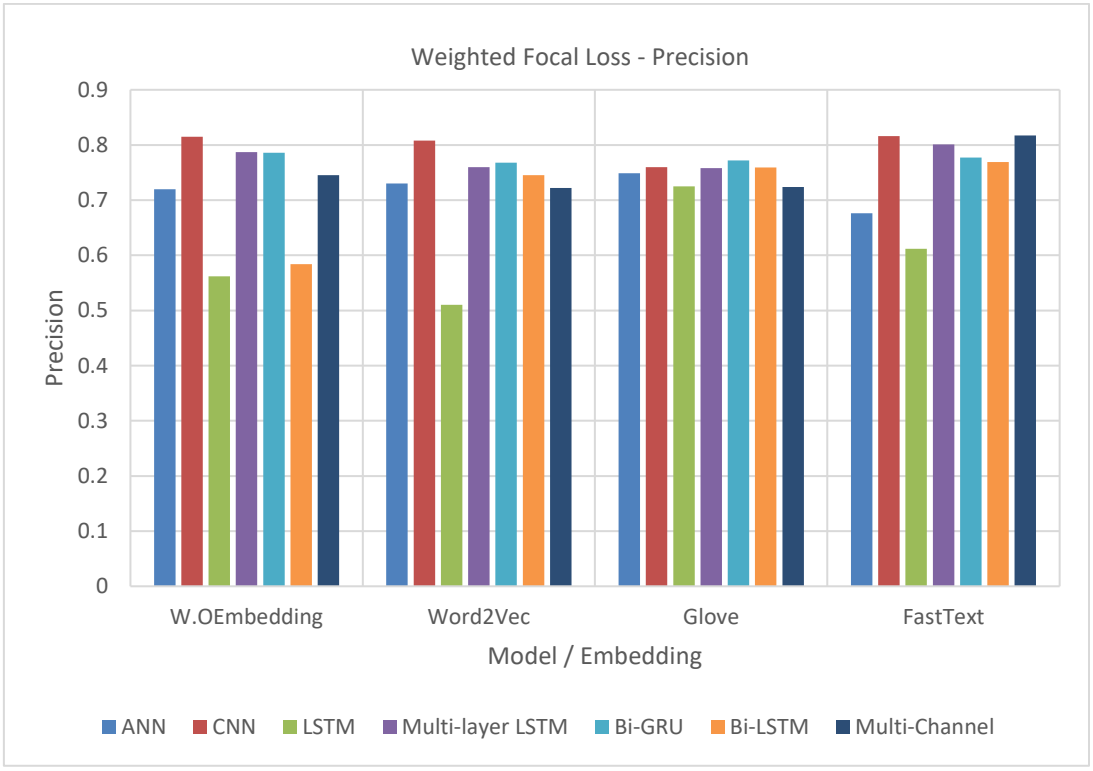


Figure 1 (c)(d) : Precision Measure

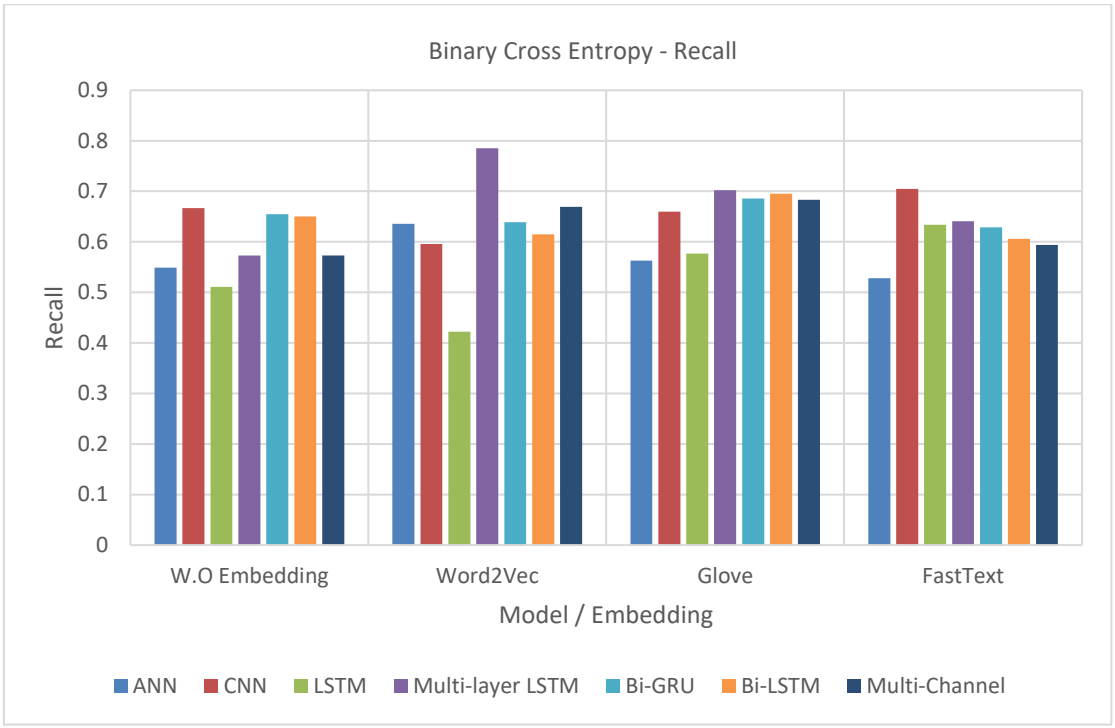


Figure 2 (a)(b) - Recall Measure

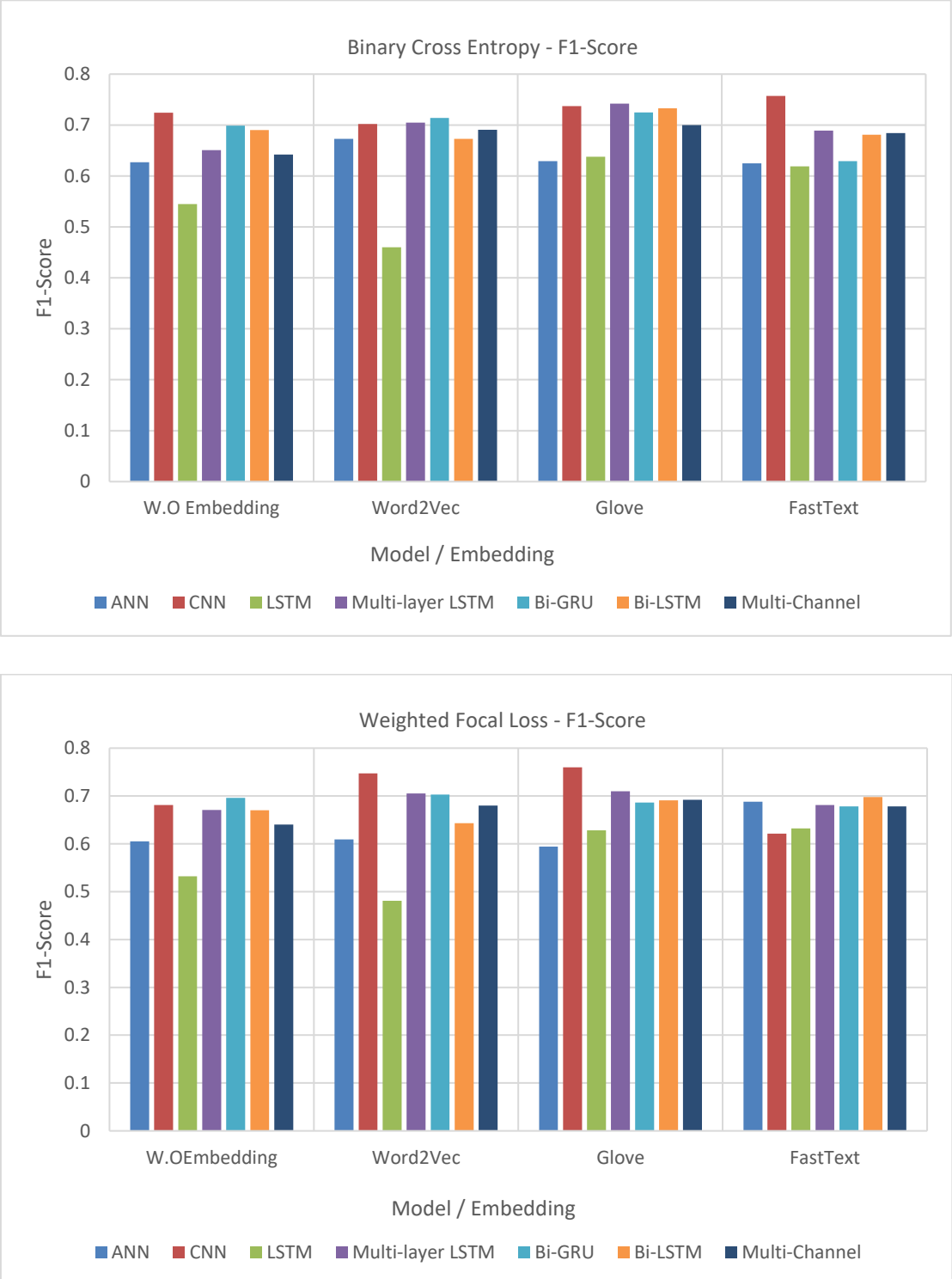


Figure 2 (c)(d) – F1-Score Measure

6.0 References

2019 - MIDAS at SemEval-2019 Task 9 Suggestion Mining from Online Reviews. (2019).

2019 - SemEval-2019 Task 9 Suggestion Mining from Online Reviews and Forums. (2019).

Alexandros Potamias, R., Neofytou, A., & Siolas, G. (2019). *NTUA-ISLab at SemEval-2019 Task 9: Mining Suggestions in the wild*. https://www.tripadvisor.com.gr/

Almeida, F., & Xex, G. (2015). *Word Embeddings: A Survey*. *1991*.

Anand, S., Mahata, D., Aggarwal, K., Mehnaz, L., Shahid, S., Zhang, H., Kumar, Y., Shah, R. R., & Uppal, K. (2019). *Suggestion Mining from Online Reviews using ULMFiT*. http://arxiv.org/abs/1904.09076

Baby, P., & B, K. (2020). Sentimental Analysis and Deep Learning : A Survey. *International Journal of Scientific Research in Science, Engineering and Technology*, 212–220. https://doi.org/10.32628/ijsrset207135

Bengio, Y., Goodfellow, I. J., & Courville, A. (2015). Sequence Modeling : Recurrent and Recursive Nets. *Deep Learning*, 324–365.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Do, H. H., Prasad, P. W. C., Maag, A., & Alsadoon, A. (2019). Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review. *Expert Systems with Applications*, *118*, 272–299. https://doi.org/10.1016/j.eswa.2018.10.003

Fatyanosa, T. N., Hafiz, A., Siagian, A. M., & Aritsugi, M. (2019). *DBMS-KU at SemEval-2019 Task 9: Exploring Machine Learning Approaches in Classifying Text as Suggestion or Non-Suggestion*. https://seaborn.pydata.org/generated/seaborn.heatmap.html

Golchha, H., Gupta, D., Ekbal, A., & Bhattacharyya, P. (2018). *Helping each Other: A Framework for Customer-to-Customer Suggestion Mining using a Semi-supervised Deep Neural Network*. http://arxiv.org/abs/1811.00379

Huang, W., Chen, E., Liu, Q., Chen, Y., Huang, Z., Liu, Y., Zhao, Z., Zhang, D., & Wang, S. (2019). Hierarchical multi-label text classification: An attention-based recurrent network approach. *International Conference on Information and Knowledge Management, Proceedings*, 1051–1060. https://doi.org/10.1145/3357384.3357885

Jing, R. (2019). A Self-attention Based LSTM Network for Text Classification. *Journal of Physics: Conference Series*, *1207*(1), 012008. https://doi.org/10.1088/1742-6596/1207/1/012008

Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the*

Conference, 1746–1751. https://doi.org/10.3115/v1/d14-1181

Krishna, A., Akhilesh, V., Aich, A., & Hegde, C. (2019). Sentiment Analysis of Restaurant Reviews Using Machine Learning Techniques. In *Lecture Notes in Electrical Engineering* (Vol. 545). Springer Singapore. https://doi.org/10.1007/978-981-13-5802-9_60

Laskari, N. K., & Sanampudi, S. K. (2019). *Aspect Based Sentiment Analysis Survey*. *18*(2), 24–28. https://doi.org/10.9790/0661-18212428

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lin, Y., Wang, X., & Zhou, A. (2016). Opinion spam detection. In *Opinion Analysis for Online Reviews* (Issue May, pp. 79–94). WORLD SCIENTIFIC. https://doi.org/10.1142/9789813100459_0007

Liu, B. (2016). *Aspect based sentiment analysis*. *8*, 72–75. http://link.springer.com/10.1007/978-3-642-19460-3

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, *337*, 325–338. https://doi.org/10.1016/j.neucom.2019.01.078

Liu, J., Wang, S., & Sun, Y. (n.d.). *OleNet at SemEval-2019 Task 9: BERT based Multi-Perspective Models for Suggestion Mining*.

Liu, Z., Huang, H., Lu, C., & Lyu, S. (2020). *Multi-channel CNN with Attention for Text Classification*. http://arxiv.org/abs/2006.16174

Markov, I., & De La Clergerie, E. V. (2019). *INRIA at SemEval-2019 Task 9: Suggestion Mining Using SVM with Handcrafted Features*. https://www.uservoice.com

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 1–9.

Mohammadi, A., & Shaverizade, A. (2021). Ensemble deep learning for aspect-based sentiment analysis. *International Journal of Nonlinear Analysis and Applications*, *12*(Special Issue), 29–38. https://doi.org/10.22075/IJNAA.2021.4769

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H. S., & Dokania, P. K. (2020). Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, *2020-Decem*(NeurIPS).

Negi, S., & Buitelaar, P. (2017). Suggestion Mining From Opinionated Text. In *Sentiment Analysis in Social Networks* (pp. 129–139). Elsevier Inc. https://doi.org/10.1016/B978-0-12-804412-4.00008-5

Negi, Sapna, Asooja, K., Mehrotra, S., & Buitelaar, P. (2018). *A Study of Suggestions in Opinionated Texts and their Automatic Detection*. https://feedly.uservoice.com/forums/192636-

Negi, Sapna, & Buitelaar, P. (2015). Towards the extraction of customer-to-customer suggestions from reviews. In *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://doi.org/10.18653/v1/d15-1258

Negi, Sapna, de Rijke, M., & Buitelaar, P. (2018). *Open Domain Suggestion Mining: Problem Definition and Datasets*. http://arxiv.org/abs/1806.02179

Olah, C. (2015). *No Title*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Oostdijk, N., & Van Halteren, H. (n.d.). *Team Taurus at SemEval-2019 Task 9: Expert-informed pattern recognition for suggestion mining*.

Park, C., Kim, J., Lee, H., Amplayo, R. K., Kim, H., Seo, J., & Lee, C. (2019). *ThisIsCompetition at SemEval-2019 Task 9: BERT is unstable for out-of-domain samples*. http://arxiv.org/abs/1904.03339

Pasupa, K., Vatathanavaro, S., & Tungjitnob, S. (2020). Convolutional neural networks based focal loss for class imbalance problem: a case study of canine red blood cells morphology classification. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-01773-x

Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe: Global Vectors for Word Representation*. Retrieved June 27, 2020, from http://nlp.

Prasanna, S., & Seelan, S. A. (2019). *Zoho at SemEval-2019 Task 9: Semi-supervised Domain Adaptation using Tri-training for Suggestion Mining*. http://arxiv.org/abs/1902.10623

Rong, X. (2014). *word2vec Parameter Learning Explained*. 1–21. http://arxiv.org/abs/1411.2738

Sun, X., Dong, K., Ma, L., Sutcliffe, R., He, F., Chen, S., & Feng, J. (2019). Drug-drug interaction extraction via recurrent hybrid convolutional neural networks with an improved focal loss. *Entropy*, *21*(1). https://doi.org/10.3390/e21010037

Viswanathan, A., Venkatesh, P., Vasudevan, B., Balakrishnan, R., & Shastri, L. (2011). Suggestion mining from customer reviews. *17th Americas Conference on Information Systems 2011, AMCIS 2011*, *2*, 1351–1359.

Xia, W., Zhu, W., Liao, B., Chen, M., Cai, L., & Huang, L. (2018). Novel architecture for long short-term memory used in question classification. *Neurocomputing*, *299*, 20–31.

https://doi.org/10.1016/j.neucom.2018.03.020

Xing, S., Liu, F., Wang, Q., Zhao, X., & Li, T. (2019). A hierarchical attention model for rating prediction by leveraging user and product reviews. *Neurocomputing*, *332*(xxxx), 417–427. https://doi.org/10.1016/j.neucom.2018.12.027

Xue, W., & Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 2514–2523. https://doi.org/10.18653/v1/p18-1234

Yamamoto, M., & Sekiya, T. (2019). *m y at SemEval-2019 Task 9: Exploring BERT for Suggestion Mining*. https://competitions.codalab.org/competitions/19955

Yu, K., Li, H., & Oguz, B. (2019). *Multilingual Seq2seq Training with Similarity Loss for Cross-Lingual Document Classification*. *2*, 175–179. https://doi.org/10.18653/v1/w18-3023

Yue, P., Wang, J., & Zhang, X. (2019). *YNU-HPCC at SemEval-2019 Task 9: Using a BERT and CNN-BiLSTM-GRU Model for Suggestion Mining*.

Zhou, Q., Zhang, Z., Wu, H., & Wang, L. (2019). *ZQM at SemEval-2019 Task9: A Single Layer CNN Based on Pre-trained Model for Suggestion Mining*. https://github.com/google-research/bert