An approach for suggestion mining based on deep learning techniques

T Raghunadha Reddy¹, P Vijayapal Reddy², T Murali Mohan³, Raju Dara⁴

1Associate Professor, Dept. of CSE, Matrusri Engineering College, Saidabad, Hyderabad, Telangana
2Professor, Dept. of CSE, Matrusri Engineering College, Saidabad, Hyderabad, Telangana
3Professor of CSE, Principal, Swarnandhra Institute of Engineering and Technology, Narsapuram, AP
4Professor, Dept. of CSE, Vignana Bharathi Institute of Technology, Ghatkesar, Telangana

raghu.sas@gmail.com¹, drpvijayapalreddy@gmail.com², drtmm512@gmail.com³, rajurdara@gmail.com⁴

Abstract. An organization often uses forums and social media channels for getting feedback from customers or users. The ratings of products on rating platforms are a useful feedback to make a product better. The feedback from a customer is in the form of a suggestion which appears in a rating text or is directly asked from the customer. Suggestion mining is a binary classification problem that labels sentences as Suggestion or Non-suggestion. The suggestion mining is similar to sentiment analysis which is associated with common linguistic properties and challenges irrespective of the domain and application. Most of the previous works of suggestion mining proposed rule based methods and a very few developed statistical classifiers by using manually identified features. Recently, several researchers paid attention on deep learning technique based solutions to suggestion mining where features are automatically learned. In this work, various deep learning techniques like RNN, LSTM, Attention based LSTM and GRU are used in the experimentation of suggestion mining. The experiment carried out on the dataset provided in SemEval 2019 suggestion mining competition. The Attention based LSTM achieved best accuracies for suggestion mining when compared with other deep learning techniques.

1. Introduction
Online text is becoming an increasingly popular source to acquire public opinions towards entities like persons, brands, products, services etc. Opinion mining is a research area which investigates the extraction of opinions towards a target entity from the text written by a set of people. The problem of Opinion mining was introduced in 2002, which dealt with the classification of a textual unit not with regard to its topic but with regard to the opinion expressed in the text. Opinion mining is primarily used to refer opinion classification of a text unit into positive, negative, objective or neutral categories. The opinionated text also contains descriptive information like advice, tips, wishes for improvements, suggestions for improvements, warnings and recommendations which are generally overlooked in a sentiment based summary, mostly because it does not contain any sentiment towards the target entity.
Suggestions present in the reviews convey ideas for improvements to the brand owners, or tips and advice to fellow customers about commercial products and services [1]. The suggestion mining is the extraction of suggestions from the source text where suggestions of interest are likely to appear. The task of suggestion mining is similar to sentiment analysis in that the input is the same. However, the output of a sentiment analysis model and a suggestion mining model is different. While sentiment classifiers focus on grouping reviews as positive or negative or neutral, suggestion mining models identify the reviews that have suggestions or non-suggestions to other customers or service providers.

Suggestion mining is used in various applications such as ideas for product improvement, Customer to customer suggestions, Suggestion summarisation, improving sentiment detection and Recommender systems [2]. Suggestions present among the opinions expressed on social media often convey ideas for improvements to a brand owner. These ideas are otherwise sought by the stakeholders through dedicated suggestion boxes or online forms. Reviews, blogs, and other opinionated text about commercial entities also carry tips, warnings, and advice for the future customers. Often future customers seek such advice through dedicated question answering forums, where they have to wait for a considerable amount of people to answer. Suggestion mining can also be employed for summarisation of dedicated forums or surveys used for collecting suggestions around a certain topic or entity. A suggestion mining system can identify the exact sentence where a suggestion is conveyed in a larger piece of text, and thus help in summarising these responses. A significant number of sentences in the benchmark sentiment analysis datasets are labelled as neutral or objective. Often sentiment analysis systems perform poorly in identifying sentiments of neutral and irrelevant sentences. Suggestion mining is useful for determining the sentences which carry suggestions with respect to the target entity. Recommender systems predict user preference by utilizing the content from a user profile and the entity profile to recommend the entities. Suggestion mining systems will be capable of extracting such recommendations, which can then be utilized by recommender systems.

In this work, the experiment carried out with various deep learning techniques such as RNN, LSTM, Attention based LSTM and GRU for suggestion mining. All the techniques used word2vec vectors as input. Accuracy measure is used to evaluate these models. The experiment performed on training dataset of SemEval 2019 suggestion mining challenge.

This work is planned in 6 sections. The existing work in suggestion mining is explained in section 2. The section 3 presents the dataset characteristics and evaluation measures. The section 4 discuss about different types of deep learning techniques which are used in this experimentation. The experimental results are presented in section 5. The section 6 concludes this paper with future scope.

2. Existing Work
Nowadays, a huge number of texts are posted in online reviews or discussion forums. Such media can be a valuable source for obtaining a suggestion about products or services. The obtained suggestion is not only useful for readers but also important information for stakeholders. Indeed, such advice can be used to improving the quality of products or giving helpful recommendations. However, identifying a suggestion from a lot of reviews or comments needs extra effort and time. Moreover, such online texts are mostly in unstructured form. Thus, automatically mining the suggestion from given texts is challenging and significant. Suggestion mining is relatively a new research interest in text classification tasks. Several studies have initiated to mining suggestions from online texts. However, they concluded that it is not easy to identify suggestion texts automatically. In other words, it still has room to improving the classification result in the suggestion mining task.

Masahiro Yamamoto et al., explored [3] the Bidirectional Encoder Representations from Transformers (BERT) to focus on the sub task A of suggestion mining where the training and test datasets from same domain in SemEval-2019 competition. These model parameters were then fine-tuned based on the labeled training dataset. Generally, the BERT training consists of two parts such as pre-training on the general domain corpus and fine-tuning on the target task. In their work, the proposed system consists of three training steps such as pre-training on the general domain corpus, pre-training on the target domain corpus, and fine-tuning on the target task. They further pre-train
BERT model using an unlabelled corpus related to domain of documents from a windows forum and run additional steps of pre-training using these documents. For the classification task, they added one layer to output predictions like suggestion or non-suggestion. They applied the idea of distant supervision for the sub task B of cross-domain suggestion mining where the training and test datasets are from different domains. Distant supervision is a weakly supervised learning framework which tries to automatically generate noisy training examples. Specifically, they used the rule based system which is provided by the task organizer for creating a noisy training dataset and train the model based on them. Proposed system significantly outperforms baseline methods on two subtasks. Their approach achieved the 3rd place in subtask A and the fifth place in Subtask B.

Tirana Noor Fatyanosa et al., explored [4] various machine learning algorithms such as Logistic Regression (LR), Random Forest (RF), Multinomial Naive Bayes (MNB), Sublinear Support Vector Classification (SSVC), Linear Support Vector Classification (L SVC), Variable Length Chromosome Genetic Algorithm-Naive Bayes (VLCGA-NB) and Convolutional Neural Network (CNN). Their system obtains reasonable results of F1-Score 0.47 and 0.37 on the evaluation data in Subtask A and Subtask B, respectively. They identified that the obtained result of all classification methods outperformed that of the baseline for the Non-suggestion class. MNB yielded the best results with 0.95. In terms of F1-Score of the Suggestion class, they found that RF, SSVC, and VLCGA-NB obtained a competitive result outperforming baseline for Suggestion class. The highest F1-Score was obtained by SSVC at 0.47. RF and VLCGA-NB produced F1-Score at 0.29 and 0.31, respectively. Overall, note that MNB and SSVC obtained the best F1-Score for Non-suggestion and Suggestion classes, respectively.

Tobias Cabanski created [5] a model for task A by using the combination of CNN and LSTM networks. This model used input features of BERT word embeddings. For subtask B, the idea is to extend the model from subtask A with a domain discriminator and shared features. Since that adds a lot of parameters to the model, the text classifier has a simpler structure than in subtask A and uses only CNNs for classification. In contrast to subtask A, no labelled data for supervised training is provided, so that additional unlabelled data is taken to apply a cross-domain classification. This is done by using adversarial training of the three model parts label classifier (Model that predicts if an example is a suggestion), domain classifier (Model for the prediction of the domain of an example) and the shared feature representation (Model that applies a transformation on the input features). For subtask A, the best model reached a F1-score of 0.7273 for the test data. This model archived a validation F1-score of 0.875 which is a noticeable difference to the test data score. The model for subtask B reached a final F1-score on the test data of 0.8187.

Aysu Ezen-Can et al., employed [6] a hybrid approach that utilizes a recurrent neural network (RNN) along with rule-based features to build a domain-independent suggestion mining model to identify suggestions within reviews. They used two sets of features such as rule-based and model-generated from word embeddings for suggestion mining. The rule-based features are extracted from the heuristics used in the baseline system for this challenge. The first rule-based feature is using a pattern matching algorithm based on regular expressions. The second rule-based feature utilizes keywords without any patterns such as ‘suggest’, and ‘recommend’. The third rule-based feature relies on part of speech tags. Recurrent neural networks require a mechanism to convert textual input to numerical vectors to be able to perform computations. They used pre-trained word embeddings where each word in the embedding table has a vector of size 100 and Glove embeddings which was trained on Wikipedia 2014 and Gigaword 5 corpora. RNN used a fully connected layer that takes the rule-based features and the review as the inputs. Then two bidirectional LSTM layers follow for modelling the textual input. Before the softmax layer, an LSTM layer takes the advantage of both learned representations form bidirectional layers and the rule based features. With the test dataset, the model obtained 74.49% F1 score where the majority baseline was 57.77%.

Ilia Markov et al., used [7] scikit-learn implementation of the Support Vector Machines (SVM) algorithm which is trained on handcrafted features, function words, sentiment features, digits, and verbs for Subtask A, and handcrafted features for Subtask B. Handcrafted features are manually
selected list of representative keywords and patterns of a suggestion from the training and development data. For Subtask A, we used a set of 57 handcrafted keywords and 77 keywords were used for Subtask B. They used term frequency weighting scheme. They observed that function words, sentiment features, digits, and verbs did not improve the performance of a system for Subtask B and also observed that handcrafted features and function words are the most indicative features in their system. Their system achieved a F1-score of 51.18% on Subtask A and 73.30% for Subtask B.

Junyi Li et al., combined [8] the attention mechanism with the LSTM model. They trained a sequence of neural network models such as long short-term memory(LSTM), TextCNN, convolutional neural network(CNN) and C-LSTM and all these models are combined to get results by soft voting. Among these models, the attention-based LSTM models achieved the best results. The word-embedding used for all models in this task is Word2Vec. Then, the word vectors are fed into the long short-term memory (LSTM) layer. Finally, an attention mechanism (Luong et al., 2015) is added into the neural networks, and the prediction results are output via the softmax activation. The final system achieved an accuracy of 0.6776 and got 14th place in SubTask A. They find that TextCNN achieved a better result than a single convolutional neural network.

3. Dataset Details and Evaluation Measures

In this work, the dataset is taken from SemEval 2019 suggestion mining competition. The dataset for subtask A provides an overall count of 8500 examples, where 6415 examples are labeled as non-suggestion and 2085 as suggestion. Every example contains only one sentence, which could be part of a whole post in the forum where it was extracted. The dataset provided in the Semeval 2019 Suggestion Mining Challenge was highly imbalanced. The Table 1 shows the characteristics of the dataset.

| Dataset       | Total No. of Sentences | Total No. of Suggestions | Total No. of Non-Suggestions |
|---------------|------------------------|--------------------------|-----------------------------|
| Training Data | 8500                   | 6415                     | 2085                        |

The researchers of suggestion mining used precision, recall, F1-score and accuracy as evaluation measures to test the efficiency of the proposed approach. In this work, accuracy is used which is the number of test sentences are correctly predicted their suggestion from a set of test sentences.

4. Deep Learning based Approach for Suggestion Mining

The exponential growth of deep learning techniques helped to solve problems across different fields of research. Deep learning techniques extract features from the data automatically without any manual adjustment. Deep learning is a subdomain of machine learning in which algorithms are inspired by the structure of the human brain. Deep learning models are built upon the way the human brain is structured, having several layers and the information passes through each layer to generate an output. Traditional machine learning models are not scalable i.e. the model performance does not improve as we add more data to the training set. On the other hand, deep learning models are scalable. Deep learning model performance increases as you add more data to the training set. The deep learning model can extract features from the data automatically without extensive handcrafted feature engineering.

Deep Neural Networks (DNNs) have shown great ability at performing tasks on a wide range of applications such as image captioning, object detection, and segmentation. A better understanding of cost functions and the amount of training data has helped the neural network model to learn complex structures in images, videos, audios, and texts. The performance of deep learning algorithms is directly
proportional to the amount of data available for training. Deep learning algorithms simultaneously learn feature extraction and classification parameters through the process of back-propagation. Vectors are used as an input and output to a deep learning model. These vectors are a bunch of numbers which are learned by train the model to minimize the loss. The data coming from various modalities (image, text, audio, video.) is first converted into its vector representation before passing on to the deep learning models. Therefore, it is important that the vector is good enough to represent the data properly.

There are a lot of deep learning algorithms out there for different problems such as CNNs and RNNs. CNN’s are good at extracting features from image data, while RNN’s perform well on temporal data. Similarly, variations of recurrent neural networks such as Long-Short Term Memory and Gated Recurrent Unit architectures do a good job extracting useful information from temporal data such as text and time series data. Although, these networks are good at extracting features for a particular modality, learning features across multiple modalities is still a challenging task. In this work, the experiment conducted with RNN, LSTM, Attention based LSTM and GRU for suggestion mining. All deep learning models used the input vectors generated through word2vec model.

4.1. Recurrent Neural Network (RNN)

RNNs are good at extracting features from sequential data such as time series, text, and audio. RNNs consist of an encoder which encodes the data into a vector and a decoder which decodes the vector into a destination data format. RNNs use some information from the previous layer to predict future outcomes. In contrast to feed-forward networks, RNNs have a feedback loop connecting their past decisions, thus ingesting their earlier outputs as input. Due to the inherent sequential nature, RNNs require an extension of back propagation considering the temporal aspect called Back propagation through time (BPTT). In the back-propagation phase of classic RNNs, the gradient interactions between words that are several steps apart in a sequence can get lost/weaken due to vanishing or exploding gradient problem. RNNs such as Long-Short Term Memory [9] and Gated Recurrent Unit [10] consist of a memory cell which helps remember previous layer information. RNN’s are widely used in applications such as image captioning [11, 12], and video summarization [13].

The basic RNN takes input from the previous time step and current input to predict the next outcome. $h_0, h_1, h_2, h_3, \ldots, h_t$ are the inputs to the time steps $t = 0, 1, 2, 3, \ldots, t$ which are used along with $x_1, x_2, x_3, \ldots, x_t$ to predict the output $y_1, y_2, y_3, \ldots, y_t$. The equation (1) is used to compute the current hidden state output by using previous hidden state output and current input.

$$h_t = f\left(W^{hh}h_{t-1} + W^{hx}x_t\right) \quad (1)$$

The Equation (2) passes the hidden state output to tanh function to restrict the output values in the range from -1 to 1.

$$h_t = \tanh\left(h_t\right) \quad (2)$$

The equation (3) computes the final output by multiplying with the weight matrix.

$$y_t = W^{hy}h_t \quad (3)$$

The weights are updated for each time step by back-propagation using the error calculated for that time step.

RNNs do a great job at extracting features from temporal data but they sometimes face the problem of vanishing gradients. The vanishing gradient problem occurs when the value of gradients
exponentially decreases as it reaches to the early layers. To tackle this problem, LSTM’s and GRU’s are used.

4.2. Long Short-Term Memory
Long Short Term Memory Networks (LSTMs) refer to a type of artificial neural network modelled to utilize patterns occurring in data sequences. The generic term for such networks is Recurrent Neural Networks (RNNs). LSTMs make use of sequential information allowing operations over sequences of vectors. The sequences can exist in the input, output or both. Traditional neural networks consider all inputs independent of each other. On the other hand, RNNs do consider a sequential dependency between them. It can also be considered as having a “memory” recording a summary of the previous events in the sequence. This allows the information to persist throughout the network, and making every sequence output to be conditioned on the previous computations as well.

The LSTM model introduces a memory cell, a new structure to handle the past events in the sequence. The Figure 1 shows the internal structure of a LSTM cell. The Neural Network module of LSTM memory cell contains four main elements such as an input gate, a neuron with a self-recurrent connection, a forget gate and an output gate. The recurrent connection in the LSTM NN module uses a weight value of 1.0 to ensure that the state of a memory cell is remain constant moving from one time step to another in the sequence. The forget gate allows the memory cell to remember or forget its earlier state if required. The input gate controls the altering of the memory cell state following the current state incoming signal. On the contrary, the output gate controls the effect of the memory cell state on other neurons.

LSTM’s are used for problems containing temporal data. LSTM’s solve much of the vanishing gradient problem which is observed in basic RNN’s. LSTM’s come with a memory cell that helps remember the previous information for longer time steps. They were introduced by Hochreiter and Schmidhuber [9]. The Equation (4) computes the new information that is to be stored in input gate i.

\[
i_t = \sigma \left( x_t U^i + h_{t-1} W^i \right)
\]  

(4)

The Equation (5) related to forget gate f which determines the information that is not important for the model to store.

\[
f_t = \sigma \left( x_t U^f + h_{t-1} W^f \right)
\]  

(5)
The Equation (6) related to output gate \( o \) which provide activation to the final output of the LSTM block.

\[
\sigma_t = \sigma(x_t U^o + h_{t-1} W^o)
\]  
(6)

The Equation (7) is related to internal memory unit which is used to store the previous information.

\[
\hat{C}_t = \tanh(x_t U^g + h_{t-1} W^g)
\]  
(7)

The Equation (8) computes current hidden state output \( C_t \) by using the information from previous hidden state and current input.

\[
C_t = \sigma(f_t \ast C_{t-1} + i_t \ast \hat{C}_t)
\]  
(8)

The Equation (9) computes final hidden state output by combining \( C_t \) and output gate. This hidden state is a vector representation of the data which is then further used for a variety of applications such as classification and captioning.

\[
h_t = \tanh(C_t) \ast o_t
\]  
(9)

In these equations \( \sigma \) represents the sigmoid function. It outputs values between 0 and 1. The sigmoid function determines the percentage of information to be passed through the gate.

4.3. Attention based LSTM

This model combines the attention mechanism with the LSTM. The attention mechanism is a good solution to the information vanish problem in long sequence input situations. When dealing with machine comprehension problems, the combination of LSTM and the attention mechanism are more effective than they are used individually. Traditional recursive neural networks are ineffective when dealing with very long sentences. The LSTM model is developed to solve the gradient vanishing or exploding problems in the RNN. The LSTM model can alleviate the problem of gradient vanishing, but this problem persists in long range reading comprehension contexts. The attention mechanism breaks the constraint on fix-length vector as the context vector, and enables the model to focus on those more helpful to outputs. After LSTM layer, we used the attention mechanism on the output vectors produced by previous layer. It is proven effective to improve the performance.

In the attention-based LSTM model, all sentences and labels are converted to word vectors by the word embedding layer. These word vectors will be fed to the LSTM layer. Subsequently, the word vector is represented as a hidden vector. Next, the attention mechanism assigns weights to each hidden vector, and the mechanism produces attention weight vectors and weighted hidden representations.

Note that the weight vector is mainly obtained by calculating the similarity. An attention weight vector is generated by calculating a sentence vector matrix and a label vector matrix. The attention weight vector is then fed to the softmax layer.

The attention mechanism allows the model to retain some important hidden information when the sentence is long. In our mission, the information of sentences is kept for a relatively long time. Using the standard LSTM may result in the loss of hidden information. To solve this possible problem, we have facilitated the attention based LSTM model.
4.4. Gated Recurrent Unit

Fig. 2 displays the internal structure of the GRU network.

![Figure 2. The internal structure of GRU cell [15].](image)

The GRU was first introduced by Chung et al. [10]. GRU is a variant of LSTM. Unlike LSTM, GRU has two gates (reset and update gates). GRU doesn’t have a memory unit. It outputs the entire hidden unit without any control. GRU is less complex and computationally more efficient as compared to LSTM. The following Equations explain the mathematical working of GRU. Equation (10) specifies the Update gate $z_t$ which is computed by using the current input and the previous hidden state output.

$$z_t = \sigma(x^U z + h_{t-1}W^z)$$  (10)

Equation (11) is related to reset gate which determines how much information about past information to forget. Both, update and reset gate use sigmoid functions to constrain the output between 0 and 1.

$$r_t = \sigma(x^U r + h_{t-1}W^r)$$  (11)

In equation (12), $\hat{h}_t$ computes the current memory that is to be passed to final output.

$$\hat{h}_t = \tanh(x^U h + (r_t h_{t-1})W^h)$$  (12)

The equation (13) computes the final memory $h_t$ at the time step $t$.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t$$  (13)

5. Experimental Results

The hyper-parameters of deep learning models play a vital role in the accuracies of suggestion mining. In the experiments, it was identified that the same model will get different results under different parameter adjustments. Therefore, reasonable adjustment of parameters during the experiment is also a factor in obtaining a good experimental result. In this work, Tensorflow is used to run these models. Tensorflow is a machine learning framework with great documentation for training and configuring neural networks which is developed by Google.
All deep learning models used the following hyper-parameters in the experiment such as number of epoch = 10, Embedding vectors size = 200, Dropout = 0.5, Recurrent Dropout = 0.4, Maximum sequence length = 1200, layers = 3, Attributes = 200 and activation function = “tanh”. The table 2 shows the accuracies of suggestion mining when experimented with different deep learning techniques.

Table 2. Accuracies of Suggestion Mining using different deep learning methods

| Deep Learning Model | Accuracy |
|---------------------|----------|
| RNN                 | 81.34    |
| LSTM                | 82.49    |
| GRU                 | 83.82    |
| Attention based LSTM| 85.97    |

Regarding deep learning methods, the Attention based LSTM technique achieved good accuracy of 85.97 for suggestion mining when compared with other deep learning techniques. The LSTM model accuracy is improved when attention layer is added in the network of LSTM.

6. Conclusions and Future Scope
In this work, a solution is proposed for suggestion mining which is used to classify the sentences as suggestions or non-suggestions. The experiment conducted with different types of deep learning models such as RNN, LSTM, Attention based LSTM and GRU. Among these deep learning models, the Attention based LSTM technique achieved good accuracy of 85.97% for suggestion mining.

In the future, we are planned to implement a model by combining multiple deep learning models. We also planned to experiment with machine learning algorithms by identifying a set of stylistic features.

References
[1] Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pages 2159–2167, Lisbon, Portugal. ACL.
[2] Sapna Negi, Kartik Asooja, Shubham Mehrotra, and Paul Buitelaar. 2016. A study of suggestions in opinionated texts and their automatic detection. In Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, pages 170–178, Berlin, Germany. ACL.
[3] Masahiro Yamamoto, Toshiyuki Sekiya, “m_y at SemEval-2019 Task 9: Exploring BERT for Suggestion Mining”, Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 888–892, Minneapolis, Minnesota, USA, June 6–7, 2019.
[4] Tirana Noor Fatyanosa, Al Hafiz Akbar Maulana Siagian, Masayoshi Aritsugi, “DBMS-KU at SemEval-2019 Task9: Exploring Machine Learning Approaches in Classifying Text as Suggestion or Non-Suggestion”, Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1185–1191, Minneapolis, Minnesota, USA, June 6–7, 2019.
[5] Tobias Cabanski, “DS at SemEval-2019 Task 9: From Suggestion Mining with neural networks to adversarial cross-domain classification”, Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1192–1198, Minneapolis, Minnesota, USA, June 6–7, 2019.
[6] Aysu Ezen-Can, Ethem F. Can, “Hybrid RNN at SemEval-2019 Task 9: Blending Information Sources for Domain-Independent Suggestion Mining”, Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1199–1203, Minneapolis, Minnesota, USA, June 6–7, 2019.

[7] Ilia Markov, Eric Villemonte De la Clergerie, “INRIA at SemEval-2019 Task 9: Suggestion Mining Using SVM with Handcrafted Features”, Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1204–1207, Minneapolis, Minnesota, USA, June 6–7, 2019.

[8] Junyi Li, Haiyan Ding, “Lijunyi at SemEval-2019 Task 9: An attention-based LSTM model and ensemble of different models for suggestion mining from online reviews and forums”, Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1208–1212 Minneapolis, Minnesota, USA, June 6–7, 2019.

[9] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[10] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

[11] Jeffrey Donahue et al., “Long-term recurrent convolutional networks for visual recognition and description,” in CVPR, 2015, pp. 2625–2634.

[12] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in International conference on machine learning, 2015, pp. 2048–2057.

[13] S. Sah, S. Kulhare, A. Gray, S. Venugopalan, E. Prud’Hommeaux, and R. Ptucha, “Semantic Text Summarization of Long Videos,” in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 989–997.

[14] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[15] https://www.data-blogger.com/2017/08/27/gru-implementation-tensorflow/