International Journal of Engineering & Technology, 7 (4.5) (2018) 402-404

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Complaint Classification using Word2Vec Model

Mohit Rathore[1*], Dikshant Gupta[2], Dinabandhu Bhandari[3]

[1]*Indian Institute of Technology, Dhanbad, India*
[2]*Indian Institute of Technology, Dhanbad, India*
[3]*Heritage Institute of Technology, Kolkata, India*
**Corresponding author E-mail:mohit@ece.ism.ac.in*

Abstract

Attempt has been made to develop a versatile, universal complaint grievance segregator by classifying orally acknowledged grievances into one of the predefined categories. The oral complaints are first converted to text and then each word is represented by a vector using word2vec. Each grievance is represented by a single vector using Gated Recurrent Unit (GRU) that implements the hidden state of Recurrent Neural Network (RNN) model. The popular Multi-Layer Perceptron (MLP) has been used as the classifier to identify the categories.

Keywords: *Gated Recurrent Unit; Recurrent Neural Network; Text Classification; Word2Vec*

1. Introduction

Gathering grievances and their analysis is an important, sometimes mandatory task in improving the services of an organization. The most popular technique in collecting the grievances is through phone calls. The complaint made by the caller is recorded and converted to text. The texts are then analyzed and assigned to different departments for possible resolution. It has become a need to automatically categorize the grievances in a set of predefined classes to make the task easier for the organization in resolving the issues at earliest.

This problem can be considered as a more profound version of sentiment analysis. Sentiment analysis focuses on whether the attitude towards a particular text is neutral, negative or positive. The complaint classification problem can better be categorized as a text classification problem in which unlike only three categories (neutral, positive and negative), one may have number of possibilities equal to the number of departments in which the texts are to be segregated.

The major challenge in the sentiment analysis is the representation of text so that machine can process the information for possible classification. Several text representation schemes and their classification have been found in literature [1,2,3,4]. Recently, researchers have adopted a novel approach called WORD2VEC in representing a word as a vector. The first mention of word representations dates back to 1986 by Rumelhart et. al. [5] in their paper Learning representations by back propagating errors [6]. Inspired by which the word2vec was developed by Tomas Mikolov et al. at Google. The concept of word2vec is further based on two models as described in Efficient estimation of word representations in vector space [6].

The continuous-bag-of-word model (CBOW) tries to predict the word w_i given the context. The input to the model could be w_{i-2}; w_{i-1}; w_{i+1}; w_{i+2}, the preceding and following words of the current word. The output of the system will be w_i. On the other hand, skip-gram model considers the input word wi to the model, and the predicts the output w_{i-2}; w_{i-1}; w_{i+1}; w_{i+2}. Here the model aims to predict the context given a word. Also, the context is not limited to its immediate neighbouring words, training instances can be created by skipping a constant number of words in its context, so for example, w_{i-4}; w_{i-3}; w_{i+3}; w_{i+4} [7,8].

In this article, attempt has been made to develop a classifier for the orally acknowledged grievances into one of the predefined categories. The complaints are first converted to text and then each word is represented by a vector using word2vec. Each grievance is then represented by a single vector using Gated Recurrent Unit (GRU). Finally, the popular Multi Layer Perceptron (MLP) has
been used as the classifier to identify the categories.

In the next section, we will describe how a complaint can be characterized by vector representations that would be used in classifying a complaint in different classes. Section 3 presents the results of the proposed methodology and the final section contains concluding remarks and a short discussion on future scope of work.

2. Classification of Complaint

A text based complaint can be defined as a collection of sentences where each sentence is an ordered set of words. In certain cases a complaint can be as short as one sentence with few words while in some cases it can be as long as tens of sentences. In this article we have considered complaints of maximum word count up to 750 words. For efficient processing of each complaint one has to identify each complaints department and direct it to its respective department. It is clear that the problem of identifying the department is nothing but to classify the complaints in different classes such as Credit Card, Mortgage, Student loan and so on. Mathematically, let p_1; p_2; · · · ; p_N be the complaints and c_1; c_2; · · · ; c_M; (M ≤ N) be the classes of the complaints. In other words, all p_i; i = 1; 2; · · · ; N can be classified as one of the C_j; j = 1; 2; · · · ; M.

The primary challenge in developing a methodology that would automatically classify complaints, is to present a complaint to the machine. Specifically, complaints have to be fed to a machine so that it can learn patterns present in the complaints and identify each unknown complaint into a particular class. In this work, each complaint is represented as vector which is further used to identify the class.

Once, vectors for each complaint are obtained they are classified using a multi-layer perceptron (MLP) based classifier. The steps involved in the classification are as follows:

1. Voice to text Conversion: Convert each oral complaint to text.

2. Data Pre-processing: Converting each complaint into a list of words that can be fed to the model to generate vector.

3. Embedding Layer: Computing vector representation for each word.

4. GRU Layer: Representing sequential collection of word vectors as a single vector.

5. MLP classifier: A multi-layer perceptron or a dense layer to classify the vector notations of each complaint into different classes.

2.1. Data pre-processing

One of the important steps in a pattern classification technique is pre-processing of the raw data. There are two major tasks for the preparation of data. First, to convert complaints from a standard string, obtained after converting the oral complaints to texts, to a list of words. This task is completed using regular expressions based tokenizers. Regular expressions are a sequence of symbols and characters expressing a string or pattern to be searched for within a longer piece of text. Python's natural language toolkit provides excellent functions for such operations and have been used during experimentation for this paper. Stop words like 'is', 'the', 'and', 'of' are not discarded to preserve the contextual information. Second, for the embedding layer to process the text a numerical format of input text is required. This task is completed by indexing each different word and creating a word index dictionary to map each word to an index.

2.2. Embedding layer

This layer is effectively the word2vec layer of the model. We chose to implement our own layer rather than using gensim or glove vectors as it gave our model flexibility to learn word representations specific to the classification problem. This layer is a matrix of size V X 100, where V is the number of words in the vocabulary and 100 is dimension of each word vector. Hence, each row of the matrix is vector representation of a specific word. For each complaint, index of every word in the complaint is fed sequentially to the layer and corresponding row from the matrix is selected. These representations are further fed to the GRU layer sequentially.

2.3. GRU layer

A GRU (Gated Recurrent Unit) is a different way to calculate the hidden state of Recurrent Neural Net (RNN) model. Vanishing gradient problem prevents standard RNNs from learning long-term dependencies. GRUs are designed to combat vanishing gradients through a gating mechanism. A GRU has two gates a reset gate , and an update gate. Intuitively, the reset gate determines how to combine the new input with the previous memory, and the update gate defines how much of the previous memory to keep around. This mechanism helps to learn long term dependencies [9]. This procedure of taking a linear sum between the existing state and the newly computed state is similar to the Long short term memory (LSTM) unit. The GRU, however, does not have any mechanism to control the degree to which its state is exposed, but exposes the whole state (h_t) each time. The candidate activation is computed similarly to that of the traditional recurrent unit and as in [10],

$$h_t = \tanh(Wx_t + U(r_t \circledast h_{t-1})) \quad (1)$$

where, x_t is the input word vector, W and U are the weight matrices. r_t is a set of reset gates. When off (r^j_t close to 0), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state. The reset gate r^j_t is computed similarly to the update gate:

$$r = \sigma(W_r x_t + U_r h_{t-1})^j \quad (2)$$

where, $\sigma(\cdot)$ is as usual a logistic sigmoid function. Each word representation from embedding layer is fed to GRU. Output corresponding to the last word of each complaint represents entire complaint as a vector.

2.4. MLP classifier

Output from GRU layer is passed through an MLP consisting of D (= dimension of the word vector) input neurons and M output neurons, one for each class. Output layer generates the probability for each possible class and predicts the class with the highest probability. One hidden layer with $(D+M)=2$ (the average of input and output layers) neurons have been considered in our experiment.

2.5. Training and testing

Backpropagation algorithm has been used to train the model. While the MLP classifier has been trained using standard backpropagation, GRU layer has been trained using a modification of standard backpropagation namely BPTT (backpropagation through time). The only difference between both the algorithms is that standard backpropagation is used where parameters to be trained are different for each layer while BPTT is used where parameters are shared between different layers. Adam optimizer was used to optimize the training process [11]. It has been seen that Adam generally outperforms other optimization techniques such as RMSProp, Momentum, SGD.

For training and testing we have chosen a split of 3:2 i.e. 60% of the data has been used for training while 40% of the data has been used for testing. All the training and testing has been done using python programming language. We have built and trained our model using Keras library

3. Results and Discussions

To demonstrate the efficacy of the proposed approach, 105504 complaints are considered belonging to 12 classes. Each complaint is transformed into a 100 dimensional vector. To validate the effectiveness of the methodology, the experiment is performed multiple times by randomly selecting different training data sets. It has been observed that each run produces consistently similar results. We are able to achieve 85:18% classification accuracy by training the model for two epochs.

Figure 2 represents the change in validation loss (green) and training loss (blue) with respect to increasing epochs. As expected, the training loss decreases on increasing the epochs. This may be due to the fact that over time (increase in epochs), the model starts to overfit and hence a performance decline is observed on test data. Validation loss provides a general idea of wrong classifications occurred in each epoch. As seen from the graph, its

value decreases initially and then increases. These observations give us a clear idea of where the overfitting might begin. Figure 3 represents the change in validation accuracy (green) and training accuracy (blue) with respect to increasing epochs. Both the accuracy curves counter the loss curves in figure 2, as expected. The rise and fall of validation accuracy also gives us the idea of the optimum number of epochs for training. Since after four epochs validation accuracy starts to decline and validation loss increases gradually, it is evident the model starts to overfit the learning. Therefore, four epochs are chosen as an optimum for training the system with the data considered in our experiment.

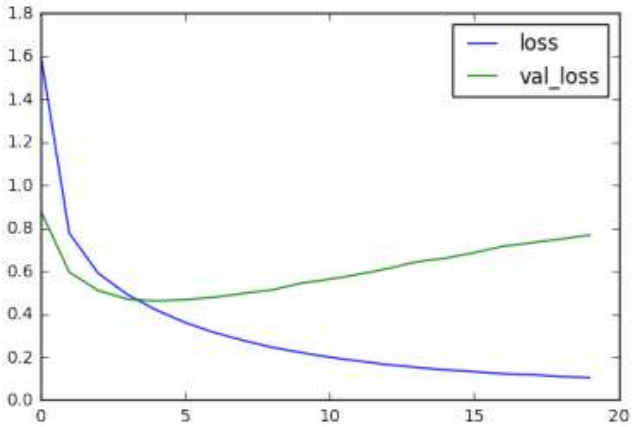

Fig. 1: Loss vs Epochs

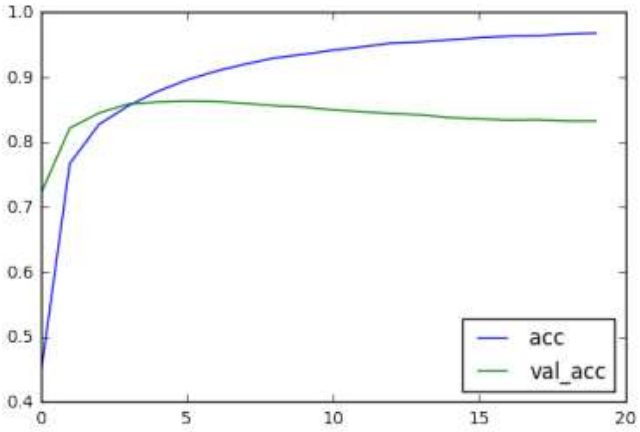

Fig. 2: Accuracy vs Epochs

4. Conclusion

An automatic complainant grievance classifier has been proposed here. Each word in the text of the grievances is first represented as a vector using the Word2Vec concept. The grievances are then represented as an aggregated vector using Gated Recurrent Unit (GRU). Multi Layer Perceptron has been used as the classifier to identify the categories. The proposed approach achieved more than 85% accuracy.

Only unidirectional RNN were employed while building the model. As a future experiment one may use bidirectional RNN (Bi-RNN) that is found to perform better in vector representation tasks [12]. Other variations like Attention Mechanism and Stacked RNNs can also be explored as a part of future experimentation.

References

[1] R. Collobert and J. Wetson, Fast semantic extraction using a novel neural network architecture. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics,, pp. 560-567, 2007.

[2] F. Sebastiani, Machine learning in automated text categorization. ACM Computing Surveys (CSUR), vol. 34, pp. 1-47, 2002.

[3] Yoshua Bengio, Rjean Ducharme, Pascal Vincent, Christian Jauvin, A Neural Probabilistic Language Model. Journal of Machine Learning Research 3 (2003) 11371155

[4] D. Bhandari and P. S. Ghosh, Parametric representation of paragraphs and their classification. In Proc. 2nd Int. Conf. on International Conference on Advanced Computing, Networking and Informatics (ICACNI-2014), Springer, 179-186, Kolkata, 2014.

[5] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams Learning representations by backpropagating errors. Nature, 323(6088):533536, 1986.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781,2013

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean**,** Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 31113119, 2013.

[8] Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig Linguistic Regularities in Continuous Space Word Representations Proceedings of NAACL-HLT 2013, pages 746751,Atlanta, Georgia, 914 June 2013

[9] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs/1412.3555, 2014.

[10] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate. Technical report, arXiv preprint arXiv:1409.0473, 2014.

[11] Diederik P. Kingma and Jimmy Ba, Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG], December 2014.

[12] Mike Schuster and Kuldip K. Paliwal, Bidirectional Recurrent Neural Networks., IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 11, NOVEMBER 1997