**Analysis of Approaches to Question Classification**

Ananya Sharma, Brighton Lema, Guru Guha Narayanan, Oishik Ghosh, Priyanka Raj, Shivaji R Chaudhuri

**Abstract-** Deep learning algorithms have repeatedly been proven successful in solving several text-mining problems, including text classification, document categorization, web mining, sentiment analysis, and spam filtering. In this paper, we illustrated and examined methods of building a question classifier using different word embeddings and methods of sentence representation (Bag of Words and BiLSTM). In addition to reviewing different methods, a comparative study that examines the impact of learning rate and batch size on the models is also presented.

**Index Terms**- Bag of Words, BiLSTM, Neural Networks

**I. INTRODUCTION**

Text classification or categorization is a classic NLP task that assigns labels or tags to textual units such as phrases, queries, paragraphs, and documents. Text classification mainly includes topic classification, question classification, sentiment analysis, spam identification, news categorization, user intent classification, and content moderation. As a result, finding a general strategy that consistently performs for all types of text categorization issues is difficult.

Furthermore, most extant research on traditional text classification focuses on a single phrase or sentence. These methods handle the text categorization problem solely based on the target sentences or target words without considering the relationship between each word.

The text's classification should be based on all the contexts. Traditional sentiment analysis focuses on determining a text's polarity (e.g., positive, negative, or neutral) based on language clues retrieved from the textual contents of sentences. Deep learning on text classification can be classified either using neural language models to learn word vector representation or by performing classification composition over the learned word vectors.

CNN and RNN are two deep learning models that are generally utilized when dealing with question classification. CNNs can learn the local response from temporal or spatial data, but not sequential correlations. However, RNNs, unlike CNNs, are designed for sequential modelling but cannot extract features in parallel. RNNs are more commonly utilized when dealing with sequential modelling tasks. Traditional RNNs cause exploding and vanishing states against their gradient for extended data sequences.

In terms of its tremendous ability to extract high-level text information, LSTM is critical in NLP. Bidirectional long short-term memory (BiLSTM) is a version of LSTM that combines the forward hidden layer and the backward hidden layer, allowing access to both previous and subsequent contexts.

**II. PLAN OF PURPOSE**

We initially start with processed and tokenized training data to build a dictionary. Random Embeddings and Pre-trained word embedding (Glove/Word2Vec) techniques were utilized for word embedding. Both Bag of Words and BiLSTM are implemented to generate sentence representations from the word embeddings. A feed-forward neural network with a SoftMax output layer is used to train the classifier. Validation is done simultaneously to observe model performance. Different combinations of embedding and sentence representation strategies have been implemented to see which one performed the best. Freezing and fine-tuning of the word embeddings have also been incorporated to test the performance.

**III. LITERATURE REVIEW**

Fabio Gasperetti [1] discusses the “word embeddings” approach, which follows a statistical language model to represent text-based learning objects as latent spaces of marginal dimensions. The Random Forest classifier is proven to be extremely useful for its capacity to learn muddled multi-dimensional mappings and distinguish the embedding patterns and relationships. This study also compares the same model for five datasets. It recognizes that the scarcity of data prerequisites in one of the datasets leads to the inadequately trained binary classification model. This highlights the necessity of pre-trained word embeddings.

Mohamed Ibrahim et al. [2] review traditional methods for vector representation. These were based on word-document co-occurrence matrices of higher dimensional and prone to sparse. Early methods tried to solve this problem by defining low dimensional vectors to a group of context words; however, current approaches employ neural network models to learn vector representation of terms due to increased computational power.

A "Bag of Words" transformation of a textual document requires decomposition into the words that comprise it, with a frequency assigned to each of these words. There could be several ways to improve the scope of this model. For example, the proposed model by Nikolaos Passalis et al. [3] directs attention towards an exciting method of pruning the word references. It broadens the BoF model by assigning a weighting mask that considers changing the significance of each coded word, advancing the model end-to-end (from the used word embeddings to the weighting filter).

A couple of unrecognized limitations of the BoW model are seen, depending on the semantic relation between words. For example, the phrases "a dozen of a couple" and "a couple of a dozen" have similar word frequencies, so notwithstanding their varying implications, they could be addressed indistinguishably in a pack of words. Zekeriya Anil Guven et al. [4] represent the semantic relationships between different terms, keeping away from the deficiency and loss in conventional models. A significant increase in accuracy (6.6 to 8.76%) is observed.

Although the impressive performance by the BERT Model when it comes to aspect detection is comparatively backed in the study conducted by Chi Sun et al. [5], it also brings forward the inability of the model in terms of sentiment classification as compared to the Dmu-Entnet model, thereby showing it has both pros and cons.

Secondly, words such as "create" and "develop" lead to the trivial model struggling. Despite having similar implications in a scenario of an invention, they are treated as unique vectorial elements. Mathias Kraus et al. [6] discusses implementing a pre-trained deep learning model to ensure similar vectorial representations assigned to such words. The recurrent neural networks allow cyclic relationships between neurons on which the network can memorize information that persists on word transitions.

Backheol Jang [7] mentions that LSTM models capture long-term connections between word sequences, resulting in improved text classification; the number of characteristics to memorize for categories remains large, causing the training process to be hindered. A hybrid Bi-LSTM-CNN model takes advantage of the benefits of bit models to assess accurate categorization on data. He demonstrates how to use the BiLSTM attention layer to access both prior and subsequent contextual information to create a representation of the sequence.

Gang Liu et al. [8] suggest an innovative and unified bidirectional method of LSTM known as BiLSTM. It has an attention mechanism and convolutional layer to overcome the above challenges. BiLSTM hence, comprises two independent LSTMs as a measure of syntactical analysis that acquires word annotations by summing input from two directions of a sentence and merging sentimental information in the annotation, allowing it to handle a selection of long sentences.

Yunxiang Zhang et al. [9]mentions how she overcomes such challenges with other deep learning techniques as the subject of NLP grows in popularity by investigating n-BILTSM, a method based on n-gram characteristics, in conjunction with BILSTM. They deployed the n-gram theory, which defines a series of n items from a given sequence of documents or speech data. Then, the n-gram parser initially classifies each item as a 'unigram,' 'bigram,' or 'trigram'.

Shunichi Ishihara [10] outlines how to carry out a score-based probability proportion way to predict the strength of a similar model incorporating BoW. A strategy for addressing text information deployed each record and the estimated score of the archives under examination by testing three special measures (Euclidean, Manhattan, and Cosine). The likelihood appropriations of the score assembled utilizing the common source strategy were parametrically displayed by the Normal, Log, Gamma, and Weibull circulations. The author chooses the best-fitting model independently for the equivalent genre and different-genre score dispersions. This method led to increased performance of validation accuracies. However, it is only shown to have significant improvements in cases of multiple text documents written by several authors.

**IV. METHODOLOGY**

The sentences will have to be preprocessed before they can be used for training the models. The sentences are split and the labels are separated from the questions. The questions are then converted to lowercase to maintain consistency in the flow, and the punctuations are removed. A vocabulary containing all the words contained in the questions is created. We remove the stop words and the words below a minimum frequency of three.

We then perform word embedding where each word is represented as vectors. Here, we use two kinds of embeddings: Random Embeddings and Pre-Trained Embeddings (GloVe). In the case of random embeddings, every word will get assigned a randomly selected vector (we have used a vector of size 200), while in the case of GloVe, every word has a 300-dimensional vector pre-trained on an alternative file. Word vector representation enhances the capability to retain useful information from unstructured text. Machine learning algorithms can easily understand a word represented in vectors and obtain meaningful information from them. The idea in word representation is that words with similar meanings will also have similar vector representations.

After the word embeddings are developed, each sentence undergoes tokenization.

The models to be used for this stage are BoW and BILSTM. An additional Ensemble model is also implemented, which combines the results of all the classes (each of the above two model predicts its own class) and results the maximum occurrences.

The training file as a whole contains 5452 questions belonging to 50 different classes. The data is split into 90% training set and 10% validation set. The test file contains 500 questions whose classes are known.

The output of the BoW/BILSTM is fed into the neural network. Six models with different embedding and representation strategies are built and are then tested using a set of questions whose classes are known.

An ensemble model that combines the results of the trained models is also implemented.

**V. EXPERIMENTING WITH HYPERPARAMETERS**

To find the best hyperparameters (batch size and learning rate), we trained the models with different values of them and compared the performance of the models using the validation accuracy.

Figure 1 displays the validation accuracies achieved for the different models according to the batch sizes. The learning rate was kept constant with a value of 0.1.

Table

Description automatically generated with low confidence

Figure 1 Accuracies for different models with different batch sizes

We can clearly see from Figure 2 that an increase in the batch size leads to a decrease in the validation accuracy. Using a batch size of 4 gives the best results, therefore it was chosen as the optimum value. Further experimenting with the hyperparameter of learning rate has been done keeping a constant batch size of 4.

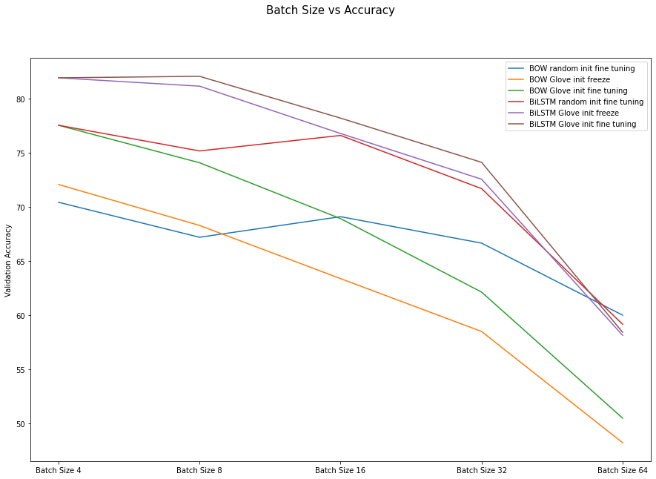


Figure 2 Batch Size vs Validation Accuracy

Figure 3 displays the recorded validation accuracies when the models were trained with different learning rates.

Graphical user interface

Description automatically generated

Figure 3 Learning rates for different models

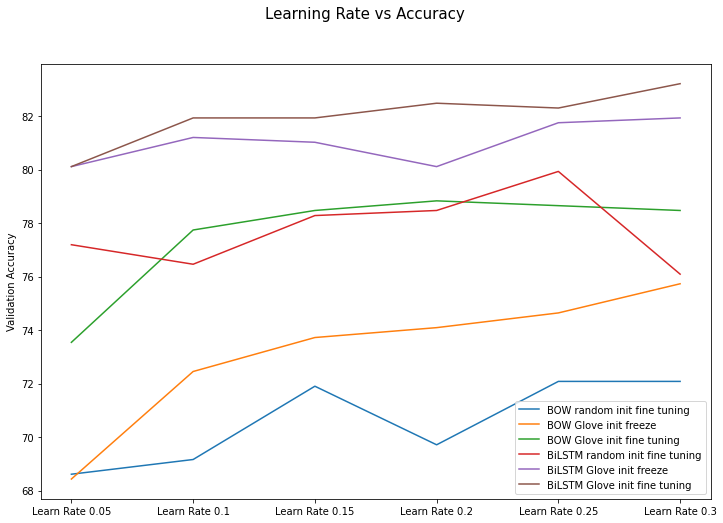


Figure 4 Variable learning rate with constant batch size

It is clear from Figure 4 that there is a general trend where accuracy increases minimally with learning rate. However, in most models after a certain learning rate there is either a drop in the accuracy or it remains constant. A balance of an appropriate learning rate has to be achieved, too low learning rate does not yield the required accuracy and too high learning rates can cause model instability. It can be observed from the graph that with a learning rate of 0.25 most models have a higher accuracy; hence this is chosen as the learning rate for the final models.

**VI. RESULTS AND ANALYSIS**

The final models are trained for 20 epochs using a learning rate of 0.25 and a batch size of 4. Figure 5 shows the changes in the train and validation accuracies over the epochs.

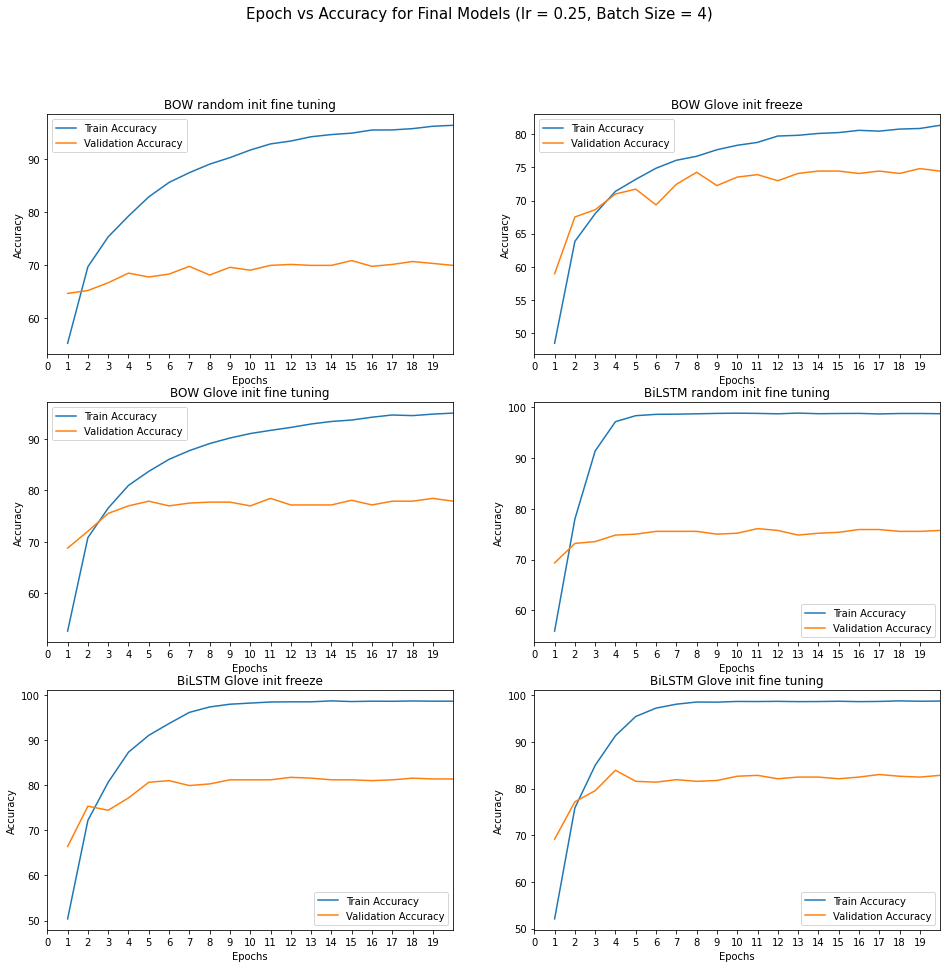


Figure 5 Accuracy at different epochs for the final models

Figure 6 displays the values of the evaluation metrics for the different models.

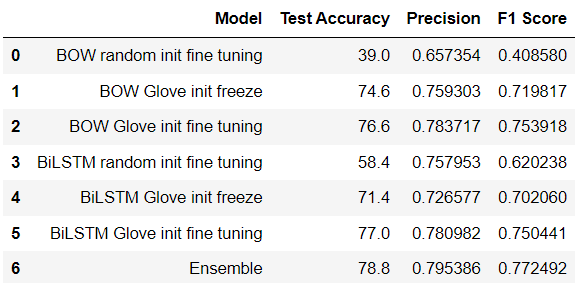


Figure 6 Performance evaluation for the final models

Figure 7 shows the confusion matrices for predicted classes of the individual models. The confusion matrix shows which classes get the most correct predictions. Since 8 of the 50 classes were not present in the test set we could not properly capture the generalizing capacity of the classifiers.

From Fig 2 and 4 it can be concluded that Models with fine-tuning predict better than models that freeze the layers in word embedding. The validation accuracy is observed to be less for randomly initialized word embeddings.

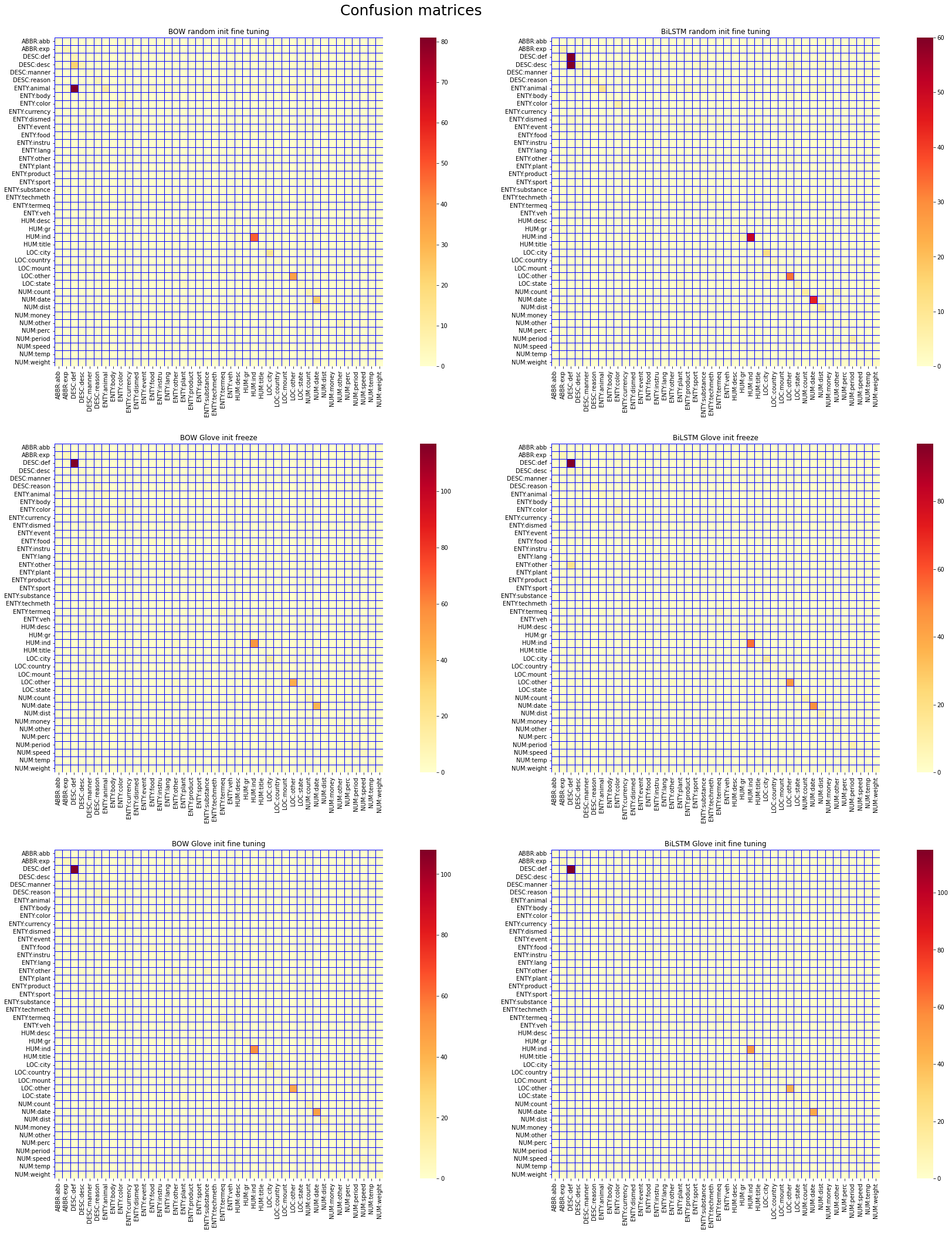


Figure 7 Confusion Matrices for the models

**VII. CONCLUSION**

We have explored deep learning algorithms that are used for text classification, in specific question classification was done. Different strategies of word embeddings and sentence representations were used to build the models. All the built models had an accuracy greater than 50% with an exception of the bag of words representation with randomly initialized word embeddings with fine tuning. It can be concluded that the model using BiLSTM sentence representation and pretrained embeddings outperformed the others.

To improve the model performance the size of train and test data set can be varied while monitoring the model accuracy. In addition, words which have been used in different variations in the text depending on their grammar (either noun or verbs) can be normalized in to their root forms (Lemmatization), this will improve model performance as one feature can represent different variations rather than having separate features for each variation. In other cases, combination of two words into a single feature yields better model performance when compared to each word having a separate feature.

**REFERENCES**

[1] Gasparetti, F. (2022). Discovering prerequisite relations from educational documents through word embeddings. Future Generation Computer Systems, 127, 31–41. https://doi.org/10.1016/j.future.2021.08.021

[2] Ibrahim, M., Gauch, S., Gerth, T., & Cox, B. (2021). WOVe: Incorporating Word Order in GloVe Word Embeddings. ArXiv, abs/2105.08597.

[3] Passalis, N., & Tefas, A. (2018). Learning bag-of-embedded-words representations for textual information retrieval. Pattern Recognition, 81, 254–267. https://doi.org/10.1016/j.patcog.2018.04.008

[4] Guven, Z. A., & Unalir, M. O. (2022). Natural language based analysis of SQuAD: An analytical approach for BERT. Expert Systems with Applications, 195, 116592. https://doi.org/10.1016/j.eswa.2022.116592

[5] Sun, C. (2019, March 22). Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence arXiv.Org. https://arxiv.org/abs/1903.09588

[6] Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. Decision Support Systems, 104, 38–48. <https://doi.org/10.1016/j.dss.2017.10.001>

[7] Jang, B., Kim, M., Harerimana, G., Kang, S. U., & Kim, J. W. (2020). Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism. Applied Sciences, 10(17), 5841. https://doi.org/10.3390/app10175841

[8] 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

[9] Yunxiang Zhang & Zhuyi Rao (2020). n-BiLSTM: BiLSTM with n-gram Features for Text Classification. IEEE Conference Publication IEEE Xplore. https://ieeexplore.ieee.org/document/9141692

[10] Ishihara, S. (2021). Score-based likelihood ratios for linguistic text evidence with a bag-of-words model. Forensic Science International, 327, 110980. https://doi.org/10.1016/j.forsciint.2021.110980

[11] PyTorch. (2022). PyTorch. https://pytorch.org/

[12] 3.3. Metrics and scoring: quantifying the quality of predictions. (2022). Scikit-Learn. https://scikit-learn.org/stable/modules/model\_evaluation.html