# Coursework 1: Question Classification

**Yiran Gao** 10462072
**Pietro Mazzaglia** 10553428
**Chris Tansey** 10587967
**Yulong Wu** 10448782

## Abstract

The challenge of question classification is long-standing, providing potential to improve question answering systems. We explore bag-of-words and Bidirectional LSTM as semantic and deep learning-based models respectively as possible solutions to this problem, and develop a framework to test and evaluate them. In the same vein, we explore ensembles of these models, as well as mixing them into a combined preprocessing stage for a common classification network. Our results showed the best performance for the non-ensemble BiLSTM model, accurately classifying most of the 50 classes of questions of the tested dataset.

## 1 Introduction

Question Answering (QA) aims to automatically provide accurate and concise answers to questions expressed in natural language (Bouziane et al., 2015). To achieve this, a QA system might need to perform Question Classification (QC), which has been shown to cause a boost in performance, helped by the contextual information connected to a question's type (Hovy et al., 2001).

Several semantic-based techniques exist for the QC task, but they generally demonstrate scarce generalisation, as discussed in (Metzler and Croft, 2005). Machine learning approaches have recently led to remarkable progress in this area (Li and Roth, 2002; Zhang and Lee, 2003; Gharehchopogh and Lotfi, 2013), with Deep Learning (DL) methods generally achieving better performance (Zhou et al., 2015)(Zhou et al., 2016/12). For example, in (Tan et al., 2015; Surkova et al., 2019), they combine word embeddings and LSTM for text classification.

Similarly, in this work we approach the QC task by combining semantic-based methods, such as bag-of-words (BoW), with DL-based ones, such as Bidirectional Long-Short-Term-Memory (BiLSTM). The paper is organised as follows: in Section 2, we discuss the general classification architectures used and the implementation of the models. In Section 3, we describe our experiments and present the results obtained. In Section 4, we evaluate these results with statistical methods and deeper performance insight with a confusion matrix.

## 2 Deep Learning Classifiers

In this work we analyse several DL-based models for QC: BoW, BiLSTM, a mixed model (BoW+BiLSTM), and their respective ensemble models. All the models can be plugged into the classification framework presented in Figure 1.

After being preprocessed, the data flows from the input layer to a word embedding. Then, the desired model extracts the information and transfers it to a feedforward NN (FFNN). Finally, the network is plugged into a softmax classifier, which delivers the predicted label in the output layer.

### 2.1 BoW

In a BoW model, each sentence is represented by the set of words appearing in it, disregarding the frequency of occurrence of each word and any structural information present in the sentence.

The vector representation of this BoW is determined by calculating the mean of the vector representations of its component words.

### 2.2 BiLSTM

A BiLSTM is an extension of the classical LSTM model (Hochreiter and Schmidhuber, 1997). It processes sequential data in both directions (forward and backward) using two hidden layers, and then combines the results obtained for each input into a unique structure. In the text mining context, the advantage of such model consists in being more tolerant to word inversions.

### 2.3 BoW+BiLSTM

In the mixed model we devised, the outputs of a BoW model and of a BiLSTM are combined.

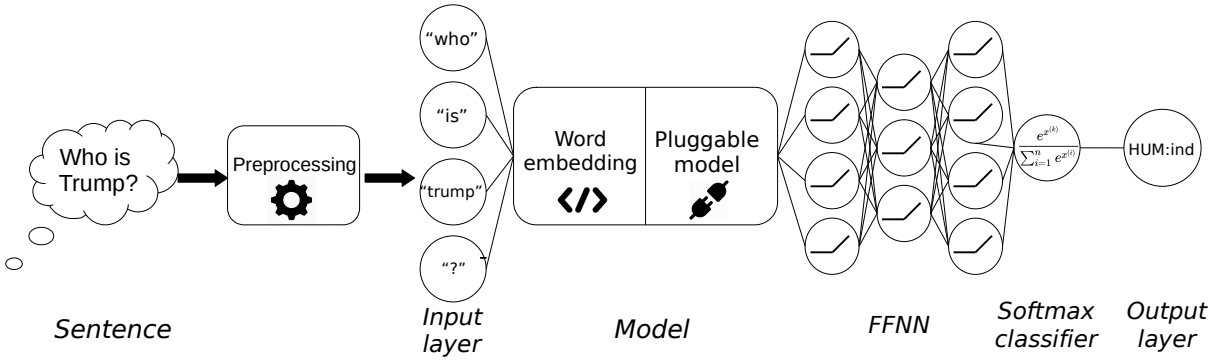
While the motivation behind this model was that more information should improve performance, no improvement was found, as discussed in Section 3.

### 2.4 Ensemble Models

The ensemble modelling procedure is a process where multiple models can be created in order to predict an overall outcome by using different model algorithms or distinctive training datasets.

In this project, we use ensembles using the bagging technique and applying the bootstrap algorithm to provide each model with different data.

Figure 1: Architecture to integrate different models into a text mining flow for question classification.




## 3 Experiments

In this section, we illustrate our experiments.

### 3.1 Settings

The experiments were executed on a machine running Ubuntu 18.04 LTS, with an Intel i7 9750H (2.60GHz) CPU, 16GB DDR4 2666Mhz of RAM and a GeForce RTX 2070 MAX-Q 8GB GPU.

All the models may be run with several different parameters defined by the user in a configuration file, such as learning rate, number of layers in the NN, hidden layers dimensions, and many more.

The hyperparameters were selected performing an extensive selection, using a validation set. However, according to (Greff et al., 2015), the hyperparameters were tuned independently from each other, apart from the batch size, the NN size and the learning rate, which showed high correlation.

### 3.2 Dataset and Preprocessing

The dataset we used is that of (Li and Roth, 2002). The training dataset provided was split 90-10 into a training and validation set, to tune the hyperparameters and facilitate early stopping.

We preprocessed the sentences by applying a simple tokenisation. Lowering tokens (default: Yes) and removing numerical tokens (default: No) are optional preprocessing steps. For each sentence, the result of preprocessing is provided to the input layer of the network as a vector of tokens.

### 3.3 Results

All the models were tested on three different configurations: using a random embedding (128 dimension), using frozen pretrained embeddings (a reduced set of Glove embeddings), and then fine-tuning the Glove embeddings in the training stage. The resulting accuracies are presented in Table 1.

For the BoW, we used a 2-layered NN with 512 ReLU neurons per layer and learning rate 0.01. For the BiLSM and the BoW+BiLSTM models (256 hidden layer dimension), we found that just using a fully connected layer before the softmax classifier provided best results, using a learning rate of 0.1. Ensemble models use 7 models with the same parameters and bootstrapped datasets 95% of the original training dataset size.

All the models were trained with Cross Entropy loss and the SGD optimization algorithm, with a batch size of 1. A sample training process is depicted in Figure 2 for the non-ensemble models.

Comparing the results, we observe that the models using the pretrained embedding consistently outperform the ones using the random embedding, with additional benefits found when fine-tuning.

BiLSTM and BoW+BiLSTM equally have the best performance, with slight differences from their respective ensemble models. BoW was the weakest performing model, but gained a slight improvement as an ensemble.

Finally, from the training plots, we observe that

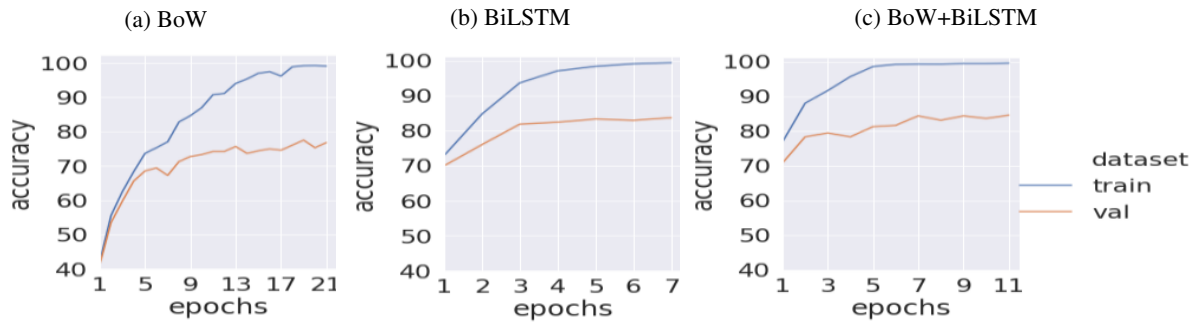
| | Random | Pretrained | |
|---|---|---|---|
| Models | Tuning | Freeze | Tuning |
| BoW | 67.8% | 78.2% | 78.6% |
| BiLSTM | 71.8% | **85.6%** | **87.2%** |
| BoW + BiLSTM | 71.4% | 85.2% | 87.0% |
| Ensemble BoW | 68.8% | 79.0% | 81.2% |
| Ensemble BiLSTM | 72.0% | 85.2% | 86.2% |
| Ens. BoW+BiLSTM | **72.2%** | 85.4% | 85.4% |

Table 1: Testing accuracies of the implemented models. Values are in % and rounded to the first decimal.

| | Random | Pretrained | |
|---|---|---|---|
| Models | Tuning | Freeze | Tuning |
| BoW | 0.43 | 0.62 | 0.64 |
| BiLSTM | 0.45 | **0.79** | 0.75 |
| BoW + BiLSTM | 0.49 | 0.69 | **0.77** |
| Ensemble BoW | 0.44 | 0.64 | 0.64 |
| Ensemble BiLSTM | 0.47 | 0.75 | 0.68 |
| Ens. BoW+BiLSTM | **0.51** | 0.71 | 0.72 |

Table 2: Macro F1 metric for the implemented models. Values are rounded to the second decimal digit.

Figure 2: Training and validation accuracies of three models using pretrained embeddings with fine-tuning enabled.




the BiLSTM and the BoW+BiLSTM models tend to learn faster than the BoW model due to the presence of 2 extra layers in the BoW NN and the lower learning rate.

## 4 Analysis

The macro F1 scores in Table 2 show slightly worse performance compared to the corresponding accuracies, which would indicate that the models trained well on larger classes but poorly on smaller classes. Nonetheless, high performance can still be observed across both metrics for the BiLSTM and BoW+BiLSTM singular and ensemble models when operating on pretrained embeddings.

The reductions in performance observed for the BoW models and models trained on randomly-initialised embeddings are to be expected. BiLSTM is capable of extracting sentence structure and word-frequency information, which BoW is not. Likewise, training randomly-initialised word embeddings on a limited dataset over a small number of epochs would not be expected to compare to specially pretrained word embeddings.

As discussed, the BiLSTM and BoW+BiLSTM models show comparable performance on pretrained, tuned embeddings. However, since BiLSTM has the advantage of being a simpler architecture, this is used for our final analysis.

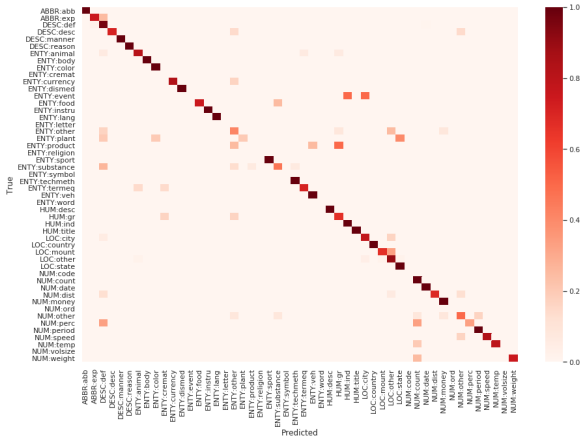



Figure 3: BiLSTM Confusion Matrix for the Test Set

The final accuracy of this chosen model is 87.2%. Further experiments on the BiLSTM model with the removal of numerical tokens (87.6%) and disabling of the lowering step (87.6%) did not show any significant improvements.

Figure 3 shows a confusion matrix for the BiLSTM model on tuned, pretrained word embeddings, normalised by row to account for differences in class sizes. It can be seen that classification across most classes is highly performant; of the 10 weak diagonal cells, only 1 is due to a failure to classify accurately, and the remaining 9 are due to a lack of representation of these labels in the test set.

The vertical lines on the far-left and bottom-right of the matrix demonstrate which questions were most difficult to classify. Multiple classes were misclassified as DESC:def and multiple NUM classes were mistaken for NUM:count, respectively. ENTY:product was frequently mistaken for ENTY:other, ENTY:veh, or HUM:gr. Analysis of these questions show a moderate amount of ambiguity even to a human classifier, hence such errors are to be reasonably expected. For example, the BiLSTM model failed to classify the question ”what county is modesto, california in?” whose label is ”LOC:city”, classifying it as ”LOC:other”.

## 5 Conclusion

DL-based QC can play an important role in QA systems. In this paper, we implemented several models and their respective ensemble models based on BoW and BiLSTM. We conducted experiments on multiple combinations of model configurations, analysed our results across multiple metrics and found that our models demonstrate good performance. Our selected BiLSM model achieved 87.2% accuracy and a macro F1-score of 0.75, with most classes shown to classify well in the confusion matrix. As future work, we might consider combining LSTM or BILSTM with convolutional layers, a current state-of-the-art technique, to achieve further improved results.

## References


Abdelghani Bouziane, Djelloul Bouchiha, Noureddine Doumi, and Mimoun Malki. 2015. Question answering systems: Survey and trends. *Procedia Computer Science*, 73:366 – 375. International Conference on Advanced Wireless Information and Communication Technologies (AWICT 2015).

Farhad Soleimanian Gharehchopogh and Yadollah Lotfi. 2013. Machine learning based question classification methods in the question answering systems. *International Journal of Innovation and Applied Studies*, 4(2):264–273.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2015. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Xin Li and Dan Roth. 2002. Learning Question Classifiers. In {*COLING*} *2002: The 19th International Conference on Computational Linguistics*.

Donald Metzler and W. Bruce Croft. 2005. Analysis of statistical question classification for fact-based questions. *Information Retrieval*, 8(3):481–504.

Anna Surkova, Sergey Skorynin, and Igor Chernobaev. 2019. Word embedding and cognitive linguistic models in text classification tasks. In *Proceedings of the XI International Scientific Conference Communicative Strategies of the Information Society*, CSIS'2019, New York, NY, USA. Association for Computing Machinery.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Dell Zhang and Wee Sun Lee. 2003. Question classification using support vector machines. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, page 26–32, New York, NY, USA. Association for Computing Machinery.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM Neural Network for Text Classification. *arXiv preprint arXiv:1511.08630*.

Zhongcheng Zhou, Xiang Zhu, Zhonghe He, and Yinchuan Qu. 2016/12. Question classification based on hybrid neural networks. In *2016 4th International Conference on Electrical Electronics Engineering and Computer Science (ICEEECS 2016)*. Atlantis Press.