Text Classification from Bank Customer Queries

Yumi Heo
230003122
MSc Data Science
yumi.heo@city.ac.uk

1 Introduction

CNN (Convolutional Neural Network) is used for text classification. However, this method cannot capture the context in the text. In this case, RNN (Recurrent Neural Network) is one of the methods used to capture context. Therefore, CNN combined with LSTM (Long Short-Term Memory) will be used for this project, which leverages the strengths of each neural network. Then, the CNN model and the CNN with LSTM model will be compared using the dataset about queries of a bank's customers.

2 Background

From the article by Kim (2014), CNN started to be used for classification tasks in natural language processing. Later, Yang et al. (2016) suggested the use of CNNs and LSTM as one of the model architectures while focusing on hierarchical attention networks for document classification. The CNN with LSTM model was used directly in text classification (Luan and Lin, 2019). As a result, it was confirmed that the performance of the model combining CNN and LSTM was better than the independent CNN model. This experiment will see whether this discovery will yield the same results in a task that requires classifying 77 classes.

3 Proposed methodology

This project will take around 1 to 2 months, starting from this proposal. Many papers introduce the CNN or LSTM model with NLP-related applications such as text classification and further performance studies of models combining CNN and LSTM. In this project, it will be discovered how much better the performance of a CNN with LSTM model is to classify 77 classes than a CNN model and what the hyperparameters of the best CNN with LSTM model are. Before implementing CNN, word vectors will be created as word embeddings. Then, CNN is used to make vectors based on the word vectors. This will be the CNN model. CNN with LSTM model will go further at this stage. LSTM will be integrated from the CNN model.

3.1 Data

The dataset 'BANKING77' was first introduced by Casanueva et al. (2020). It can also be found on Hugging Face. Since there are already classes for the target, annotation won't be needed. This dataset consists of 77 classes, making it suitable for multiclass classification. Furthermore, it has already been split into training and test sets, with 10,003 samples in the training set and 3,080 in the test set.

3.2 What baselines are you considering?

The baseline model will be CNN. Although this neural network is commonly used in computer vision, it can also be applied to text classification. From this baseline, the model to be compared is CNN with LSTM.

3.3 Proposed timeline

This project consists of 6 stages.

- **Data Preprocessing:** 3 days for duration. This stage involves loading the dataset, exploring its structure, cleaning, preprocessing the text data, and encoding the text data for input into the models.
- **Model Building:** 1 week for duration. In this stage, the architecture of the CNN and CNN with LSTM models will be designed and implemented for multiclass classification.
- **Model Training:** 3 days for duration. Model training involves feeding the preprocessed data into the models, optimizing the model parameters using backpropagation and gradient descent, and monitoring the training process by tracking metrics such as loss and accuracy.

- **Hyperparameter Tuning:** 3 days for duration. This stage involves tuning the hyperparameters of the models. This includes experimenting with different architectures, activation functions, learning rates, batch sizes, etc., and selecting the best combination of hyperparameters.

- **Model Evaluation:** 1 day for duration. Model evaluation involves assessing the performance of the trained models on the test set using evaluation metrics such as accuracy, precision, recall, and F1-score. This stage also includes comparing the performance of the CNN and CNN with LSTM models to determine which performs better.

- **Analysis and Interpretation:** 2 days for duration. In this final stage, the results of the model evaluation will be interpreted, and the strengths and weaknesses of each model will be identified. Potential areas for future improvement will also be discussed.

4 Experimental setup and tools

Since CNN and LSTM will be implemented for this project, Keras or PyTorch and Scikit-learn with other libraries in Python will be used to build neural networks and word embedding.

References

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Yuandong Luan and Shaofu Lin. 2019. Research on text classification based on cnn and lstm. In *2019 IEEE international conference on artificial intelligence and computer applications (ICAICA)*, pages 352–355. IEEE.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.