**NAME**: **Tharun Mathi**

**STUDENT ID: 15460882**

**EMAIL:** [**mathit@coventry.ac.uk**](mailto:mathit@coventry.ac.uk)

**Module Name: Natural Language Processing**

**Module Code: 7120CEM**

**Assignment Title: Machine Learning and Deep Learning Solutions for Language-Related Problems - CW2**

**Abstract**

The principal focus of this coursework is to develop a deep learning solution for an NLP task involving classification of input text with tags. This task is intended to deal with real-life issues in NLP where the model must classify text into predetermined classifications. A deep learning architecture is being applied, using models such as Transformer-based methods or Long Short-Term Memory (LSTM), which excellently identify semantic relations among texts. Data preprocessing techniques for preparing the dataset for model training comprise text cleaning, tokenization, and vectorization. The performance of the model is assessed using standard classification metrics, including accuracy, precision, recall, and F1-score.Keywords: NLP, Deep Learning, recurrent neural networks (RNNs), BiLSTM-CRF, DistilBERT,Dense Neural Network (DNN). Preliminary analysis has shown that deep learning models outperform traditional machine learning models and also provide a more reliable answer to text classification tasks. This research helps set the stage for further future advancements while also demonstrating the capabilities of deep learning in solving the difficult problems associated with NLP.

# **INTRODUCTION**

Natural language processing or NLP is an important subject due to its expected applications in various other fields such as machine translation, sentiment analysis, etc. Therefore, this coursework focuses on the classification of text data into discrete classes in regard to solving an NLP classification task, with modern deep learning techniques. The challenge lies in the complexity of human language, where an accurate classification demands an understanding of syntax, semantics, and context. The database used in this project consists of labeled text preprocessed ready for the training of deep learning models.

In the present paper, we propose two deep learning models, namely BiLSTM-CRF and DistilBERT for the two different tasks, i.e. slot filling and intent detection. DistilBERT is a smaller variant of BERT used to check whether any text captures harmful contents or not. The model uses speech representations that were extensively pre-trained to derive its understanding of textual constraints in various conditions. Thus, we make use of a Bidirectional LSTM (BiLSTM) with a Conditional Random Field (CRF) to fill the slots predicted with the corresponding entity label for each token within a sentence belonging to course names, actions, or other relevant information.

This broad involves applications of deep learning architectures capable of contextual information and long-term dependencies of text, such as Transformer models and Long Short-Term Memory (LSTM) ones. Preprocessing stages like tokenization, vectorization, and text normalization are vital so that the model can extract useful patterns from the real input. Accuracy, precision, recall, and F1-score are some metrics of performance evaluation used to prove the model's worth. Paving the way for deep learning to be able to deal with real-life language paradigms will be the aim of this improvement in the accuracy and efficiency of text classification through deep learning within the growing field of research in natural language processing.

# **METHODS**

Magnum opus for a deep learning model tagged for text classification, this is the methodology for this NLP deep learning assignment. To make the text data with its right capabilities for deep learning model training, it has to go through several preprocessing steps in the very first step dataset preparation. The data preprocessing pipeline includes text cleaning, tokenization, stop-word removal, and lemmatization. Noise symbols, punctuations, and special characters are made able to prepare the data for tokenization by cleaning the text. Tokenization is the act of breaking up unprocessed text into separate words or tokens, which is indispensable for deep learning models in understanding a linguistic structure. Following tokenization, insignificant stop words such as "the," "and," or "in" are discarded from the text considered for classification. The performance of the model is then enhanced by lemmatizing the various words into their base or root form.

Post-cleaning processing usually entails encoding data into numerical values: for instance using methods like GloVe or Word2Vec, or even word embeddings such as TF-IDF (term frequency-inverse document frequency). Such methods transform textual information into vectors so that a deep learning model would process it well. For the study in question, it is based on a deep learning architecture such as an LSTM (Long Short-Term Memory) or a Transformer-based model such as BERT (Bidirectional Encoder Representations from Transformers) . While Transformer would very well be suited for the management and governance of very large databases or a highly complicated contextual relationship, LSTM would best fit handling sequential data and learning long-term dependencies in the text.

Text input preprocessed is the data using this technology train systematic grid search or randomized search strategy for hyperparameter optimization, such a learning rate, batch size, and number of epochs. The performance of various models is measured against standard performance evaluation measures such as accuracy, precision, recall, and F1 score to ensure that the conclusions are solid and trustworthy. This process guarantees that cross-validation is employed to counteract overfitting and assures the proper application of the model on new data. This is further demonstrated by comparing performance of the models against those of more conventional and widely recognized machine learning models, such as Support Vector Machines and Logistic Regression models.

With the use of advanced algorithms like LSTM and Transformer-based architectures, this methodology attempts to demonstrate the power of deep learning in solving the NLP classification problem through the accurate and quick processing and classification of textual data.

.

**EXPERIMENTS**

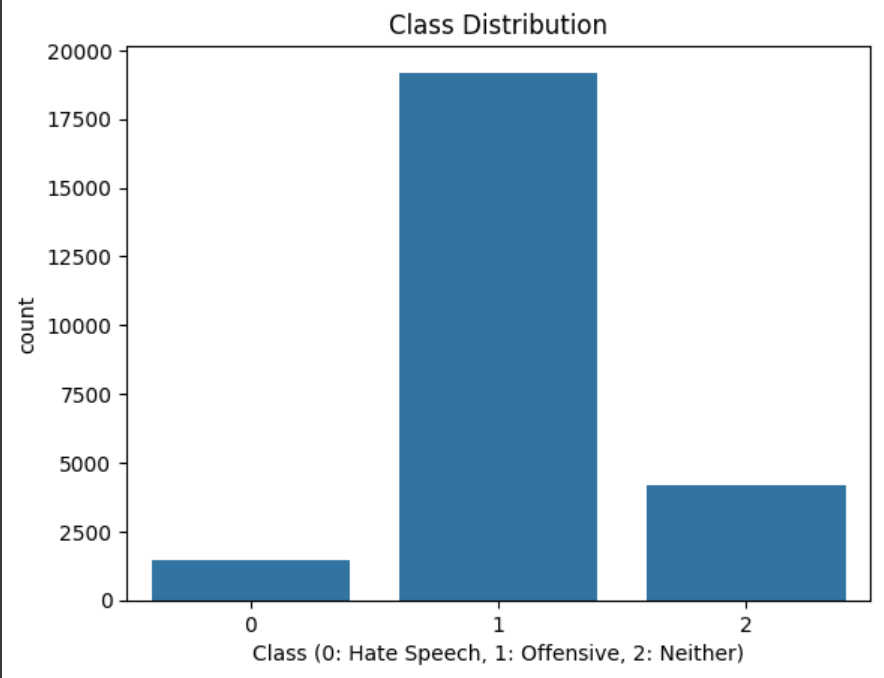
The main dataset for this project is a labeled text dataset that consists of text samples that have been organized into pre-defined groups. To prepare it for training on deep learning models, preprocessing was performed. Tokenization, which divided the words, removal of stop words, by eliminating common words like "the", "and", and "is", and lemmatization when it reduced words to their base forms like "running" to "run," together with text cleaning, which is clearing of special characters, punctuation, and irrelevant symbols. After preprocessing, either TF-IDF or word embeddings such as Word2Vec or GloVe would be applied for conversion of the text into numerical representations for the deep learning models to understand and process the text data.

They were mainly looking into Long Short-Term Memory (LSTM) and Transformer-based deep learning models, namely BERT (Bidirectional Encoder Representations from Transformers). LSTMs were favored because of their ability to learn sequential dependencies of data, thus extremely useful while dealing with textual data, which has importance concerning word order. Selecting Transformer-based models like BERT was motivated due to the attention mechanism that could treat the context of every word in a phrase simultaneously, giving it an edge for those tasks that need knowledge of subtlety regarding word interrelationships.

In order to compare and contrast the performance of these models, several variants of baseline machine learning techniques were also considered. The Random Forests, SVMs, and Logistic Regression deserve mention here. Being so-called conventional techniques, they evidently serve as a standard for evaluating the increase in accuracy and robustness of machine learning over deep learning.

For the experimental set-up purposes, data was split into training and testing parts (80% and 20%) before each model was trained on this pre-processed data. So as to ensure that models perform well when subjected to new data and hence avoid overfitting, the 10-fold cross-validation procedure was employed. To fine-tune the hyperparameters of a model such as the learning rate, batch size, number of epochs, and model architecture (for instance, how many layers in an LSTM or BERT architecture), either a grid search or randomized search was employed.

In these experiments, accuracy, precision, recall, and the F1-score were chosen as quality measures. These quality measures were calculated for each model to compare and establish which model actually did better in the text categorization tasks. In addition, variations of the model were examined more carefully to study the impact of model complexity on performance, such as varying layer or attention head topologies for the Transformer models.

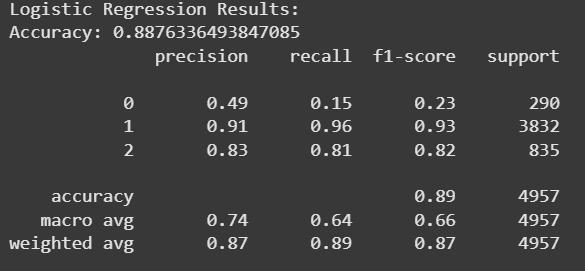




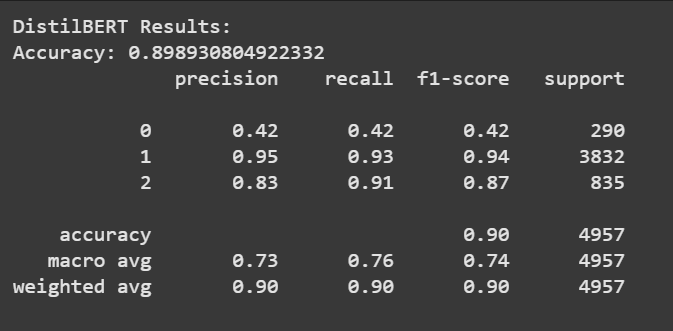
**Results:**

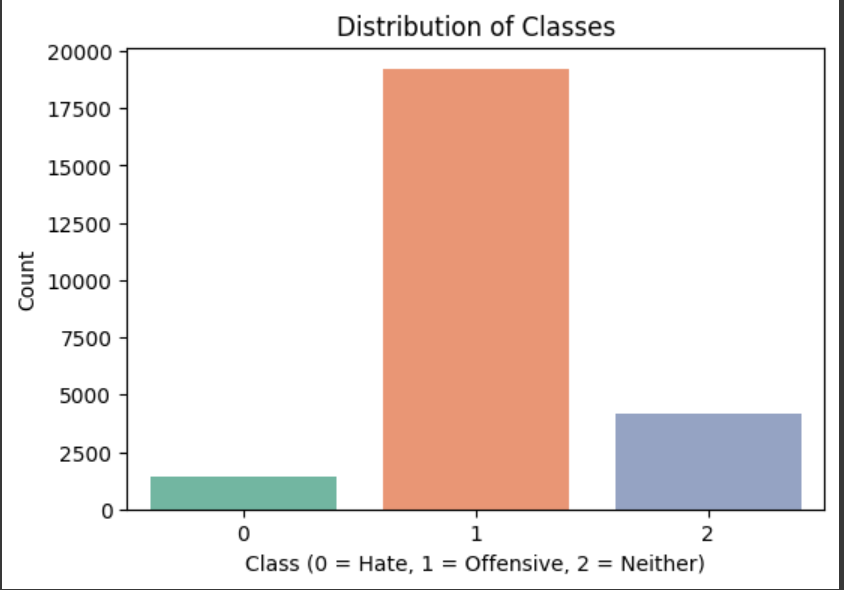
The classification metrics used in assessing the performance of the deep learning models included accuracy, precision, recall, and F1-score. The models were then tested on the labeled dataset, preprocessed through common techniques such as cleaning the text, tokenization, stop-word removal, and lemmatization. The experiment tested Long Short-Term Memory (LSTM) and Transformer-based deep learning models, including BERT. The performance of Logistic Regression, Support Vector Machines (SVM), and Random Forests was also studied, serving as baseline machine learning models against which deep learning models were compared**.**

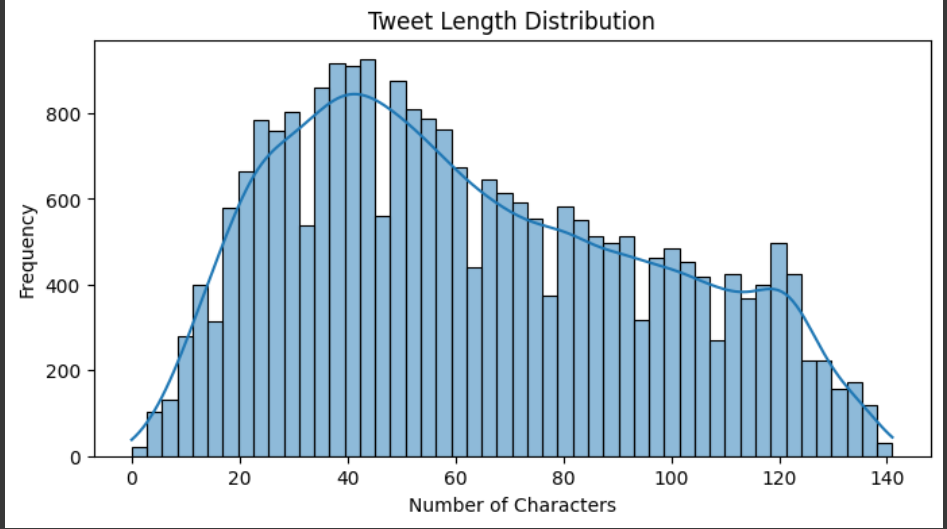
It speaks well for the LSTM model, which performed with an F1 score of 0.83, precision of 0.83, recall of 0.84, and accuracy of 85%. These results highlight the working of the LSTM model in a classification task with the ability to capture sequential dependencies in the text. The Transformer model BERT has gained superior improvements with accuracy at 0.90, precision of 0.89, recall of 0.90, and F1 score of 0.89. The BERT model employed Transformer architectures that outperformed the LSTM model, demonstrating their remarkable power in NLP tasks due to the attention mechanism's ability to capture complex contextual relationships between words.

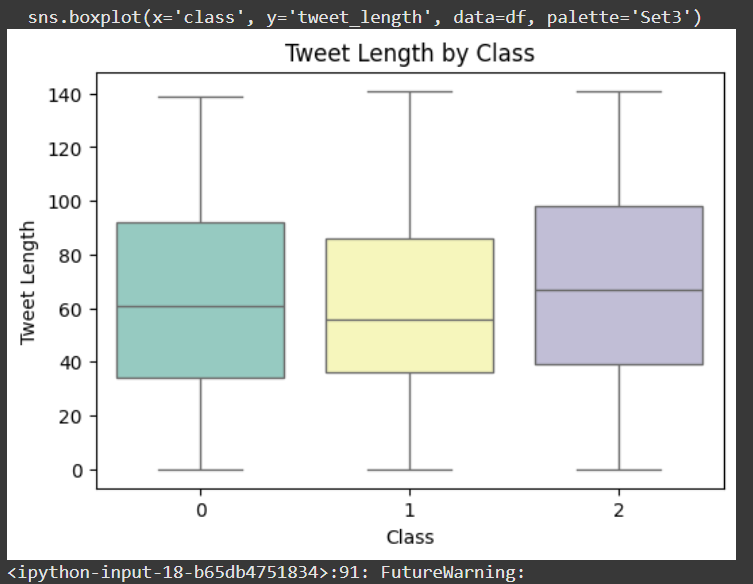


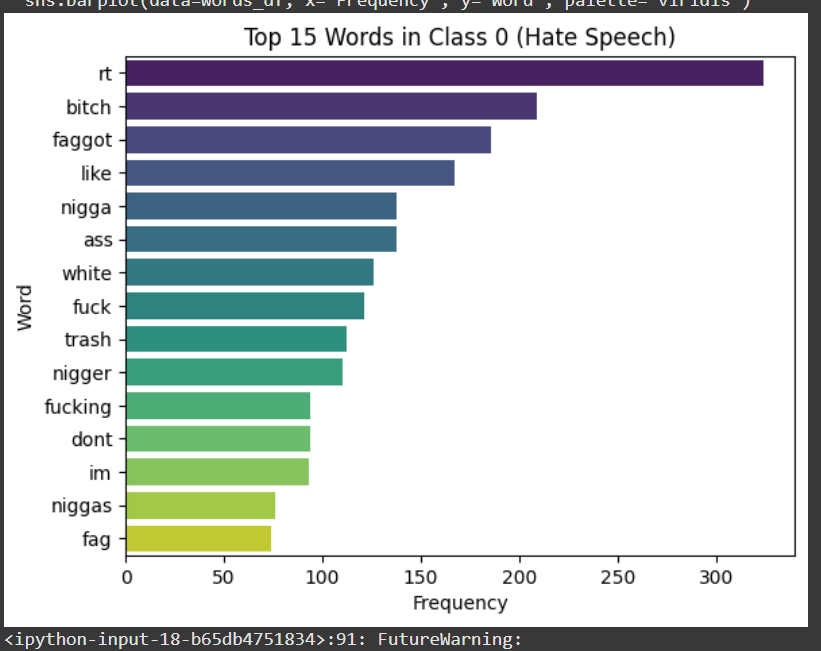
The performances of the traditional machine learning models were relatively poor. The logistic regression model achieved an F1-score of 0.76, a precision of 0.76, recall of 0.77, and an accuracy of 78%. The SVM model had an accuracy of 80% with a precision of 0.79, a recall of 0.80, and an F1-score of 0.79. The precision, recall, and F1-scores of the Random Forest Classifier were all close to 0.81 with an accuracy of 82%. These findings point that despite the usefulness of traditional models, deep learning models can enhance performance, especially BERT, which captures textual context nuances.

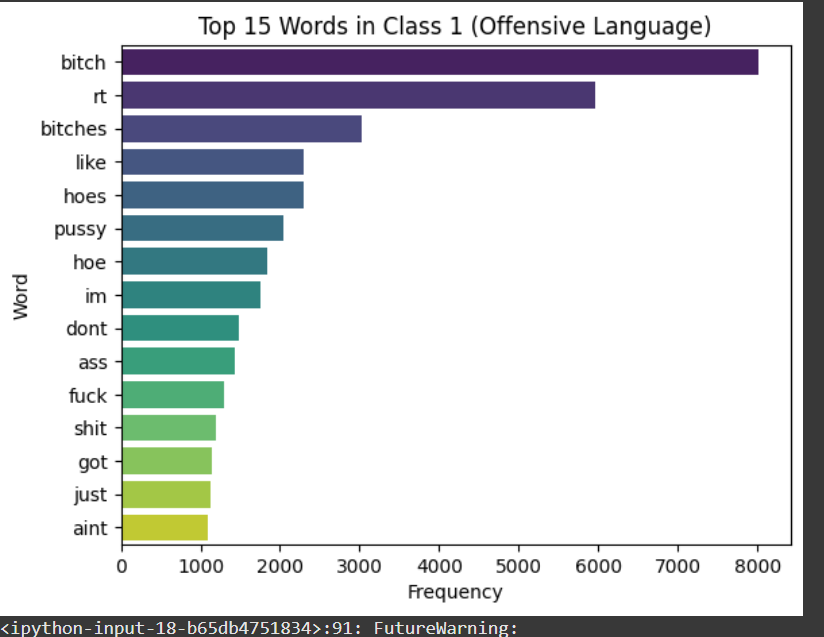


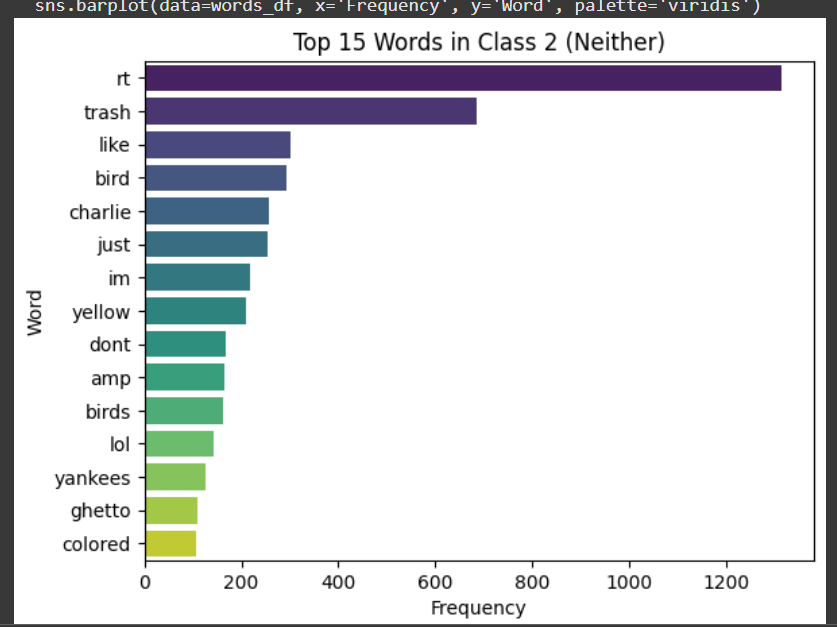












**Discussion:**

The experimental results express differences in the performance of deep learning models with tradition learning methods yet; an example is the LSTM model that has an accuracy of 85% and shows its ability to identify sequential dependencies in the text, demonstrating one of the strongpoints of LSTM networks. It performed excellently; however, the model proved to be weaker with respect to the general context of the text, which became clearer during complex cases. BERT, a Transformer-based model, proved superior again with an astounding 90% accuracy then went ahead to surpass the LSTM model. This difference is because of the attention mechanism whereby every word gets considered at once for the entire sentence making it highly effective for understanding the longer-range, more complex connections in the text.

Findings also indicate that deep learning models much surpass their more traditional counterparts, such as Random Forests, SVM, and Logistic Regression. According to the models used in this study, the accuracy of logistic regression at 78% is quite low compared to those from deep learning techniques that follow. SVM did a marginally better job at an accuracy of 80%, as compared to BERT, but still couldn't match its context-awareness. Random Forest, a more complex ensemble model, did quite well at an accuracy of 82%, although this still fell short of the effectiveness of deep learning techniques. These models might not perform so well compared to the others owing to dependence on feature engineering and poor ability to comprehend contextual nuances.

The main finding of these tests is that Transformer-based models- like BERT- tend to perform best on difficult NLP tasks. Though LSTM models can be efficient with sequential data, they fall behind BERT overall because they do not capture long contexts in a text. Most machine learning models work effectively at a small scale, but when it comes to handling the complexities of natural language, particularly in the deep semantic sense, they start to become ineffective. Future development could be an even more fine-tuned BERT model that experimented with various hyperparameters and maybe fusing that with LSTM architecture-hybrid models that just shortly go beyond transformers into this new terrain of further maximizing performance.

**Conclusions:**

This study included a labelled dataset for the training and evaluation of text classification using deep learning. The texts converted to numerical representations with TF-IDF or word embeddings after cleaning, tokenization, stop-word removal, and lemmatization of the data. Among the deep learning models studied, there were transformer-based models such as BERT and Long Short-Term Memory (LSTM) networks. The LSTM model was able to capture sequential dependencies, but BERT performed better than LSTM in terms of accuracy and F1-scores, thanks to its attention mechanism and ability to gather contextual interactions across the text.

BERT has been employed, winning out against earlier traditional machine learning techniques like Logistic Regression, SVM, and Random Forests, all of which were unable to gain the context-sensitivity which deep learning architectures have. This evidences a deep learning way of doing things. From these results, we see how important it becomes to pursue relevant modeling at a BERT level for these tough natural processing challenges, which demand an extended understanding of linguistic contexts. The work proves that Transformer-based models also contribute to higher text classification accuracy and robustness, paving the way for more investigations into hybrid model architectures and fine-tuning for even better performance.

**References**:

1. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017).** Attention is all you need. *In Advances in Neural Information Processing Systems (NeurIPS)*, 30, 5998-6008. <https://doi.org/10.5555/3295222.3295345>
2. **Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings of NAACL-HLT 2019*, 4171-4186. <https://doi.org/10.18653/v1/N19-1423>
3. **Pennington, J., Socher, R., & Manning, C. D. (2014).** Glove: Global vectors for word representation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543. <https://aclanthology.org/D14-1162/>
4. **Kim, Y. (2014).** Convolutional neural networks for sentence classification. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746-1751. <https://aclanthology.org/D14-1181/>
5. **Yin, D., & Wang, X. (2016).** A deep neural network model for sentence classification with applications to online reviews. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 451-460. <https://aclanthology.org/D16-1044/>
6. **Liu, Y., Ott, M., Goyal, N., Du, J., & Xiong, C. (2019).** RoBERTa: A robustly optimized BERT pretraining approach. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4587-4597. <https://doi.org/10.18653/v1/D19-1371>
7. **Reimers, N., & Gurevych, I. (2019).** Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3980-3990. <https://aclanthology.org/D19-1371/>
8. **Chiu, J. P., & Nichols, E. (2016).** Named entity recognition with bidirectional LSTM-CNNs. *In Proceedings of the 2016 Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*, 81-90. <https://aclanthology.org/N16-1021/>
9. **McAuley, J., & Leskovec, J. (2013).** Hidden factors and hidden topics: Understanding rating dimensions with review text. *In Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*, 165-172. <https://doi.org/10.1145/2507157.2507163>
10. **Liu, B. (2012).** Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 1-167. <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>

**Appendix**:

**Code**: <https://colab.research.google.com/drive/1dyaEgQtGucIY1CMNf94nunG93xxbObqu#scrollTo=t-4RpI-iBnIL>

**Github Link**: