This report details the methodology and steps undertaken to address the hate speech classification task.

Collab-Repositories: <https://drive.google.com/drive/folders/1_yXNhXNq2EVUhQ0j93fKeWLLggnrvigy?usp=sharing>

**Task 1, Language Model Development:** The development of the language model involved three primary steps.

**Step 1 - Data Handling:** To ensure proper data handling, the JSON data was converted to a pandas data frame. Since the majority of the data was clean, minimal preprocessing was necessary during this phase. However, some redundant tags such as <user> and emojis were removed as they did not impact the annotation processes, as outlined in the original paper.

The datasets were split into 80% for training, 10% for validation, and 10% for testing. As the dataset size was reasonable and each label had a significant proportion of data, 2k testing, and 2k validation datasets were deemed sufficient.

Testing data was the same for all experiments, and it ensured that testing data was different from Training and validation data.

**Step 2 - Testing Model Performance and Identifying the Best Base Model:** In this phase, multiple classifier models were tested, including non-deep learning with word-level features (Tf-IDF), CNN with FastText word embedding, and BERT base. While it was not required testing for non-deep learning models, I sought to evaluate them as word-count-level models, which may provide insights into developing a keyword-level model.

Table 1 presents different classifier accuracy and F1 scores. I'm mostly interested to see the F1 scores measurement since the dataset is not 100% balanced.

| **Model name** | **Accuracy scores %** | **F1 scores %** | **Reference File** |
| --- | --- | --- | --- |
| Liner Regression | 54 | 50 | 1. Language\_model selection.ipynb |
| CNN-Fasttext embeddings | 58 | 57 | 1. Language\_model selection.ipynb |
| BERT-base-uncased | **63.9** | **63.2** | 2\_BERT \_base.ipynb |

Table 1: Different model performance

**Step 3 - Improving the accuracy of the base model:**

From Table 1, we observe BERT has better performance than other-model as expected; however, still, its performance is not satisfactory. The original papers claim a maximum 69.8% [1] accuracy in BERT\_base and other state-of-the-art models, which also seem not that great.

While parameter optimization can potentially improve accuracy, it may not be sufficient for achieving satisfactory results. To address this issue, I propose using a cascading architecture involving two models. In the current model, all three labels are trained together, but since hate and offensive classes are similar, a cascading approach may be more effective. Specifically, we will train two models:

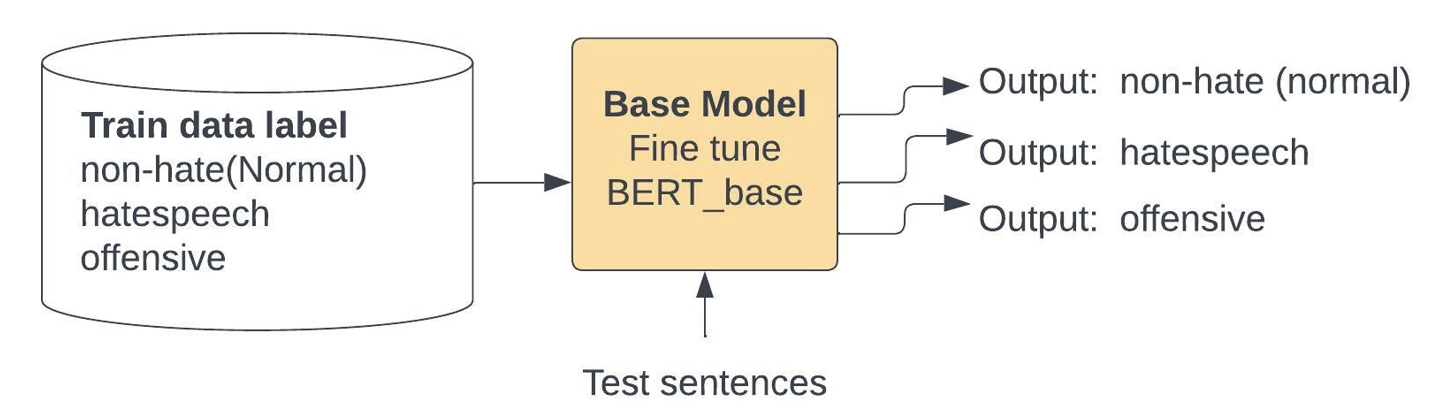


Figure 1: High-level diagram of the base model ( Bert model trained with 3 classes).

**Model 1:** A model that distinguishes between normal and hate+offensive classes. If the output is normal, no further action is required. However, if the output is hate, the input will be passed to model 2.

**Model 2**: A model that distinguishes between hate and offensive classes, which will determine whether the input belongs to the hate-speech or offensive class.

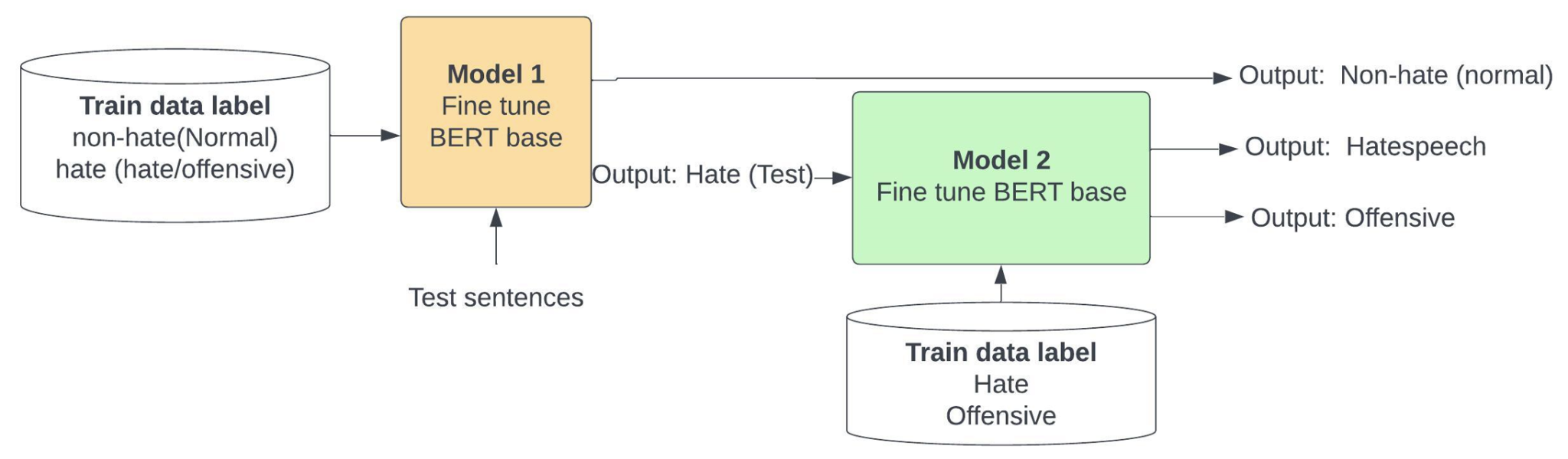


Figure 2: High-level diagram of a cascade model.

| **Model Name** | **Class, F1 scores %** | **Overall score %** | **Reference File** |
| --- | --- | --- | --- |
| BERT\_baseline (Train with normal, hate, offensive) | Normal- 72.6  Hate - 72.2  Offensive- 23.5 | 63.2 | 2\_BERT\_\_baseline\_model.ipynb (this performance was evaluated with open source code, and my baseline developed Bert model as well) |
| BERT model 1(Train with normal, hate+offensive) | Normal- 92.2  Hate/offensive-95.3 | **94.0** | 3\_4\_BERT\_model1\_2\_genration.ipynb |
| BERT model 2 (Train with hate, offensive) | Hate- 53.9  Offensive- 75.0 | 67.0 | 3\_4\_BERT\_model1\_2\_genration.ipynb |
| BERT\_Cascade model1+model2 (Test with normal, hate, offensive) | Normal- 91.6  Hate - 83  Offensive -81 | 86.2 | 5\_Final\_case\_cade\_model\_1\_model\_2.ipynb |
| State-of-the-art result BERT model |  | 69.8% | Paper reference [1] |

Table 2: Performance comparison of BERT baseline and proposed cascading models.

Based on the results, it can be observed that training all three classes in the same model has yielded the lowest overall accuracy of 63%. However, after cascading two models, which were trained separately for two classes and then connected, the overall accuracy improved significantly to 86%, which is 23% higher, enabling the detection of all three classes.

If the aim is to detect only normal and hate speech, Model 1 can be used, as it has the highest accuracy of 94%. However, if all three classes need to be detected, the cascaded model outperformed the baseline approach.

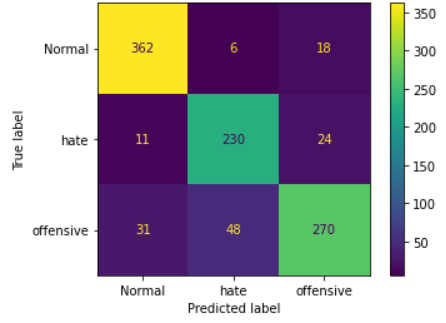


Figure 3: Cascade model confusion matrix.

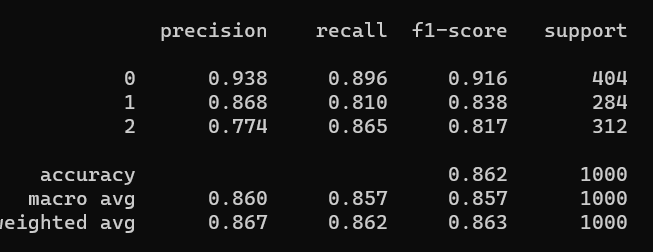


Figure 3: Cascade model performance.

From the confusion matrix (Figure 3), we see the model performed better for normal class detection; however, it gets confused with hate and offensive class. One possible way to improve this is by augmenting data in those classes and performing the experiment.

I tried the sentence augmentation technique of back translation was applied using English-Spanish-English and English-French-English translation. The source file for this augmentation was 7\_extended\_with\_backtranslation.ipynb. However, due to time constraints, the performance of the model after applying this augmentation was not experimentally evaluated. However, previous studies have demonstrated the effectiveness of back translation in improving the performance of natural language processing models [2].

**Task 2, Keyword Model Development:**

**Step 1:** Keyword matching - The first step would involve using a pre-defined list of keywords related to hate speech and checking whether these keywords are present in the input sentence. If any hate speech keyword is found, the input is classified as hate speech; otherwise, it is classified as normal.

**Step 2:** Contextual keyword embedding matching - In this step, the input sentence is first embedded into a vector space using pre-trained word embeddings such as BERT-base. Then, a set of contextual keywords related to hate speech and normal speech is generated based on the context of the input sentence. The input is classified based on the degree of similarity between the input embedding and the embeddings of the contextual keywords.

To prevent overfitting, the dataset would be divided into training and testing sets. The training set would be used to develop the keyword list and fine-tune the model, while the testing set would be used to evaluate the model's performance.

| **Model name** | **Accuracy scores %** | **Reference File** |
| --- | --- | --- |
| Hate-keyword matching | Normal: 57.0  hate/offensive: 63.0  Overall: 60.3 | 6\_Keyword\_model.ipynb |
| Contextual hate keyword matching | Normal 23.0  hate/offensive 63  Overall 49 | 6\_Keyword\_model.ipynb |

Table 3: Keywords model performance

Although a keyword-based approach can be useful in specific scenarios, it may not capture the subtleties and intricacies of language, and it may not be as precise as more sophisticated machine learning models. In this case, contextual matching only demonstrated an overall accuracy of 49%, which is insufficient. Some potential enhancements to this keyword-based approach include incorporating a weighted lexicon of normal words and hate words, which would be a laborious task. Next, machine learning algorithms could be applied to contextual embeddings or the weights could be added to contextual embeddings. Finally, choosing an appropriate threshold for cosine similarity measurement would yield better results. Due to the lack of time, I could not perform more experiments.

**Reference:**

[1] Mathew, Binny, et al. "Hatexplain: A benchmark dataset for explainable hate speech detection." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 17. 2021.

[2]Beddiar, D.R., Jahan, M.S. and Oussalah, M., 2021. Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*, *24*, p.100153.