NLP Multitasking: Named Entity Recognition, Sentiment and Topic Classification

Task description

In this project, three Natural Language Processing (NLP) techniques are applied: Named Entity Recognition (NERC), Sentiment Analysis (SA), and Topic Classification (TC). These techniques are used to analyze a set of test sentences.

The goal in the NERC task is to identify and classify famous named entities such as persons, locations, and organizations at the word level.
The goal in the SA task is to determine the sentiment of each sentence in the test set and categorize it as positive, neutral, or negative.
The goal in the TC task is to determine the topic of each sentence and categorize it as either book, movie, or sport-related.

Data description

NERC

The CoNLL-2003 training dataset contains over 203,000 tokens across 14,041 sentences [1], while the test set is significantly smaller with only 216 tokens across 15 sentences [2]. As shown in Figure 1 and Figure 4, the training data is heavily imbalanced, with over 83% of tokens labeled as ‘O’, compared to 74% in the test set. Because of this, we randomly remove tokens labeled as ‘O’ from the training data to prevent the model from becoming biased toward predicting the dominant ‘O’ class.

The NERC label formats differ between the two sets (e.g., 'B-PER' in training vs. 'B-PERSON' in testing), as visible in the two NERC distribution graphs. These labels were normalized during preprocessing.

The training data contains detailed POS information, as seen in Figure 3, while the test set contains no POS tags at all, shown in Figure 6. For consistency, POS features were removed from the training set.

Figure 2 and Figure 5 show that average sentence lengths are similar between both sets (around 14–15 tokens), though the test set contains fewer and more evenly distributed sentences.

Figure 1

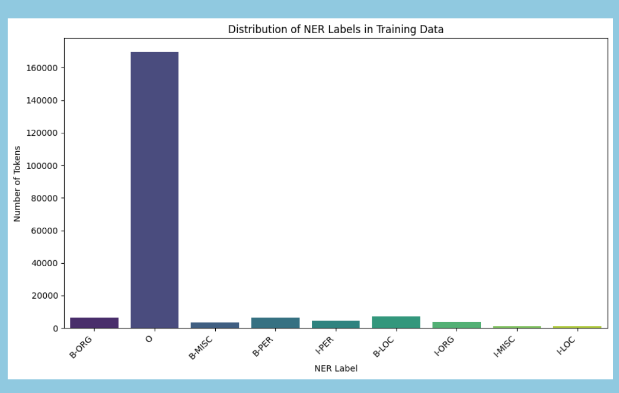


Figure 4

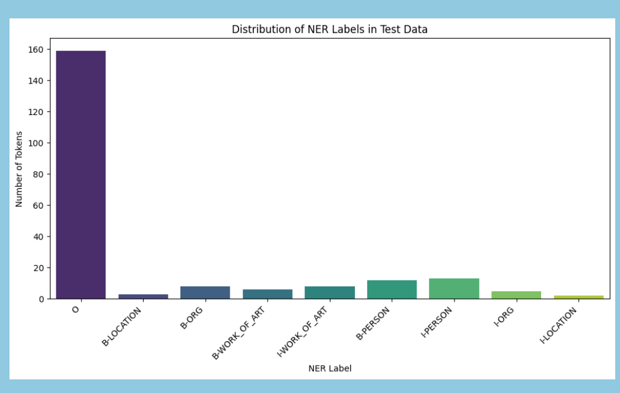


Figure 2

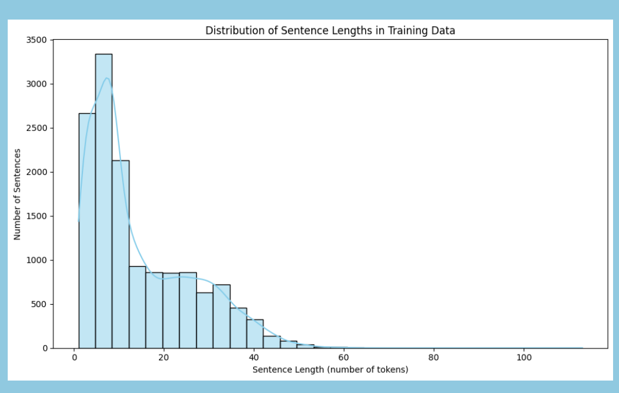


Figure 5

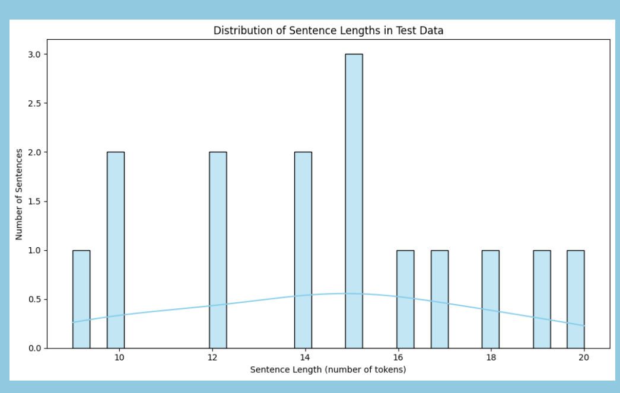


Figure 3

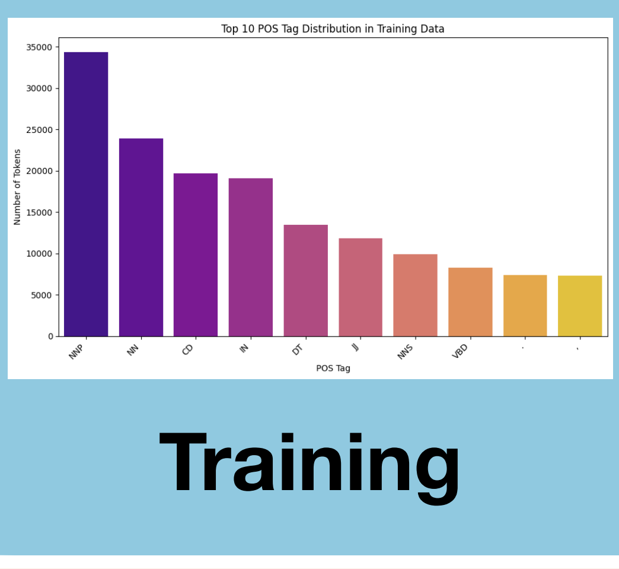
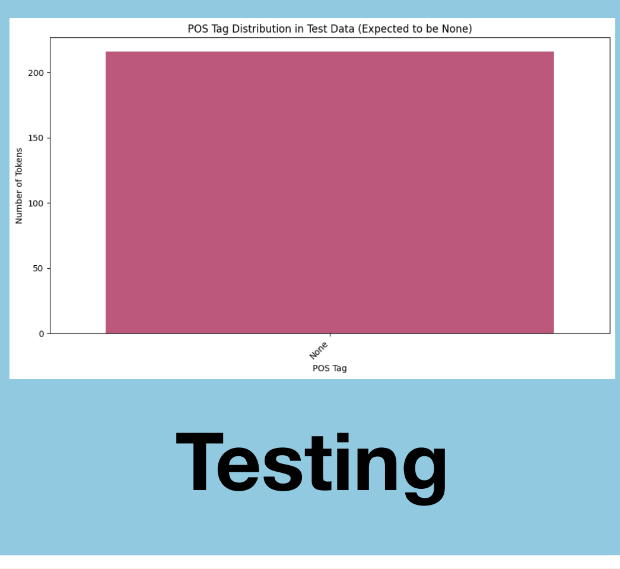


Figure 6



Training **Testing**

Figure 15

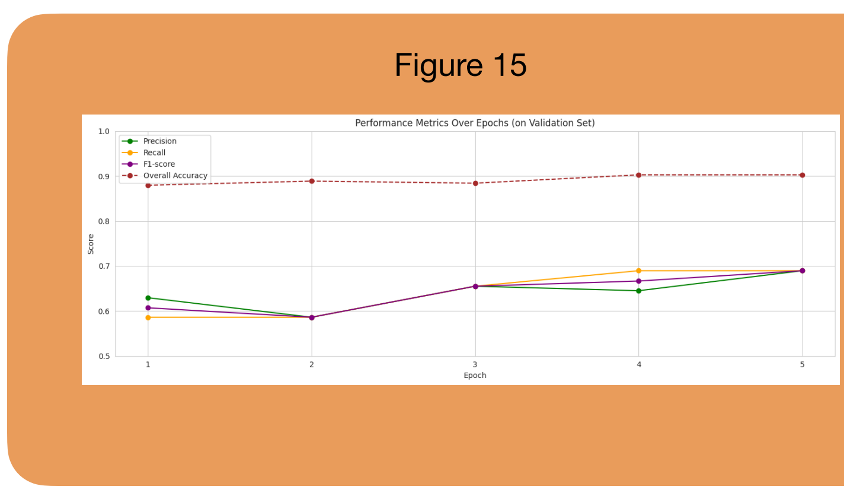
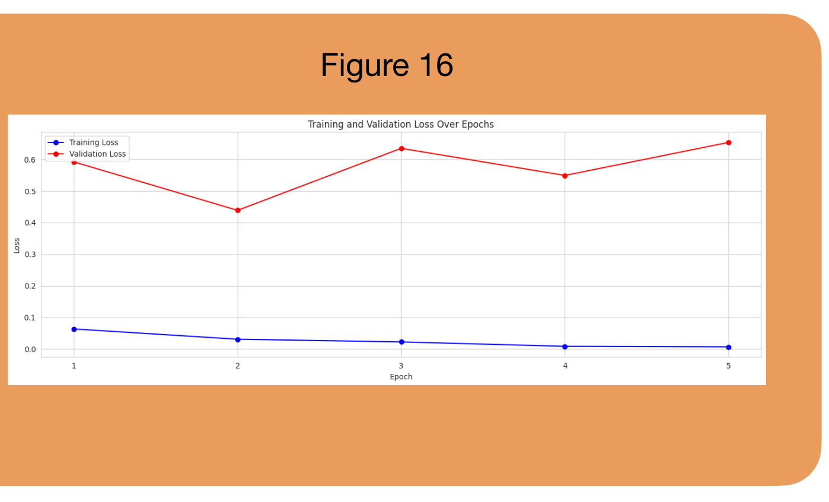


Figure 16



SA & TC

For sentiment classification, the SST-1 dataset was chosen because it is clean, comprehensive, and includes all three required sentiment classes [3]. For topic classification, the Kaggle Topic Analysis dataset was selected because it is large and contains the same topic labels as the test set [4]. Both datasets are relatively balanced and well-structured, as shown in Figure 7 and Figure 9.

The sentiment-topic test set contains 18 evenly distributed sentences across all sentiment and topic classes [5], as shown in Figure 11 and Figure 12.

Although minor class imbalances and sentence length differences exist between the training and test sets, they were not substantial enough to justify correction. Preprocessing steps such as lowercasing, lemmatization, normalization, and tokenization were applied to all data. Overall, the datasets were suitable for training.

Training SA **Training TC**

Figure 7

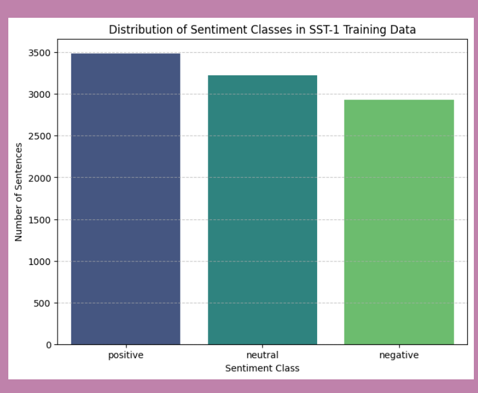


Figure 9

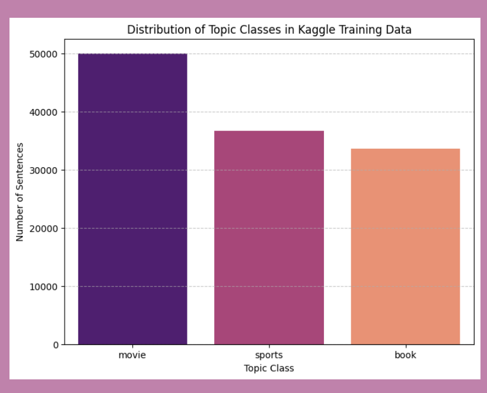


Figure 8

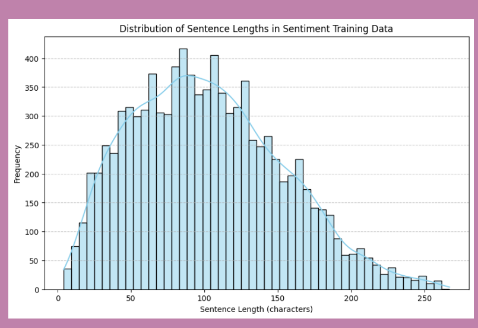


Figure 10

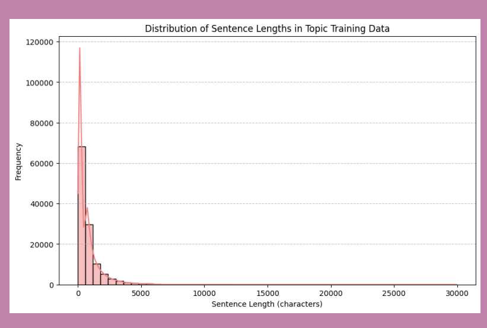


Figure 11

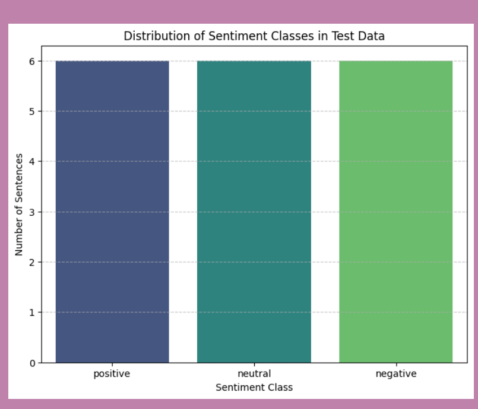


Figure 12

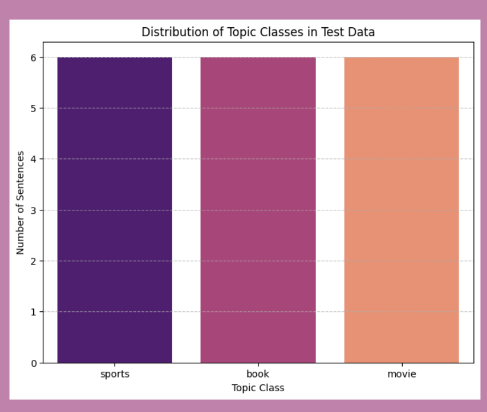
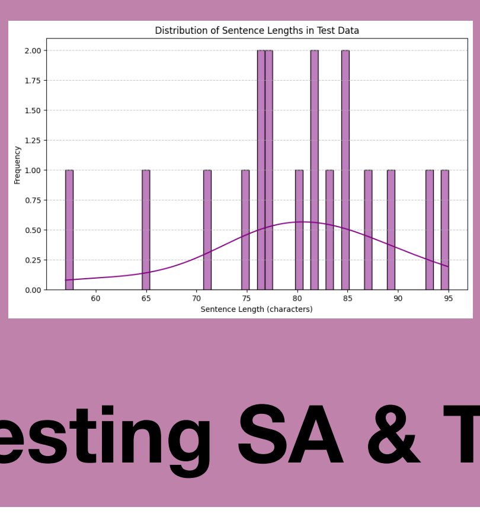


Figure 13



Testing SA & TC

Approach & Motivations

The general pipeline is as follows: dataset loading and data preprocessing -> model training -> testing and evaluation.

NERC

- For NERC, the CoNLL-2003 dataset was used for training. It was chosen as it is a popular and comprehensive dataset specifically designed for NERC.
- The test dataset was provided by the course authors.
- Then, we used a combination of methods for preprocessing to give the model as much context as possible for the best results. These methods were chosen as they are proven to be effective and increase the performance of models.
    - NER tags were normalized across the training and testing datasets, as they had mismatching label and feature names.
    - Tokens and labels from both datasets were extracted, grouping them by sentence.
    - Additional enriched features were extracted for each token in both datasets (token, case, digit/alpha, hyphen, prefixes/suffixes).
    - Because the test dataset didn’t include POS (parts of speech) tags, we had to remove them from the train dataset too to match.
    - Word embeddings were generated for each token in both datasets using a pre-trained Google News Word2Vec model [6].
    - The enriched token features were vectorized using DictVectorizer to create sparse feature matrices for both training and test sets.
    - The sparse feature matrices from DictVectorizer were horizontally stacked with the dense word embedding matrices for both sets.
    - Corresponding label arrays were prepared for both final datasets.
    - (For BERT model) Sentences were additionally tokenized using AutoTokenizer, and labels were aligned with the tokenized output to match BERT input requirements.
- Then, we trained two models on the datasets: LinearSVC and BERT. BERT was chosen as it is a specialized NLP model, and LinearSVC was chosen as a simple and fast algorithm to compare results against [7].
- For LinearSVC, ‘balanced’ class_weight was used to avoid class imbalance issues. Regularization parameter C was set to 1.0, as a starting point. 1000 max_iterations were chosen to allow for convergence.
- For BERT, standard values were chosen, such as a learning rate of 2e-5, batch size of 16, and 5 epochs, as they proved sufficient.

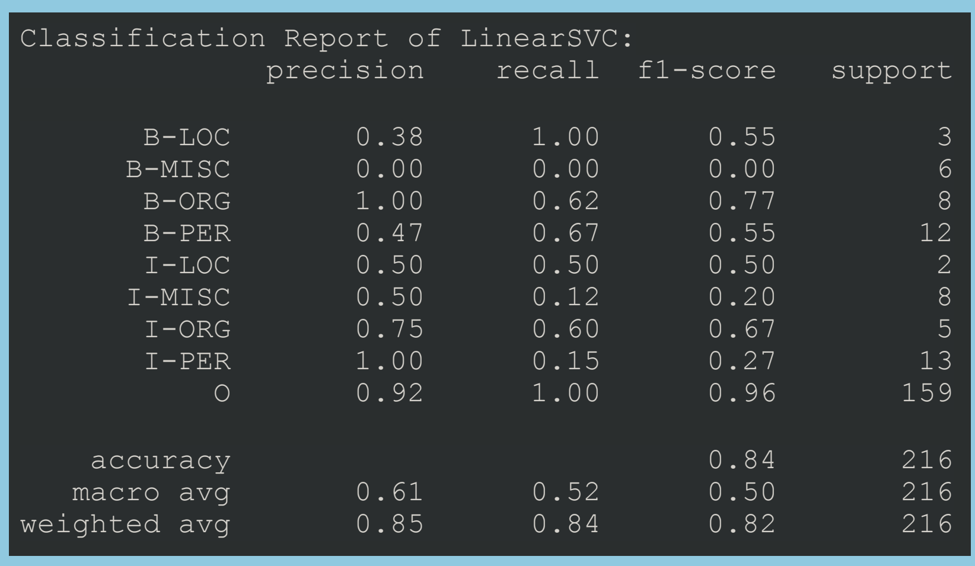
SA & TC

- For SA, the SST-1 dataset was chosen, as it is a comprehensive dataset. Crucially, we downgraded from SST-2, as it excluded the ‘neutral’ class, which was required, as it was included in the test dataset.
- For TC, a dataset from Kaggle was used, which included the precise labels from the test dataset: book, movie, and sport.
- The test dataset was provided by the course authors.
- Then, we used a combination of methods for preprocessing to give the model as much context as possible for the best results. These methods were chosen as they are proven to be effective and increase the performance of models.
    - Lowercasing: Convert text to lowercase.
    - Emoji removal: Remove emoji characters.
    - Punctuation removal: Remove punctuation marks.
    - Number removal: Remove numerical digits.
    - Whitespace normalization: Replace multiple spaces with a single space and remove leading/trailing spaces.
    - Tokenization: Split text into individual words or tokens.
    - Stop word removal: Remove common stop words that don’t carry any sentiment.
    - Lemmatization: Reduce words to their base or root form.
    - TF-IDF vectorization: Convert text tokens into numerical feature vectors.
- Then, for both tasks, we train a LogisticRegression classifier model on their respective datasets. It was chosen because of its efficiency, good performance, and because it’s common practice for this type of task.
- For the parameters, the ‘liblinear’ solver was chosen because of its stability on medium-sized datasets and fast convergence time. Regularization parameter C was set to 1.0, as a starting point. 1000 max_iterations were chosen to allow for convergence.
- Afterwards, we compared and evaluated the results obtained from the classification reports generated after the test runs.

Results discussion & Analysis

NERC

- Erroneous predictions
    - At first, we didn’t normalize NER tags, which led to erroneous results, as most classes had a value of 0.0, as they were unseen by the model during training.
    - Additionally, we didn’t fix ‘O’ class imbalance initially, which led to incorrect results, as the scores were skewed by the good performance solely on the ‘O’ class.
    - Also, we improved performance significantly by using additional feature extraction methods, which we didn’t use at first.
- Final results



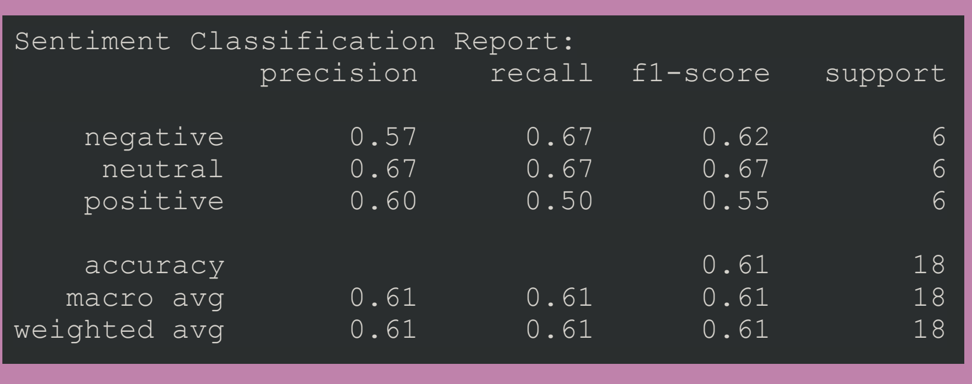
```
Classification Report of LinearSVC:
              precision    recall  f1-score   support

       B-LOC       0.38      1.00      0.55         3
      B-MISC       0.00      0.00      0.00         6
       B-ORG       1.00      0.62      0.77         8
       B-PER       0.47      0.67      0.55        12
       I-LOC       0.50      0.50      0.50         2
      I-MISC       0.50      0.12      0.20         8
       I-ORG       0.75      0.60      0.67         5
       I-PER       1.00      0.15      0.27        13
           O       0.92      1.00      0.96       159

    accuracy                           0.84       216
   macro avg       0.61      0.52      0.50       216
weighted avg       0.85      0.84      0.82       216
```

The updated LinearSVC model shows a slight overall improvement, with accuracy increasing from 0.83 to 0.84 and macro F1-score rising from 0.45 to 0.50. Notably, there is a strong improvement in B-ORG (F1-score up from 0.36 to 0.77) and partial recovery for I-MISC (F1-score now 0.20). However, the model still completely fails to detect B-MISC, and I-PER recall dropped sharply despite perfect precision, which suggests inconsistent entity recognition. While the aforementioned changes to the model parameters and data preprocessing show progress, especially in some underperforming classes, the model's performance is still insufficient for reliable NERC use.

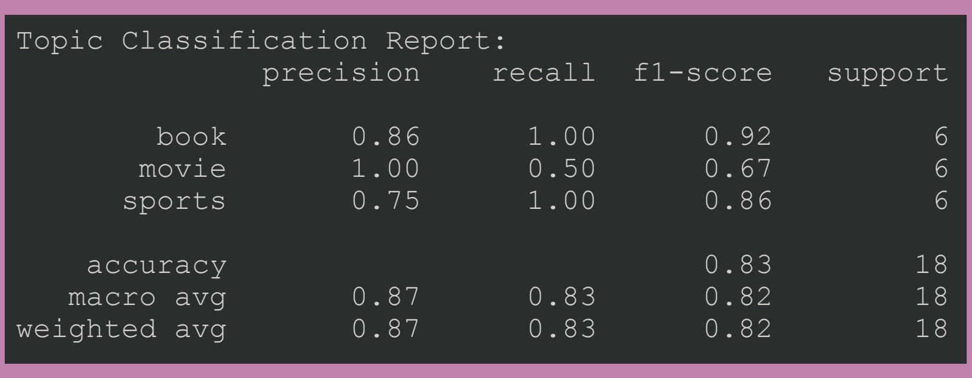
The BERT model shows a significant improvement over the previous LinearSVC approach, with accuracy rising from 84% to 90.3% and F1-scores jumping from 0.50 (macro avg) to around 0.70. Most notably, it finally succeeds in recognizing the previously unidentifiable MISC entities (F1-score 0.50) and significantly improves performance on PER (F1 0.73) and LOC (F1 0.86). Even though ORG F1 dropped slightly, BERT provides more balanced recognition overall. These results suggest a much more reliable NERC model.

SA & TC

- Erroneous predictions:
    - At first, we used SST-2 dataset, which doesn’t have a ‘neutral’ class, which led to erroneous results with 0.0 values.
    - After switching to SST-1, which has a float value in the range 0.0-1.0 to represent the class, we set such thresholds for the classes (0.0 <= negative <= 0.4 <= neutral <= 0.6 <= positive <= 1.0), that it resulted in the neutral class not being predicted during testing. After we adjusted the thresholds, we obtained correct results
- Final results



```
Sentiment Classification Report:
              precision    recall  f1-score   support

    negative       0.57      0.67      0.62         6
     neutral       0.67      0.67      0.67         6
    positive       0.60      0.50      0.55         6

    accuracy                           0.61        18
   macro avg       0.61      0.61      0.61        18
weighted avg       0.61      0.61      0.61        18
```



```
Topic Classification Report:
              precision    recall  f1-score   support

        book       0.86      1.00      0.92         6
       movie       1.00      0.50      0.67         6
      sports       0.75      1.00      0.86         6

    accuracy                           0.83        18
   macro avg       0.87      0.83      0.82        18
weighted avg       0.87      0.83      0.82        18
```

For the topic classification, the model performed strongly, achieving 83% accuracy and a macro F1-score of 0.82. It showed excellent precision and recall for the ‘book’ and ‘sports’ classes (both with 100% recall), while the ‘movie’ class had high precision (1.00) but low recall (0.50), missing half of the actual movie-related sentences. The confusion matrix confirmed this, showing that 2 ‘movie’ sentences were misclassified as ‘sports’ and 1 as ‘book’. Overall, the model successfully identifies book and sports topics, but it slightly struggles with movie-related content and needs improvement.

Conclusions & Limitations

In conclusion, the models performed well overall. As expected, BERT outperformed LinearSVC in the NERC task. LogisticRegression worked effectively for topic classification but showed weaker performance in sentiment analysis.

Improvements could include tests of current models with different parameter combinations and tests of other models. Additional data and deeper error analysis could help enhance performance. Additional data preprocessing approaches can also be explored, such as different feature extraction strategies.

Division of work

Tim was responsible for finding the datasets, performing analysis, and handling preprocessing. Sami selected and implemented the models, tuned the parameters, and analyzed the results. The poster was created collaboratively.

Source code

https://drive.google.com/drive/folders/10HcJLV0fzr3J9UZ-pcNF_CDX8ngAWn54?usp=sharing

References

[1] E. Tjong Kim Sang and F. De Meulder, 'Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition', in *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003. Dataset accessed via Hugging Face: https://huggingface.co/datasets/eriktks/conll2003. [Accessed: June 1, 2025]

[2] NERC Test Dataset. Provided as course material for *Text Mining*, Vrije Universiteit Amsterdam, 2025. Available via Canvas: NER-test.tsv. [Accessed: June 1, 2025]

[3] S. Socher, A. Perelygin, J. Wu, et al., *Stanford Sentiment Treebank* (SST-1), Dataset accessed via Hugging Face: https://huggingface.co/datasets/stanfordnlp/sst. [Accessed: June 1, 2025]

[4] Kaggle, 'Topic Analysis Dataset: sport, book, movie', [Online]. Available: https://www.kaggle.com/datasets/aronferencz/topic-analysis-dataset-sportbookmovie.[Accessed: 01-June-2025].

[5] VU Canvas, 'Sentiment-topic-test.tsv dataset', [Online]. Available: sentiment-topic-test.tsv. [Accessed: 01-June-2025].

[6] Hugging Face, 'fse/word2vec-google-news-300', [Online]. Available: https://huggingface.co/fse/word2vec-google-news-300. [Accessed: 01-June-2025].

[7] D. Meyer, F. Leisch, and K. Hornik, '*The support vector machine under test*', *Neurocomputing*, vol. 55, no. 1–2, pp. 169–186, Sep. 2003. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231203004314. [Accessed: 01-June-2025].

Authors

Timofei Polivanov, 2808108
Sami Rahali, 2771157
Group 41