Contribution

Luca Rosic: 1) coded topic classification, 2) analyzed topic classification results 3) topic classification poster part
Matt Hrkal: 1) code sentiment analysis, 2) analyzed sentiment, 3) sentiment topic & data analysis
Thomas Norton 3: 1) coded NERC, 2) analyzed NERC, 3) NERC poster part

Sentiment classification

Methods

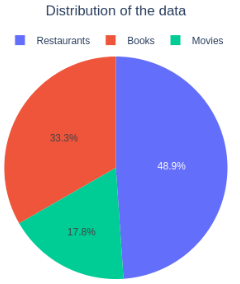
In this part, we perform sentiment classification using pre-trained transformer models (BERT, DistilBERT, RoBERT, and AlBERT) to analyse reviews from various domains.Sentiment classification is a natural language processing task that aims to determine the polarity of a given text, typically as positive, negative, or neutral. During training, early stopping is employed to prevent overfitting, and Weights & Biases is used for logging model performance and hyperparameters. By comparing these models, we can determine the best-performing model for sentiment analysis across different review sources.

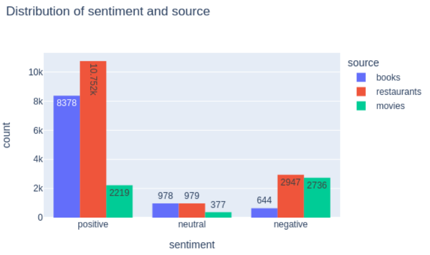
Datasets

For the sentiment classification, we have combined three datasets containing reviews of books, restaurants, and movies from various sources. The datasets were preprocessed to ensure consistency and ease of analysis.

The preprocessing steps include reading the datasets from various file formats like JSON, TSV, and SQLite, and selecting relevant columns such as review text and ratings. Rows with missing values are removed, and numerical ratings are converted to sentiment labels (negative, neutral, or positive) using a custom function. Columns are renamed for consistency across datasets, and a new column is added to indicate the source of the data, such as books, restaurants, or movies.

After preprocessing, the datasets were combined into a single DataFrame, which was then saved as a CSV file (combined_train_sentiment.csv). This combined dataset serves as the basis for further analysis and model training in the sentiment analysis task.

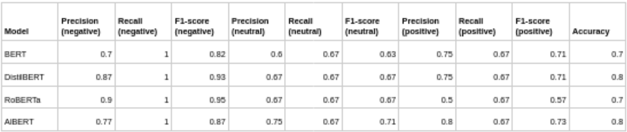




Results and suggestions

In summary, DistilBERT and AlBERT performed better than BERT and RoBERTa in this sentiment classification task. The differences in performance could be attributed to various factors such as model architecture, pre-training strategy, tokenization method, and the size of the training dataset.

To improve the models performance, fine-tuning on a larger, balanced dataset, addressing class imbalance with techniques like oversampling (SMOTE), and experimenting with different learning rates, training epochs, or optimization algorithms could be applied to achieve better generalisation and higher classification accuracy.



Model	Precision (negative)	Recall (negative)	F1-score (negative)	Precision (neutral)	Recall (neutral)	F1-score (neutral)	Precision (positive)	Recall (positive)	F1-score (positive)	Accuracy
BERT	0.7	1	0.82	0.6	0.67	0.63	0.75	0.67	0.71	0.7
DistilBERT	0.87	1	0.93	0.67	0.67	0.67	0.75	0.67	0.71	0.8
RoBERTa	0.9	1	0.95	0.67	0.67	0.67	0.5	0.67	0.57	0.7
AlBERT	0.77	1	0.87	0.75	0.67	0.71	0.8	0.67	0.73	0.8

Topic Classification

Method

Topic Classification is a NLP technique to takes texts and determine which predefined label relates to it. The methods used for this topic classification task include SVM and BERT transformers. SVM is seen as a generic supervised learning algorithm that can classify texts by finding optimal divisions in the training set features (lab 2.2)(Multi-category news classification using Support Vector Machine based classifiers). The features used in the SVM was the TF-IDF scores of tokens for classification, preventing generic words from seeming of the same importance as unique related to the topics (e.g. 'was' being as important as 'restaurant')(Lecture 3: Machine Learning nlp part 1). BERTs feature input is the unprocessed text and the gold labels. BERT pre-trains on a generic schema and then fine-tunes to the training set(Transformers: State-of-the-art natural language processing). Transformers use contextual embedding, which reduces ambiguity by taking surrounding tokens into account(lecture 4 - Sentiment Analysis).

Related to how small the test set is, comparing the 2 methods is to see if the basic nature of SVM is good enough to Classify accurately or if a large model technique such as transformers is necessary.

Data Used

Apart from the test set given a training set was created. For the test, three datasets on Kaggle were used and processed into a single training set with 3000 data points(1000 for each review topic). This is used to fine tune BERT and train SVM. This allows the models to determine how the features inputted relate to the topics.

This data was imported, removed all columns except the text column, added the column 'topic' and input topic, select 1000 rows from each dataset, and then appended the datasets rows together to from 'train-dataset1.csv'.

Dataset sources:
Restaurant reviews
https://www.kaggle.com/datasets/fahadsyed97/restaurant-reviews?select=precovid_reviews.csv
Book reviews
https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews
Movie reviews
https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

SVM

```
              precision    recall  f1-score   support

           0       0.67      1.00      0.80         2
           1       1.00      0.67      0.80         3
           2       1.00      1.00      1.00         5

    accuracy                           0.90        10
   macro avg       0.89      0.89      0.87        10
weighted avg       0.93      0.90      0.90        10
```

BERT

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         3
           2       1.00      1.00      1.00         5

    accuracy                           1.00        10
   macro avg       1.00      1.00      1.00        10
weighted avg       1.00      1.00      1.00        10
```

Analysis and Potential Improvements

BERT achieved a perfect score due to its pre-training and contextual embedding. SVM predicted one review incorrectly, resulting in a lower precision score for book, a lower recall score for restaurant and a lower accuracy than BERT. In the sentence that was a restaurant review but was classified as a book review, the token 'diner' (which was the only word associated to a restaurant review) was not located in the training set, so this word could not help classify the review and the rest of the feature were more related to book reviews. To improve SVM, a function could be used to find the best feature by training different features to train the model or the training set could be expanded.

Named Entity Recognition Classification

Method

NERC is a NLP that seeks to classify named entities in unstructured text. In this section, we used an SVM, SVM with word embeddings and a pretrained tranformer BERT to recognise named entities. We decided to employ multiple methods in order to compare the difference in results between them. While SVM uses supervised learning, BERT is unsupervised and utilises a bidirectional Transformer.

Datasets

To train the models, we used CoNLL-2003. This dataset was chosen as we have already used it for a previous assignment, and as it include PoS and chunk tags with the words. For the word embedding model, we used gensim vector embeddings.

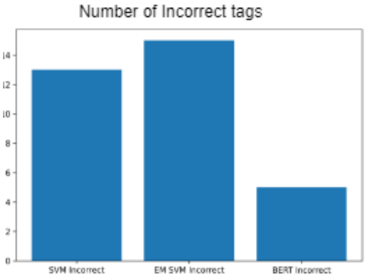
The final test set had to be modified in order to include PoS tags, as it did not include them. This was accomplished using NLTK's built in PoS tagger, then adding these tags as a new row in the dataframe. This was the saved as a new csv file.

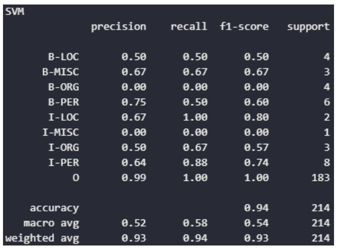
Results

From the number of incorrect tags, BERT performed the best by far. Of the 214 words in the test set, it made only 5 mistakes. SVM and EM SVM performed very similarly, but surprisingly, word embedding seemed to reduce peformance. This is possibly due to the short sentences in the small test set, and context was not very important. BERT had an average of 0.98, whereas SVM had 0.94 and EM SVM had 0.93. The largest difference in results was for macro average, where BERT got 0.92 but SVM/EM SVM got 0.52/0.53.

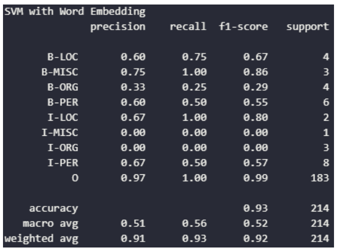
Improvements

Using a larger or different training set would likely be an improvement to this project. This could mean that SVM could potentially perform closer to that of BERT. Another improvement to the project could be to use different pretrained language models such as roBERTa or XLNet which could further improve upon the results of BERT since they are based on it.

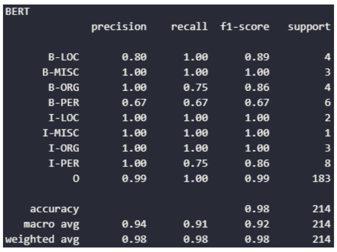




```
SVM
              precision    recall  f1-score   support

       B-LOC       0.50      0.50      0.50         4
      B-MISC       0.67      0.67      0.67         3
       B-ORG       0.00      0.00      0.00         4
       B-PER       0.75      0.50      0.60         6
       I-LOC       0.67      1.00      0.80         2
      I-MISC       0.00      0.00      0.00         1
       I-ORG       0.50      0.67      0.57         3
       I-PER       0.64      0.88      0.74         8
           O       0.99      1.00      1.00       183

    accuracy                           0.94       214
   macro avg       0.52      0.58      0.54       214
weighted avg       0.93      0.94      0.93       214
```



```
SVM with Word Embedding
              precision    recall  f1-score   support

       B-LOC       0.60      0.75      0.67         4
      B-MISC       0.75      1.00      0.86         3
       B-ORG       0.33      0.25      0.29         4
       B-PER       0.60      0.50      0.55         6
       I-LOC       0.67      1.00      0.80         2
      I-MISC       0.00      0.00      0.00         1
       I-ORG       0.00      0.00      0.00         3
       I-PER       0.67      0.50      0.57         8
           O       0.97      1.00      0.99       183

    accuracy                           0.93       214
   macro avg       0.51      0.56      0.52       214
weighted avg       0.91      0.93      0.92       214
```



```
BERT
              precision    recall  f1-score   support

       B-LOC       0.80      1.00      0.89         4
      B-MISC       1.00      1.00      1.00         3
       B-ORG       1.00      0.75      0.86         4
       B-PER       0.67      0.67      0.67         6
       I-LOC       1.00      1.00      1.00         2
      I-MISC       1.00      1.00      1.00         1
       I-ORG       1.00      1.00      1.00         3
       I-PER       1.00      0.75      0.86         8
           O       0.99      1.00      0.99       183

    accuracy                           0.98       214
   macro avg       0.94      0.91      0.92       214
weighted avg       0.98      0.98      0.98       214
```