Group 68 Work Division

D.G.J.K. Linger (2741629): 1) Coding: topic; 2) Analysis: topic; 3) Poster Preparation: overall & topic
F. Moser (2819777): 1) Coding: NERC (CRF, spaCy, BERT), preprocessing; 2) Analysis: NERC performance & evaluation; 3) Poster Preparation: NERC
H. Song (2791848): 1) Coding: sentiment analysis(SVM); 2) Analysis: SVM performance, limitations; 3) Poster Preparation: sentiment SVM, RoBERTa
H. Lokhandwala (2774699): 1) Coding: sentiment (RoBERTa); 2) Analysis: RoBERTa, SVM vs RoBERTa ; 3) Poster Preparation: sentiment preprocessing, RoBERTa methods & setup
Github: https://github.com/AIVU2026/TextMining-project.git

Topic Classification

Data

Selection

The dataset used for topic classification was synthesized from three existing datasets sourced from various websites, and one data-scraped dataset sourced from Reddit. First, the ""movie" data was sourced from an IMDB movie review dataset which was initially used for sentiment analysis. The set consists of a list of ~49000 text values with a sentiment label attached to it. Secondly the "book" data was sourced from an amazon book reviews dataset used initially for sentiment analysis as well. The set consisted of ~4000 unique values each paired to a sentiment label. Thirdly the first "sports" dataset was used from the 20-newsgroup dataset used as a benchmarking set in sci-kit learn, which is a collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups. The particular section of the dataset that was used was from categories rec.sport.hockey and rec.sport.baseball. This data was directly available in a dataframe and consisted of ~1200 documents.

The second sports dataset was scraped from Reddit and carefully curated to align with the content of the test set. Since the test set included references to specific sports tropes and keywords, these were deliberately targeted during data collection to ensure topical similarity. The training data was then compiled into a unified set by combining content from each dataset and assigning topic labels accordingly. As a result, the training sets for the movie and book categories remained the same, while two distinct datasets were used for the sports category, each getting a separate training run.

Preprocessing

First all datasets were tokenized to sentence level using NLTK to split the document, comment and review text-entries into sentences that could be used for training. Next each sentence was "cleaned" using a regular expression protocol that removed all punctuation, as well as being case lowered so that there would be uniform case amongst all sentences. These "cleaned" sentences were then loaded into a pandas dataframe and given a label corresponding to the dataset category that they came from. The same treatment was applied to the given test dataset

The individual categories sets were then merged together so that the smallest set size of the three categories would be the sample size to be randomly sampled from the other two categories. This was to ensure that all topics were balanced and equally represented in the training data. In both training-set synthesis cases sports happened to be the smallest dataset.

For the initial dataset there were "17573" sentences labelled with book, "415194" sentences labelled with movies and "7288" sentences labelled with sports, with this sports being sourced from the 20 newsgroup dataset. Only the sentences longer than 50 characters were taken to ensure that there would be enough topical content in the sentence to remain relevant for classification. This then led to the first combined dataset containing all three categories at a perfectly balanced ratio. The second training dataset was made in the same way but using the data scraped sports dataset instead.

The datasets were then split off into a training and test split at an 80 to 20 ratio, from which the remaining training instances another 10 percent was taken as a validation set. This left ~15000 training samples, and a mock test set of ~4000 samples comprised of data taken from the combined dataset.

Methods

To evaluate performance on the topic classification task, several transformer-based language models were selected. BERT (Bidirectional Encoder Representations from Transformers) served as the baseline, as it is one of the most influential models in natural language processing and provides a strong reference point for comparison.[1]

- ALBERT (A Lite BERT) is a lightweight variant of BERT that reduces model size and training time through parameter sharing and factorized embeddings. It was included to test whether a more efficient architecture can maintain or improve classification performance without sacrificing accuracy.[3]
- MPNet (Masked and Permuted Pre-training) builds upon BERT and XLNet by combining masked language modeling with permuted language modeling, resulting in better dependency modeling and improved performance on sentence-level tasks. It was chosen to test whether its advanced pre-training strategy offers advantages in topic classification.[5]
- DistilBERT is a compressed version of BERT, trained via knowledge distillation. It is significantly faster and smaller while retaining most of BERT's language understanding capabilities. It was included to evaluate how well a resource-efficient model can perform compared to larger architectures.[4]

Together, these models represent a spectrum of design trade-offs in terms of size, speed, and architecture innovations, enabling a comprehensive comparison of their performance on the topic classification task.

The training loop the step size was first calculated by taking the length of the training set and dividing it by the batch size. This allowed for initiation of the validation during training at precise intervals. The validation was implemented at every half an epoch. Furthermore an early stopping method was used using the eval loss as a metric, with a patience of 2, meaning that the algorithm stopped if there was no improvement measured after at most 2 evaluations..

Experimental Setup

To evaluate the performance of various transformer models on topic classification and to determine the impact of domain specific input data on performance, two datasets were prepared—each containing three categories: book, movie, and sports. The primary goal was to investigate how the source and quality of the data, particularly for the sports category, influence model performance.

The experiment was conducted in two phases:
Baseline Dataset:
- The sports category was taken from the 20 Newsgroups dataset, a classic benchmark in text classification. The book and movie categories were compiled from other consistent sources to form a balanced dataset.

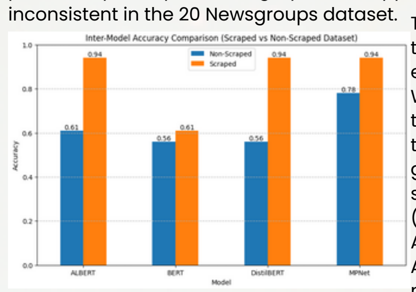
Scraped Dataset:
- To test the effect of domain relevance and content quality, the sports category was replaced with data scraped from Reddit. The Reddit content was selected to better reflect contemporary language use and a style more consistent with the book and movie categories.

BERT: Served as the baseline model for comparison.
ALBERT: A parameter-efficient version of BERT.
MPNet: A model combining masked and permuted language modeling for improved contextual understanding.
DistilBERT: A lightweight, distilled version of BERT designed for faster inference with minimal performance loss.

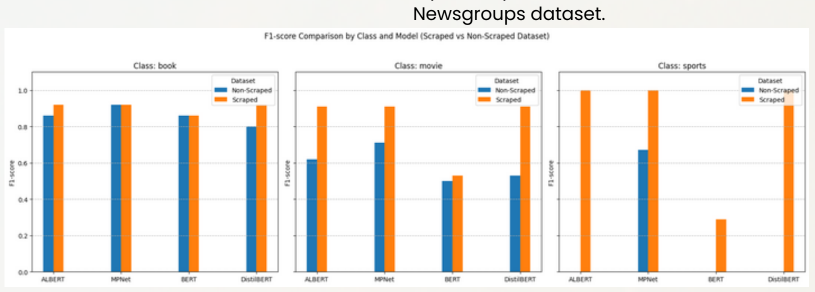
Each model was fine-tuned on the training set and evaluated using precision, recall, F1-score, and accuracy metrics. Performance was compared across both datasets to observe the effect of replacing the sports data with the Reddit-sourced version.

Results

The performance graphs clearly illustrate the impact of dataset quality on model effectiveness in the topic classification task. When trained using the dataset containing the sports category from 20 Newsgroups, the transformer models show mixed and generally suboptimal performance. F1 scores on this dataset range from 0.44 (DistilBERT) to 0.77 (MPNet), with BERT and ALBERT scoring 0.45 and 0.49, respectively. Accuracy values mirror this trend, with models achieving between 0.56 (BERT and DistilBERT) and 0.78 (MPNet). These results suggest difficulty in distinguishing among the classes, particularly the sports category, which appears to be poorly represented or stylistically inconsistent in the 20 Newsgroups dataset.



The performance graphs clearly illustrate the impact of dataset quality on model effectiveness in the topic classification task. When trained using the dataset containing the sports category from 20 Newsgroups, the transformer models show mixed and generally suboptimal performance. F1 scores on this dataset range from 0.44 (DistilBERT) to 0.77 (MPNet), with BERT and ALBERT scoring 0.45 and 0.49, respectively. Accuracy values mirror this trend, with models achieving between 0.56 (BERT and DistilBERT) and 0.78 (MPNet). These results suggest difficulty in distinguishing among the classes, particularly the sports category, which appears to be poorly represented or stylistically inconsistent in the 20 Newsgroups dataset.



Discussion & Analysis

The experimental results show that transformer-based models can achieve strong performance in topic classification tasks when provided with high-quality, domain-consistent data. Across both datasets, MPNet consistently ranked among the top performers, while BERT generally lagged behind, reaffirming the benefits of architectural improvements such as permuted masking and deeper dependency modeling introduced in MPNet. When trained using the original 20 Newsgroups sports data, all models suffered from limited performance, with F1 scores ranging from 0.44 to 0.77 and accuracy from 0.56 to 0.78. However, performance sharply improved across the board—especially for ALBERT, MPNet, and DistilBERT—when the sports category was replaced with data scraped from Reddit. These models reached F1 scores and accuracy values as high as 0.94, showing that the data source and quality had a significant effect on learning outcomes.

Limitations

While the results are promising, there are several limitations that must be acknowledged. First, hyperparameter tuning was limited, with most models trained using default or minimally adjusted parameters. This lack of fine-tuning may have suppressed the full performance potential of certain models, especially BERT, which is known to be sensitive to learning rate, batch size, and training duration.

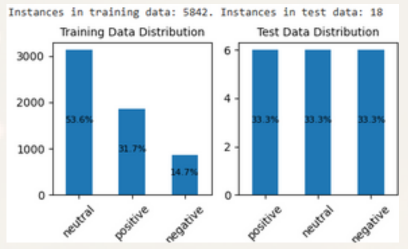
Secondly, no random seed was set during training, meaning that the experiments lack strict reproducibility. Transformer models initialize their weights stochastically, and the absence of controlled seeding introduced notable variation between runs. As a result, individual model scores could fluctuate depending on the initialization and training dynamics, which weakens the reliability of any single evaluation. Future work should address this by enforcing deterministic behavior and systematically tuning hyperparameters to allow for fairer comparisons and more stable conclusions.

Despite these limitations, the experiment underscores two key findings: Firstly modern, efficient transformer architectures can outperform the BERT baseline even under modest training conditions. Secondly, data quality and domain alignment[2] have a critical impact on classification performance, sometimes more than model architecture itself. These insights can guide future applications where labeled data is scarce, and model robustness across domains is essential.

Sentiment

Data

Selection



The dataset used in this study is the "Financial Sentiment Analysis" dataset, containing over 5,800 entries. Each entry consists of two features: text expressing opinions related to financial topics and the corresponding sentiment label (positive, negative, or neutral). The training data is highly imbalanced: neutral ~53%, positive ~31%, and negative ~14%. The test dataset contains 18 text samples labeled with the same three sentiment classes, where all classes are evenly balanced.

Preprocessing

The text data was cleaned using regular expressions to retain only words and exclamation marks, as these elements are crucial for expressing opinions. The sentiment labels for both the training and test sets were integer-encoded as follows: negative = 0, neutral = 1, and positive = 2. To evaluate the impact of text normalization, we conducted experiments involving stop word removal and lemmatization. Both the training and test datasets were preprocessed using spaCy to lemmatize tokens and eliminate stop words.

We use the RoBERTa and SVM models in this section. For the RoBERTa transformer-based experiments, the text was tokenized, attention masks were generated, and batches were padded to match the length of the longest sentence to ensure proper input formatting. For the SVM experiments, the text was vectorized using TfidfVectorizer from scikit-learn, which has benefits like adjusting the weights of words by evaluating their frequencies throughout the corpus and focusing on informative words by reducing the influence of function words (e.g., articles). The embedded data was converted into numerical arrays per sentence for input into the SVM model.

Methods

We chose SVM as a strong classical baseline for text classification due to its effectiveness on small, high-dimensional datasets. RoBERTa, a state-of-the-art transformer model, was selected to evaluate the benefits of deep contextualized embeddings and large-scale pretraining. This comparison highlights the performance gap between traditional ML and modern NLP on noisy, imbalanced tweet sentiment data.

Support Vector Classification (SVC)

SVC is a classical implementation of the Support Vector Machine (SVM) algorithm, which was widely used before the advent of deep learning models such as BERT and its variants. SVM creates optimal hyperplanes that maximize the margin between classes, calculated using the closest data points (support vectors)[12,13].

RoBERTa

RoBERTa (A Robustly Optimized BERT Pretraining Approach) is an enhanced version of BERT. It retains BERT's transformer-based architecture but removes certain limitations: it is pretrained on larger datasets, for longer durations, with dynamic masking, and without the next sentence prediction task[6]. These improvements make RoBERTa strong in understanding context and semantic differences in language[7]. For this project, we use a RoBERTa model from the Hugging Face Transformers library that has been further pretrained on over 58 million English tweets using three sentiment classes: positive, negative, and neutral.

Evaluations

The results of the experiments were evaluated using the following:

1) Classification Report
- Accuracy: While accuracy may be skewed by class imbalance, it still serves as a straightforward indicator of overall correctness.
- Precision: The ratio of correctly predicted positive observations to the total predicted positive observations.
- Recall: The ability of the classifier to find all relevant positive instances.
- Macro F1 Score: This metric is well-suited to our imbalanced dataset. It calculates the F1 score for each class independently and then averages them, giving equal weight to each class[8].

2) Confusion Matrix

A confusion matrix is a 3×3 table that visualizes classification performance by showing the number of predicted and actual labels for each class.

Experiment Setup

To assess the effectiveness of our sentiment classification models, we performed experiments comparing two different preprocessing pipelines:
- Raw text input: The dataset was used in its original form with minimal preprocessing.
- Processed text input: The dataset was processed using lemmatization and stopword removal.

SVM

We tested the effect of lemmatization and stop word removal with and without preprocessing. Unlike RoBERTa, this model was not re-evaluated across epochs. The SVM hyperparameters used were:
- penalty='l1'
- loss='squared_hinge'
- dual=False
- multi_class='ovr'
- class_weight='balanced'
- max_iter=5000
- C=1.2
- tol=2e-4

RoBERTa

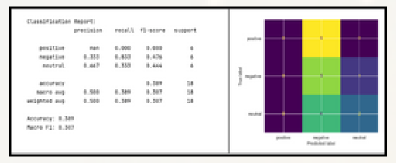
For both experiments, we fine-tuned the RoBERTa model on the training data for 5 epochs using the following hyper-parameters:
- Learning rate: 5e-5
- Batch size: 32
- default parameters of RoBERTa

We tracked accuracy and macro F1 across all five epochs and visualized performance trends using line plots. Final performance metrics from epoch 5 were used to compare the impact of preprocessing.

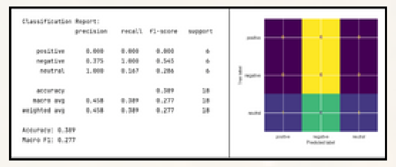
Results

SVM

Without stopword removal & lemmatization



With stopword removal & lemmatization



RoBERTa

Without stopword removal & lemmatization

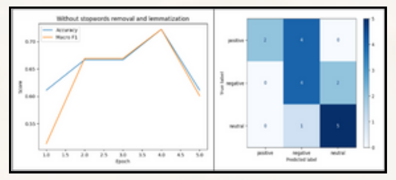
Without stop words and lemmatization. lr=3e-05

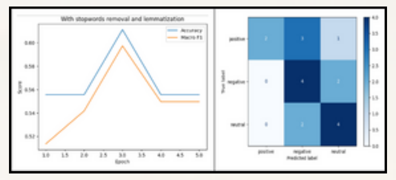
	precision	recall	f1-score	support
0	1.000	0.500	0.667	6
1	0.444	0.667	0.533	6
2	0.667	0.667	0.667	6
accuracy			0.611	18
macro avg	0.704	0.611	0.622	18
weighted avg	0.704	0.611	0.622	18

With stopword removal & lemmatization

With stop words and lemmatization. lr=5e-05

	precision	recall	f1-score	support
0	1.000	0.333	0.500	6
1	0.444	0.667	0.533	6
2	0.714	0.833	0.769	6
accuracy			0.611	18
macro avg	0.720	0.611	0.601	18
weighted avg	0.720	0.611	0.601	18





Discussion & Analysis

SVM

The removal of stop words and lemmatization led to modest improvements. The overall macro F1 increased by 10%, with a notable 30% improvement in classifying neutral sentiment. However, classification performance for negative sentiment dropped by 15%.

RoBERTa

Without preprocessing, accuracy and macro F1 peaked at ~72% by epoch 4 before dropping to ~60%, likely due to overfitting on the small dataset. With preprocessing, both metrics plateaued at ~60% from epoch 3 onward. It appears that lemmatization and stop word removal removed helpful context cues RoBERTa had been pretrained to recognize. RoBERTa still performed reasonably due to its strong pretraining, but the limited and imbalanced dataset hampered its full potential. Both models struggled most with the positive class (low recall), indicating a need for more balanced data.

SVM vs. RoBERTa

RoBERTa clearly outperforms SVM across all key metrics. Without preprocessing, RoBERTa achieves a macro F1 of 0.60 and accuracy of 61%, while SVM only manages 0.27 F1 and 39% accuracy. Similarly, with stopword removal and lemmatization, RoBERTa still maintains higher performance (F1: 0.55, accuracy: 56%) compared to SVM (F1: 0.30, accuracy: 39%). SVM particularly struggles with the positive class, showing 0% recall in both settings—indicating a failure to predict that class altogether. In contrast, RoBERTa maintains moderate performance across all classes, especially neutral. Precision and recall values for SVM are inconsistent, with some classes reaching high recall (e.g., 1.0 for negative) but extremely poor precision, suggesting a bias in predictions. RoBERTa's precision-recall balance is more stable, leading to better F1 scores. Overall, the results confirm that deep transformer-based models like RoBERTa, especially when pretrained on massive sentiment datasets, are far more effective for nuanced tasks like tweet sentiment classification than traditional models like SVM.

Limitations

SVM

Our results show that SVM has significant difficulty identifying positive sentiments despite their 31% representation in the training data. Khan et al. (2024) suggest this underperformance may be due to the nature of the 'TfidfVectorizer', which implements a bag-of-words model that ignores word order[9]. In financial sentiment analysis, word order often matters. Additionally, SVM struggles with non-linear sentiment types such as negation, sarcasm, and domain-specific jargon. These limitations make LinearSVC unsuitable for tasks requiring deeper semantic understanding.

RoBERTa

While RoBERTa handles non-linear semantics better, it was pre-trained on general corpora like Wikipedia, BookCorpus, and news articles—not financial data[13]. As a result, it may miss subtle cues specific to financial language[15]. Future work could consider domain-specific models like FinBERT, introduced by Araci (2019), which adapts BERT to financial text classification by pre-training on finance-specific corpora and extending the model's architecture for improved domain adaptation[16].

NERC

Data

Selection

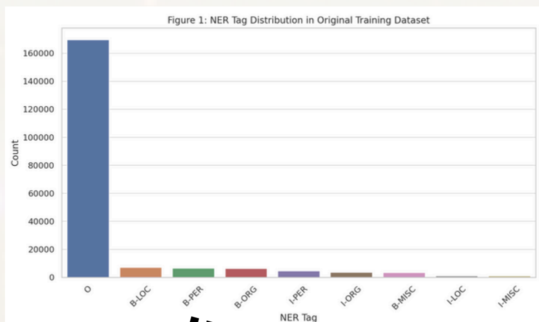
To be able to perform a robust Named Entity Recognition and Classification (NERC) we looked at the tags that are present in the test set and these tags included: "B-PERSON", "I-PERSON", "B-ORG", "I-ORG", "B-LOCATION", "I-LOCATION", "B-WORK_OF_ART", and "I-WORK_OF_ART". For our training dataset we selected the CoNLL-2003 NER training dataset from hugging Face[17], as the source for the supervised learning. This dataset has over 200000 tokens and has a large amount of entity types that are not relevant to our dataset and the target evaluations that we want. Thus we filtered the dataset to only hold the sentences that contained the entity tags that are relevant to make sure that we do not hold noise in the dataset and that the domain is aligned with out model and the test set.

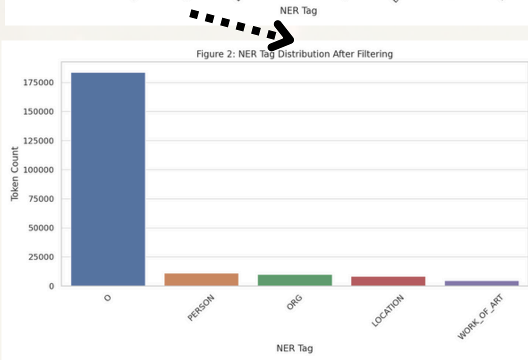
Preprocessing

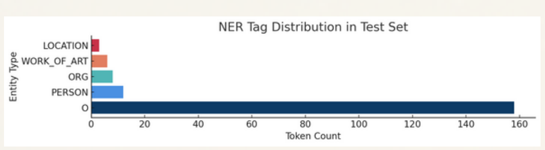
For preprocessing we wanted to normalize tag inconsistencies across the dataset. So we implemented a procedure to transform them.
The ones that were applied were combined "PER" and "PERSON" into "PERSON", combined "LOC" and "GPE" into "LOCATION". We also normalised all the tags to the uppercase BIO formatOther normalisation procedures we implemented were that we also removed the sentences with no target entities, converted the datasets to a BIO tagging scheme. We also extracted and aligned POS tags with nltk.pos_tag [18] for models that need them.The test set contained 233 sentences and 29 labeled tokens which is spread across 4 main classes which includes "PERSON", "ORG", "WORK_OF_ART" and "LOCATION".

Description

The filtered CoNLL training dataset contained over 14000 sentences and 204,030 tokens. Of those tokens 81,7% of them were tagged as "O". The most common named entity labels in the set were "B-Person" and "I-Person" which made up 5.3% and 5.4% respectively. "B-ORG", "I-ORG" and "B-LOCATION" made up 5.8% and "B-WORK_OF_ART" and "I-WORK_OF_ART" made up 0.2% in total. The test set contained 233 sentence and 4740 tokens. 87.6% of those tokens were labeled "O". "PERSON" accounted for 4.1%, "ORG" 3.7%, "LOCATION" 2.3% and "WORK_OF_ART" 2.2%. The underrepresentation of "WORK_OF_ART" in the training set compared to the test set could become a challenge for generalization across the entity types.







Methods

Motivation

For our model we ended up choosing CRF as it is strong in modelling sequence dependencies using features like POS tags and word context which makes it great for structured prediction tasks like NERC [19]. To fix the overfitting, regularization was applied and then later on was extended with syntactic features. BERT was then chosen as it is able to learn contextual relationships without feature engineering that is manually given [20]. Alongside that spaCy's en_core_web_sm was added as more of a lighter pre-trained baseline to compare the transformer based aproaches[21].

NERC-CRF (Tuned Parameters)

This baseline CRF uses contextual word features, capitalization, prefixes, suffixes, digits and character patterns. Overfitting was nulled by L1 and L2 regularisation parameters (c1,c2) [18]. POS tags were also added to the training feature set from NLTK which matched the conll2012 Penn Treebank style.

spaCy Pretrained Model

We used this model as the en_core_web_sm model from spaCy which is trained on the OntoNotes corpus and is optimized for English NER [21]. While it is not tuned for the specific label set it still provides a fast and an easily reproducible baseline available for multilingual pipelines.

Transformers (BERT)

We added the bert-base-NER transformer model from Hugging Face as one of the models which is fine tuned for NER using contextual embeddings. It doesn't use feature engineering and is a high performing benchmark due to its attention seeking modeling [20].

Results

NERC-CRF (Tuned Parameters)

	precision	recall	f1-score	support
LOCATION	0.40	0.67	0.50	3
ORG	0.57	0.50	0.53	8
PERSON	0.60	0.75	0.67	12
WORK_OF_ART	0.00	0.00	0.00	6
micro avg	0.54	0.52	0.53	29
macro avg	0.39	0.48	0.42	29
weighted avg	0.45	0.52	0.47	29

spaCy Pretrained Model

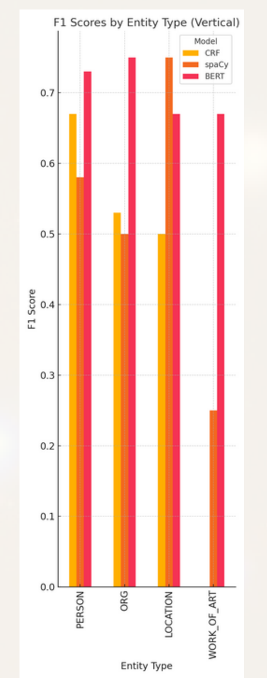
	precision	recall	f1-score	support
LOCATION	0.60	1.00	0.75	3
ORG	0.50	0.50	0.50	8
PERSON	0.58	0.58	0.58	12
WORK_OF_ART	0.50	0.17	0.25	6
micro avg	0.56	0.52	0.54	29
macro avg	0.55	0.56	0.52	29
weighted avg	0.54	0.52	0.51	29

Transformers (BERT)

	precision	recall	f1-score	support
LOCATION	0.67	0.67	0.67	3
ORG	0.75	0.75	0.75	8
PERSON	0.80	0.67	0.73	12
WORK_OF_ART	0.56	0.83	0.67	6
micro avg	0.70	0.72	0.71	29
macro avg	0.69	0.73	0.70	29
weighted avg	0.72	0.72	0.71	29

Discussion and Analysis

The comparison of the NERC models is the study shows that there a large trade offs between the models complexity, generalization and the overall coverage. NERC-CRF despite the regularization and POS features added still showed signs of overfitting, especially towards the memorized "PERSON" names in the training set. It showed promising recall in the common classes like PERSON and ORG, but struggled with the lesser represented labels like "WORK_OF_ART". The POS did help with the structure of the predictions but could not get the semantic distinctions in more advanced or abstract cases. spaCy's pretrained model showed a balanced recall and precision on the "PERSON" and " LOCATION" tags, however it still misclassified and missed rear frequent or more specific entity types. This shows its limited domain adaption when applied to labels distributions it hasn't seen before. BERT was the best out of the bunch on almost all of the entity types, especially "ORG", "PERSON" and "WORK_OF_ART". It showed better recall if the entity boundaries an a contextual relevance due to its deep attention layers. Even though BERT did do better than the other two it still struggled with tags underrepresented in the training dataset like "DATE" and "PRODUCT", which shows the importance of balanced data. In conclusion the transformer models like BERT are very high performing and with its contextual depth it allowed it to get an edge while the CRF was more tuned to structural features and spaCy was more of a benchmark. All model results greatly show the importance of balanced training data fine tuning for the best NERC performance.



Limitations

The key limitation of the models were the imbalance of the entity label distributions, and largely in the underrepresented tags like WORK_OF_ART made it more difficult to be able to learn correctly. While the CRF model benefit from manual features and POS tagging it showed signs of overfitting with the common names. spaCy was fast and efficient but could not consistently recognize the correct tags as it wasn't trained on the same tag types we used. Although BERT had the highest performance due to its contextual understanding, it was still not fine-tuned on our training data, thus it also missed some tags. For future work we could expand the dataset and add cross validation in order to make the evaluation more reliable. We could also introduce tag-specific augmentation and more fine tuning

References

1. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1810.04805
2. Karimian, M., & Beigy, H. (2023). Concept drift handling: A domain adaptation perspective. Expert Systems With Applications, 224, 119946. https://doi.org/10.1016/j.eswa.2023.119946
3. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A lite BERT for self-supervised learning of language representations. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1909.11942
4. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1910.01108
5. Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. (2020). MPNET: Masked and Permuted pre-training for language understanding. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2004.09297
6. Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, 2019.
7. F. Barbieri et al., "Unified Benchmark and Comparative Evaluation for Tweet Classification," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7596–7610, 2020.
8. D. Harbecke et al., "Why only Micro-F1? Class Weighting of Measures for Relation Classification," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3233–3239, 2019.
9. Khan, T. A., Sadiq, R. ., Shahid, Z. ., Alam, M. M., & Mohd Su'ud, M. B. . (2024). Sentiment Analysis using Support Vector Machine and Random Forest. *Journal of Informatics and Web Engineering*, *3*(1), 67–75. https://doi.org/10.33093/jiwe.2024.3.1.5
10. Rahat AM, Kahir A, Masum AKM. Comparison of Naive Bayes and SVM Algorithm based on sentiment analysis using review dataset. 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), 2019, p. 266–70. https://doi.org/10.1109/SMART46866.2019.9117512.
11. Jannah, Nurul & Kusnawi, Kusnawi. (2024). Comparison of Naïve Bayes and SVM in Sentiment Analysis of Product Reviews on Marketplaces. Sinkron. 8. 727-733. 10.33395/sinkron.v8i2.13559.
12. Tineges, R., Triayudi, A., & Sholihati, I. D. (2020). Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM). Jurnal Media Informatika Budidarma, 4(3), 650. https://doi.org/10.30865/mib.v4i3.2181
13. Sida Wang and Christopher Manning. 2012. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.
14. Xing, Frank & Cambria, Erik & Welsch, Roy. (2018). Natural language based financial forecasting: a survey. Artificial Intelligence Review. 50. 10.1007/s10462-017-9588-9.
15. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A robustly optimized BERT pretraining approach* (arXiv:1907.11692). arXiv. https://arxiv.org/abs/1907.11692
16. Araci, Dogu. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. 10.48550/arXiv.1908.10063.
17. Hugging Face. (n.d.). CoNLL-2003 Named Entity Recognition Dataset. https://huggingface.co/datasets/conll2003
18. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media, Inc.
19. Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the Eighteenth International Conference on Machine Learning, 282–289.
20. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv. https://doi.org/10.48550/arXiv.1810.04805
21. Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.