NLP: NERC, SENTIMENT, TOPICS

NAMED ENTITY RECOGNITION & CLASSIFICATION: SVM

INTRODUCTION

Named entities refer to specific instances of people, organizations, locations, objects, events, dates, and other real-world entities that are mentioned within natural language text. Named Entity Recognition is the task of using algorithms to automatically identify these entities, which may consist of single words or multi-word expressions, within a piece of unstructured text. Following recognition, Named Entity Classification is performed to assign each identified entity to a predefined overarching category, such as Person, Organization, or Location [1]. While different annotation schemes exist for marking the named entities in a text [1], this project makes use of the BIO (Beginning, Inside, Outside) tagging format. In this scheme, each token is labeled to indicate whether it is at the beginning (B), inside (I), or outside (O) of the span of a named entity, with the tag also specifying the category of the entity [1]. We train a supervised machine learning model to label named entities using this scheme.

DATA ACQUISITION & PREPROCESSING

The training dataset utilized for the Named Entity Recognition and Classification (NERC) task in this project is the OntoNotes 5.0 dataset, which is available on the Hugging Face website [2]. Originally introduced in a 2006 research article [3] and officially released in 2007, the OntoNotes corpus had several releases before reaching its final version in 2013 with OntoNotes 5.0 [4]. It consists of multiple genres of text, including telephone conversations, newswire, newsgroups, broadcast news, broadcast conversation, weblogs, and religious texts. The OntoNotes 5.0 dataset was selected for this task because it closely aligns with the domain and format of the test set provided by the course, while also offering a high number of annotated instances (see Table 1 for an indication). The only other dataset with a similar label scheme to the test set that we could find was CoNLL-2003, which had already been used in Lab 4, so we decided it would be more appropriate to use a new dataset for this project.

We downloaded the OntoNotes 5.0 dataset in the form of several JSON files from the Hugging Face website. Upon inspection, we discovered that the dataset was already very clean and required minimal preprocessing. In the case of NERC, it is important to retain stop words and punctuation, as they help the model properly identify and classify entities by providing cues about their context (e.g., preceding propositions) and format (e.g., abbreviations). Thus, no data were removed, only the instances were transformed from sentences annotated with a list of numerical named-entity labels to tokens annotated with categorical named-entity labels, based on the encoding provided in the Hugging Face website [2]. Each token was also assigned a corresponding sentence and token ID.

The dataset was already split into training, validation, and test sets at the time of acquisition. However, as the test set for the task was provided by the course, the original training and validation subsets were merged to form a single larger training set, while the original test set was repurposed for validation. Table 1 presents the distribution of entity categories across the resulting training and validation sets, as well as the provided test set. The training set contains over 1 million instances, and the validation set includes more than 150,000, while the test set comprises only 237 instances. This indicates that there is a sufficient amount of data for effective training and validation of the model before final evaluation. However, Table 1 also shows that the category distribution in all subsets is highly imbalanced, with the majority of tokens (over 80%) labeled as non-entities ("O"). Although the training and validation sets show similar distributions in terms of percentage, only 9 out of the 37 entity types present in those sets appear in the test set. This pronounced class imbalance warrants the use of specialized techniques designed to handle imbalanced data during model training and evaluation. These techniques will be discussed in more detail in the subsequent sections.

Table 1. Distribution of NERC Labels in the Training, Validation, and Test Sets.*

Named-Entity Label	Training Set Frequency	Training Set Percentage	Validation Set Frequency	Validation Set Percentage	Test Set Frequency	Test Set Percentage
O	1,066,810	86.30	131,814	86.31	186	78.48
I-ORG	20,582	1.66	2,406	1.58	2	0.84
B-GPE	17,673	1.43	2,240	1.47	0	0.00
B-PERSON	17,449	1.41	1,988	1.30	11	4.64
I-DATE	15,142	1.22	2,011	1.32	0	0.00
B-ORG	14,560	1.18	1,795	1.18	3	1.27
I-PERSON	12,542	1.01	1,412	0.92	8	3.38
B-DATE	12,429	1.01	1,602	1.05	0	0.00
B-CARDINAL	8,292	0.67	935	0.61	0	0.00
B-NORP	7,717	0.62	841	0.55	0	0.00
I-MONEY	5,499	0.44	683	0.45	0	0.00
I-GPE	4,234	0.34	628	0.41	0	0.00
I-PERCENT	2,756	0.22	523	0.34	0	0.00
I-WORK OF ART	2,734	0.22	337	0.22	10	4.22
B-MONEY	2,682	0.21	313	0.20	0	0.00
I-CARDINAL	2,579	0.21	331	0.22	0	0.00
B-PERCENT	1,940	0.16	349	0.23	0	0.00
I-EVENT	1,877	0.15	130	0.09	0	0.00
B-ORDINAL	1,872	0.15	195	0.13	0	0.00
I-TIME	1,767	0.14	255	0.17	0	0.00
B-LOC	1,718	0.14	179	0.12	7	2.95
I-FAC	1,670	0.14	213	0.14	0	0.00
I-LOC	1,583	0.13	180	0.12	1	0.42
B-TIME	1,447	0.12	212	0.14	0	0.00
I-QUANTITY	1,444	0.12	206	0.13	0	0.00
B-WORK OF ART	1,116	0.09	166	0.11	9	3.80
B-FAC	975	0.08	135	0.09	0	0.00
B-EVENT	891	0.07	63	0.04	0	0.00
I-LAW	869	0.07	105	0.07	0	0.00
B-QUANTITY	757	0.06	105	0.07	0	0.00
I-PRODUCT	705	0.06	69	0.05	0	0.00
B-PRODUCT	678	0.05	76	0.06	0	0.00
I-NORP	490	0.04	160	0.10	0	0.00
B-LANGUAGE	337	0.03	22	0.01	0	0.00
B-LAW	322	0.03	40	0.03	0	0.00
I-LANGUAGE	13	0.01	0	0.00	0	0.00
I-ORDINAL	9	0.01	4	0.01	0	0.00
TOTAL	1,236,160		152,723		237	

*The total percentage of some columns might not sum to 100 due to rounding.

METHODOLOGY

Multiple new features were derived from the original tokens to help the model learn how to perform NERC. The feature engineering process was inspired by the Kaggle dataset on NERC presented in Lecture 6 and provided in the assignment for Lab 4 [5]. These features included adding the part-of-speech (POS) tag, shape, and lemma of each token, as well as all the information about the two preceding and two following tokens to each instance. To extract POS tags and lemmatize the tokens, we utilized spaCy's en_core_web_sm NLP pipeline. For determining the shape of each token, we implemented a custom function that distinguishes between various token forms: lowercase, uppercase, capitalized, camel-case, mixed-case, abbreviations, punctuation, numbers, and honorifics ending in a dot (e.g., "Mr."). This categorization was based on the "shape" column in the Kaggle NERC dataset from Lab 4 [5].

Following the feature extraction, we selected the Support Vector Machine (SVM) algorithm for classification. This choice was guided by findings from Rosadi et al. [6], who reported that SVM (LinearSVC from Scikit-learn) outperformed Naive Bayes, Logistic Regression, and Conditional Random Fields on the CoNLL-2003 dataset—another news-domain corpus similar to OntoNotes 5.0—achieving an F1-score of 89.91% on the NERC task. Since SVMs require numerical input, we transformed all categorical features in our data into numerical vectors using one-hot encoding.

To address the high class imbalance in our training set where over 80% of tokens are labeled as non-entities ("O"), we applied class weighting. Class weights were computed automatically based on the distribution of labels in the training data, assigning higher importance to the minority classes prior to model training. We set the maximum number of iterations to 2,000, consistent with the settings in Rosadi et al.'s article [6]. To ensure the reproducibility of our results, we also fixed the random state parameter of the model to 0. All other parameters, including the type of loss function and regularization, were left at Scikit-learn's default values, as there was no mention of their modification in the referenced paper [6].

Initially, we replicated the same feature set as in the study by Rosadi et al. [6], which included the current token, the previous and following tokens, their respective POS tags, the shape of the current token, and the sentence ID as features for each instance. However, this setup did not result in very satisfactory performance on our validation set (accuracy of 87.00%). Through several rounds of manual testing, we determined that expanding the feature set led to improvements, particularly in underrepresented entity categories. While the gains in overall accuracy were small (typically about 1%) with the addition of each batch of new features (e.g. lemmas of all tokens), they were meaningful given that over 80% of the tokens in the validation set were non-entities. Our final feature set results includes the following for each instance: the token, its lemma, POS tag, and shape, along with the corresponding token, lemma, POS tag, and shape of the two preceding and two following tokens, as well as the sentence ID and token ID of the current token. This comprehensive representation captures the token's context, capitalization, and linguistic attributes, while also providing an indication about entity spans. It yielded the best results on the validation set (accuracy of 90.68%) and was thus used for final model training before the test set evaluation.

RESULTS & DISCUSSION

Table 2 presents the precision, recall, and F1-score for the different named-entity categories, along with the overall accuracy, macro averages, and weighted averages obtained by the SVM on the test set. The classifier achieved an overall accuracy of 84.81%, with weighted averages of 92.72% precision, 84.81% recall, and 87.65% F1-score. While these results may appear strong, due to the high class imbalance in the test set, they are heavily influenced by the prevalence of the majority class of non-entities ("O"). As such, these metrics primarily reflect the model's ability to correctly identify non-entity tokens rather than its performance on actual named-entity categories. A more balanced assessment is offered by the macro averages, which treat each class equally, regardless of frequency. The macro precision, recall, and F1-score are 29.79%, 28.07%, and 26.21%, respectively, indicating that the SVM struggles to identify many of the actual named-entity categories in the test set.

Table 2 also reveals that the classifier— perhaps because it was trained on a broader label set of 37 categories—predicts seven entity types in the test set that are not part of its gold labels. These include: B-FAC/I-FAC (facilities, buildings, roads), B-GPE/I-GPE (geopolitical entities such as countries, cities, and states), B-NORP (nationalities or political or religious groups), B-CARDINAL (cardinal numbers), and B-ORDINAL (ordinal numbers). This discrepancy is likely due to the richer and more fine-grained label set used for the training data compared to the test data.

Table 2. SVM Classification Performance on the Test Set.

Named-Entity Label	Precision (%)	Recall (%)	F1-Score (%)	Number of Instances
B-LOC	0.00	0.00	0.00	7
I-LOC	0.00	0.00	0.00	1
B-ORG	21.43	100.00	35.29	3
I-ORG	0.00	0.00	0.00	2
B-PERSON	100.00	63.64	77.78	11
I-PERSON	70.00	87.50	77.78	8
B-WORK OF ART	100.00	44.44	61.54	9
I-WORK OF ART	85.71	60.00	70.59	10
B-FAC	0.00	0.00	0.00	0
I-FAC	0.00	0.00	0.00	0
B-GPE	0.00	0.00	0.00	0
I-GPE	0.00	0.00	0.00	0
B-NORP	0.00	0.00	0.00	0
B-CARDINAL	0.00	0.00	0.00	0
B-ORDINAL	0.00	0.00	0.00	0
O	99.43	93.55	96.40	186
Macro Averages	29.79	28.07	26.21	237
Weighted Averages	92.72	84.81	87.65	237
Accuracy (%)			84.81	237

For categories that were actually present in the test set, we see that the model particularly struggles with distinguishing location (LOC) entities. It achieved 0% precision, recall, and F1-score for both the B-LOC and I-LOC categories. For instance, "Barcelona" was incorrectly classified as B-ORG (beginning of an organization), and "Bristol" as B-FAC, though both are labeled as B-LOC in the gold annotations. However, most of the misclassifications seem to stem from the mismatch in label granularity between the training and test data. For example, "Dublin" and "Berlin" were labeled by the model as B-GPE, although their label in the test set is B-LOC. Similarly, "Wembley" and "Stadium" were predicted as B-FAC and I-FAC, respectively, while their true label was B-LOC. In all these cases, the predicted labels are semantically plausible, just more specific than the test set categories, which suggests that the classifier is capable of making reasonable distinctions for places, even if they do not align with the test set annotations.

The SVM also had difficulty with organization (ORG) entities. For B-ORG, the model reached 100.00% recall but only 21.34% precision, resulting in an F1-score of 35.29%. This indicates that while the model correctly identified all true B-ORG instances, it also over-predicted the label, assigning it to many tokens incorrectly, which resulted in a high number of false positives. Examples of misclassified tokens include "GOAT," which was originally labeled as O, "Oppenheimer," originally B-WORK OF ART, and "Banksy," originally B-PERSON. Based also on the scores and misclassifications for I-ORG—where the model achieved 0% precision, recall, and F1-score—we can conclude that the classifier frequently confuses PERSON and WORK-OF-ART entities with organizations. For example, "Queen" and "Elizabeth," which have true labels B-PERSON and I-PERSON, were instead predicted as B-ORG and I-ORG. Similarly, "Harry" and "Potter," with true labels B-WORK OF ART and I-WORK OF ART, were both classified as I-ORG. These errors suggest that the model, perhaps due to its reliance on hand-engineered features and a linear kernel, lacks the contextual understanding required to distinguish between categories such as PERSON, ORG, and WORK-OF-ART. These categories often share surface characteristics, including capitalization, no difference between token and lemma, and overlapping vocabulary (e.g., names of people frequently appear in organization names and titles of works of art). The classifier also labeled "United" from "Manchester United" as I-GPE and "Miami" from "Inter Miami" as I-PERSON, while their gold annotation was I-ORG. These examples further highlight the model's limitations in capturing context, despite the inclusion of features for the two preceding and two following tokens.

Despite these challenges, the SVM showed relatively strong performance on PERSON and WORK-OF-ART categories. It achieved 100.00% precision for B-PERSON and B-WORK-OF-ART, indicating that every prediction made for the beginning of these entity types was correct. However, the precision for the middle tokens of these entities was slightly lower, with 70.00% for I-PERSON and 85.71% for I-WORK OF ART, indicating some difficulty in consistently identifying entity spans. For instance, in the phrase "The Catcher in the Rye," the tokens "in" and "Rye" were incorrectly predicted as O and I-GPE, respectively, while their gold label was I-WORK OF ART. In this case, "Rye" can be considered an ambiguous entity, as it is nested within a larger named entity and has multiple possible interpretations—it can refer to either a type of grain, or one of several towns with the same name.

While the model achieved high precision overall, its recall was comparatively lower, with 63.64% for B-PERSON, 87.50% for I-PERSON, 44.44% for B-WORK OF ART, and 60.00% for I-WORK OF ART. These scores suggest that though the model can identify some entity instances from these categories with high confidence, it also misses a significant number of others. Some of the misclassified entities were mentioned previously. However, additional misclassifications include "1984" and "Barbie," which were originally labeled as B-WORK OF ART but predicted as B-CARDINAL and B-ORG, respectively. In the case of "1984," the misclassification is understandable, as recognizing it as a literary title requires background knowledge, which the classifier does not inherently possess.

In summary, although the SVM achieved only moderate performance in classifying named entities, the evaluation on the test set offers valuable insights into its behavior. Its low macro average scores are caused in part by the mismatch in label granularity between the training and test data, particularly for location entities, and in part by its limited capacity for capturing contextual information. The model relies heavily on surface-level features, such as shape and POS tags, which are insufficient to distinguish between entity types that frequently overlap in form and vocabulary, as is often the case with persons, organizations, and works of art. This limitation becomes particularly evident in the presence of ambiguous entities and instances where correct classification would require background or world knowledge, which the model lacks.

CONCLUSION & FUTURE WORK

For the NERC task of this project, we developed an SVM model that demonstrated solid performance considering the time constraints of the project. While the model encountered notable difficulties with location entities and achieved lower scores for organization entities—likely due to its limited contextual awareness—it attained near-perfect precision for person and work-of-art entities. The qualitative analysis also revealed that some misclassifications were either justifiable, or difficult to avoid without background knowledge, which highlights some of the innate challenges of NERC.

To address the limitations of the SVM in future work, we propose several possible enhancements. First, replacing the one-hot encoding of the tokens with word embeddings could enable the model to capture semantic similarities and improve generalization. This may, in turn, enhance the model's recall, which has been consistently worse compared to its precision. Conducting a full feature ablation study might also be beneficial, as it would help quantify the impact of each feature on performance and result in more informed feature selection, which could further improve the scores of the SVM. However, both implementing a model that combines word embeddings with other features and performing a detailed feature ablation study were beyond the scope of this project due to the time constraints. Regarding the model's lack of background knowledge, a hybrid approach incorporating external resources like a gazetteer to verify predicted entities might potentially reduce misclassifications in cases where real-world knowledge is essential [1], such as with works of art. In terms of improving contextual awareness, the current model already includes features from the two preceding and following tokens, but this approach offers only a shallow understanding of the context compared to models designed to capture long-range dependencies. Thus, a promising direction for future work would be to replace the SVM with a transformer-based model, such as RoBERTa, which uses self-attention to modify the token representations based on the context provided by surrounding words.

Finally, to mitigate the effects of class imbalance in the training data, application of data augmentation methods, such as generating variations of minority-class instances through synonym replacement, might have a positive effect [7]. We believe the proposed methods could help improve the performance on the NERC task, given time to implement them.

SENTIMENT ANALYSIS: VADER VS BERT

INTRODUCTION

Sentiment analysis is a natural language processing (NLP) task that involves identifying and categorizing emotions or opinions expressed in text, typically as positive, negative, or neutral [8]. In this section, we evaluate and compare the performance of two sentiment analysis models: VADER, a rule-based approach specifically designed for social media text, and BERT, a transformer-based deep learning model known for its strong contextual understanding, to assess whether a sophisticated model like BERT can outperform a standard lexicon-based approach, as suggested by Saha et al. [9]. This comparison includes both qualitative and quantitative analyses, which demonstrate not only how the models perform numerically but also how they handle real examples of complex sentiment.

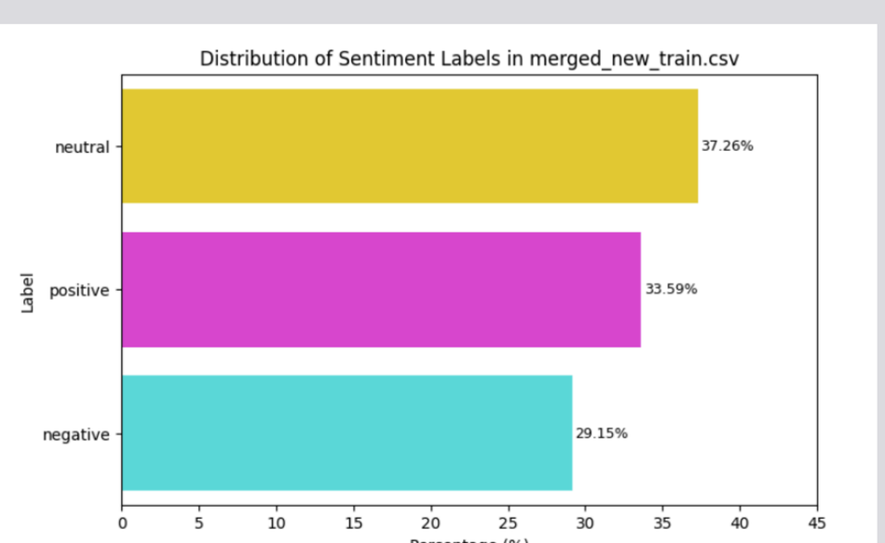

Figure 1. Distribution of Sentiment Labels in the Training Set

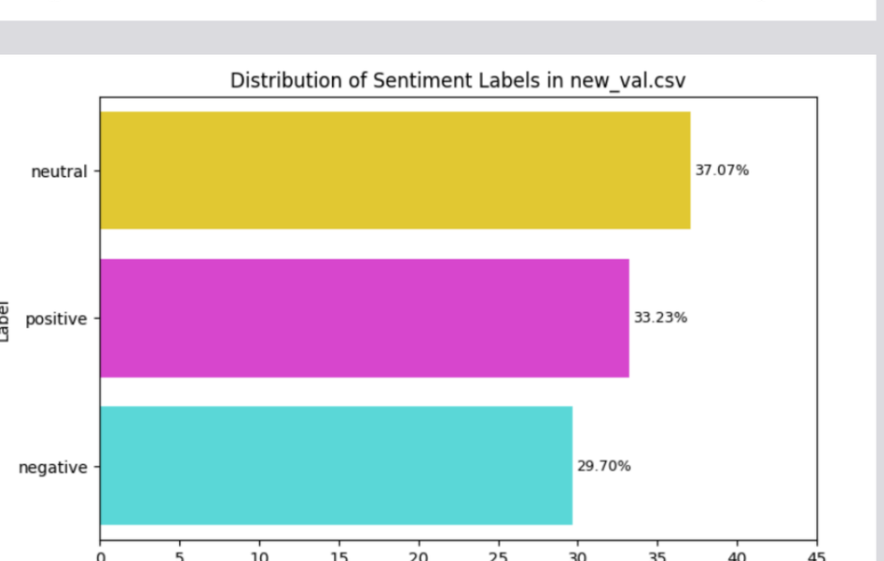

Figure 2. Distribution of Sentiment Labels in the Validation Set

DATA ACQUISITION & PREPROCESSING

The training dataset used for the Sentiment Analysis task is a multiclass sentiment analysis dataset found on the Hugging Face website [10]. It consists of short social media posts, similarly to the test data provided by the course. The dataset was originally split into training, validation and test CSV files. However, since the test data was already given by the course, we merged the training and validation sets to create a larger training set, and we used the test set as validation data.

After merging, the class distribution across the three sentiment categories (neutral, positive, and negative sentiment) remained relatively balanced in both the new training set (Figure 1) and validation set (Figure 2). While the neutral class has a slightly higher number of instances and the negative class the fewest, the differences are small, around 8% in both sets, so no additional class balancing techniques were applied.

In terms of sample counts, the training set contains 36,437 examples: 13,577 neutral, 12,238 positive, and 10,622 negative. The validation set includes 5,206 entries: 1,930 neutral, 1,730 positive, and 1,546 negative. The provided test set is perfectly balanced, consisting of 18 entries total, with 6 examples per class.

To prepare the text data for model training, a basic cleaning function was applied. This included converting all text to lowercase and removing URLs, hashtags, mentions, and non-alphanumeric characters, while preserving punctuation that may carry sentiment. These steps follow common preprocessing practices in sentiment analysis, aiming to reduce noise and standardize input without discarding emotionally relevant textual features [11]. After the preprocessing, the class distributions remain unchanged.

METHODOLOGY

We conduct a comparative analysis between VADER and BERT (base-uncased) and their performance on the sentiment analysis task. Our goal is to evaluate whether the performance of a fine-tuned transformer-based model significantly exceeds that of a traditional rule-based approach. To formalize this, we define our hypotheses as follows:

- **H_0: There is no significant difference in sentiment classification performance between VADER and BERT on the given dataset.**
- **H_1: BERT demonstrates a statistically significant improvement in performance over VADER for the sentiment classification task.**

Based on the findings of Saha et al. [9], we expect to reject H0 with significance level $\alpha = 0.05$ and verify that BERT outperforms VADER in this sentiment classification task.

VADER

VADER (*Valence Aware Dictionary and sEntiment Reasoner*) is a lexicon and rule-based sentiment analysis tool optimized for social media contexts and short-form text [12]. We chose VADER as a baseline due to its lightweight nature, ease of use, and respectable performance in comparison to other lexicon models, such as AFINN—all without the need for any training or large computational resources [13].

Since VADER is a lexicon-based model, it does not need any specific feature engineering, extensive text preprocessing, or training. It was tested directly on the provided test set without prior fine-tuning. An optional preprocessing step using spaCy's *en_core_web_sm* model was implemented, which performs tokenization and lemmatization, if enabled. When preprocessing is disabled, VADER receives the raw input sentence. Upon testing, tokenizing the input led to worse performance compared to using raw text, with lemmatization negatively impacting performance even further. Based on these findings, raw input was used in the final evaluation to maximize model performance. Similar to the assignment in Lab 3, sentiment labels were derived from the VADER's compound score using standard thresholds: scores > 0 were classified as positive, scores < 0 as negative, and scores equal to 0 as neutral. We evaluated the model using classification metrics such as precision, recall, and F1-score, and included a detailed misclassification breakdown to support error analysis.

BERT

BERT (*Bidirectional Encoder Representations from Transformers*) is a deep learning-based language representation model that has set new benchmarks across a wide range of natural language processing (NLP) tasks, including sentiment analysis [14]. In contrast to lexicon-based models such as VADER, BERT captures contextual relationships between words by leveraging large-scale self-supervised pre-training on massive text corpora. BERT effectively captures syntactic, semantic, and world knowledge, making it highly adaptable across tasks with minimal task-specific modifications [15]. BERT was selected as the second method for sentiment analysis due to its strong performance on classification tasks and its ability to model complex, context-dependent sentiment. These are capabilities that exceed those of traditional lexicon-based approaches, making it a valuable point of comparison with VADER [16].

The following section outlines the feature engineering, preprocessing, and model configuration steps applied to fine-tune BERT for sentiment classification. The pre-trained BERT model was fine-tuned to predict sentiment labels, allowing the model to adapt its internal representations to the specific task of sentiment analysis. Sentiment labels were mapped to integers, with negative as 0, neutral as 1, and positive as 2, to comply with BERT's input format. Furthermore, unlike lexicon-based models, BERT requires inputs to be tokenized using a subword tokenizer; therefore, input sentences were preprocessed using the model's built-in WordPiece tokenizer, which splits text into subword units and encodes them as token IDs. Each input sequence was standardized to a fixed maximum length of 512 tokens, the maximum supported by BERT, to accommodate longer text without loss of information. The key training parameters and their motivations are summarized in Table 3. The initial parameter values were chosen based on course-provided code and further adjusted through experimentation to suit the specific characteristics of the dataset and task. As with VADER, model performance was evaluated using precision, recall, and F1-score. Predictions were compared to gold labels from the test set, and misclassifications were analyzed to identify common failure cases.

Table 3. BERT Model Configuration and Justification.

Parameter	Value	Impact / Motivation
num_train_epochs	10	More epochs allow deeper training, selected for sufficient learning without overfitting
train_batch_size	32	Offers a good balance between training stability and GPU memory
learning_rate	4×10^{-6}	Lower learning rate for gradual and stable updates to prevent catastrophic forgetting
max_seq_length	512	Maximum length BERT can handle to fully capture longer input text
fp16	True	Speeds up training and reduces memory usage
gradient_accumulation_steps	2	Simulates a larger batch size
use_early_stopping	True	Avoid unnecessary training epochs once validation performance stops improving
early_stopping_patience	5	Allows minor metric fluctuations without stopping training prematurely

RESULTS & DISCUSSION

VADER

The VADER sentiment analysis model shows relatively poor performance when evaluated on the test set. As shown in Table 4, the overall accuracy is only 33.33%, indicating that VADER correctly classified just a third of the samples. The model struggles the most with negative sentences, achieving 0% precision, recall, and F1-score, meaning it failed to identify any negative instances correctly. While the model performs slightly better on neutral and positive examples, demonstrating moderate recall (50%) across both classes, indicating it detects half of the actual instances, its lower precision (33.33% and 42.86% respectively) reveals a high rate of false positives, especially in the neutral class. The F1-scores for the neutral (40.00%) and positive (46.15%) classes indicate the model is slightly better at balancing precision and recall for the positive class compared to the neutral. The low macro and weighted averages further confirm the model's limited effectiveness across all classes.

Table 4. VADER Classification Performance on the Test Set.

Class	Precision (%)	Recall (%)	F1-Score (%)	Number of Instances
Negative (0)	0.00	0.00	0.00	6
Neutral (1)	33.33	50.00	40.00	6
Positive (2)	42.86	50.00	46.15	6
Macro Average	25.40	33.33	28.72	18
Weighted Average	25.40	33.33	28.72	18
Accuracy (%)			33.33	18

In total, VADER misclassified 3 out of 6 positive examples, mistaking them for either neutral or negative. It was observed that VADER underperforms when presented with informal or implicit expressions of positivity and excitement, especially when such expressions are common in spoken language but not in formal sentiment lexicons. For example, in "It had me hooked from the first chapter.", the non-formal use of "hooked" as a positive expression is ignored, leading to a neutral prediction. The negative class was entirely misclassified. The most frequent source of error for negative sentences is VADER's inability to handle sarcasm, rhetorical language, or implicit criticism, highlighting VADER's weakness with more subtle and indirect forms of negativity. For instance, "It's really incredibly impressive to mess up such a tested blockbuster formula." is clearly sarcastic, which VADER fails to capture, likely due to the words "incredibly" and "impressive" being typically positive. Regarding the neutral class, again 3 instances were misclassified. It shows a tendency where VADER may overemphasize isolated positive or negative words without fully considering the sentence's overall sentiment. For example, "It's split into two timelines, which keeps it interesting but also a bit confusing at times." was misclassified as negative due to the presence of "confusing", which has a negative connotation in the lexicon.

BERT

The BERT model achieves an overall accuracy of 88.89% on the test set, as shown in Table 5. Precision is highest for the negative class at 100%, but the negative class also shows the lowest recall at 66.67%, meaning the model correctly identifies all predicted negatives but misses some actual negative sentences. In contrast, both the neutral and positive classes have perfect recall of 100%, with a slightly lower precision of 85.71%. These results suggest that the model tends to misclassify other classes as neutral or positive. The F1-scores for neutral and positive are equal at 92.31%, while the negative class has a lower F1-score of 80.00%. Macro and weighted averages for precision, recall, and F1-score are consistent at 90.48%, 88.89%, and 88.21%, respectively, indicating balanced performance across classes. Overall, the model performs strongly, but its lower recall on the negative class highlights an area for improvement in detecting negative sentiments.

Table 5. BERT Classification Performance on the Test Set.

Class	Precision (%)	Recall (%)	F1-Score (%)	Number of Instances
Negative (0)	100.00	66.67	80.00	6
Neutral (1)	85.71	100.00	92.31	6
Positive (2)	85.71	100.00	92.31	6
Macro Average	90.48	88.89	88.21	18
Weighted Average	90.48	88.89	88.21	18
Accuracy (%)			88.89	18

Table 6 demonstrates the two misclassified instances by the BERT model, which offers us insight into the model's limitations. First of all, the sentence "It's really incredibly impressive to mess up such a tested blockbuster formula" is labeled as negative, but the model predicts it as positive. This is a brilliant example of a model's shortcomings in detecting sarcasm or subtle negative sentiment masked by seemingly positive language like "incredibly impressive". Second of all, the sentence "The only way it's helped me is by keeping my table from being wobbly" is also labeled negative but predicted as neutral. This is another challenging instance for a model since the negative sentiment is implied rather than explicitly stated. These errors show that while BERT handles most cases effectively, it can misinterpret nuanced language, such as sarcasm and indirect negativity. This type of language understanding requires contextual or pragmatic comprehension that goes beyond surface-level cues, underscoring a common gap in many NLP tasks.

Table 6. BERT Misclassified Instances.

Text	Label	Predicted
It's really incredibly impressive to mess up such a tested blockbuster formula.	Negative	Positive
The only way it's helped me is by keeping my table from being wobbly.	Negative	Neutral

COMPARISON

1. **Overall Comparison**

It can be clearly seen that BERT outperforms VADER across all metrics. Overall accuracy increases from 33.33% to 88.89%, indicating a much higher rate of correct predictions. VADER's precision of 25.40% compared to BERT's 90.48% shows that BERT produces far fewer false positives. Recall improves from 33.33% to 88.89%, meaning the model is better at capturing relevant instances. Similarly, the F1-score of VADER is 28.72% compared to BERT's 88.21%, demonstrating high and balanced overall performance. These results present BERT's advantage in understanding more context and capturing nuanced language, where VADER's rule-based approach proves insufficient.

2. **Per Topic Comparison**

For the negative class, VADER fails to identify any instances correctly, with precision, recall, and F1-score all at 0%. In contrast, BERT achieves 100% precision, 66.67% recall, and an F1-score of 80%, indicating a significantly better ability to recognize negative sentiment. Still, both approaches seem to struggle the most with negative sentiment. In the neutral class, VADER performs with 33.33% precision, 50% recall, and a 40% F1-score. BERT outperforms it with 85.71% precision, perfect recall at 100%, and an F1-score of 92.31%, reflecting much stronger classification in this category. In like manner, for the positive class, VADER reaches 42.86% precision, 50% recall, and a 46.15% F1-score, while BERT again scores 85.71% precision, 100% recall, and 92.31% F1-score. From this comparison, we can conclude that BERT has more accurate and consistent performance across all sentiment classes, especially in handling neutral and positive instances.

3. **Statistical Analysis**

To support the observed performance differences, a Z-Test for proportions was conducted to assess the statistical significance of the results. The null hypothesis (H_0) stated that there would be no significant difference in sentiment classification performance between VADER and BERT, while the alternative hypothesis (H_1) stated that BERT would demonstrate a statistically significant improvement in performance over VADER. The test results show that for all four key metrics—precision ($p = 0.000028$), recall ($p = 0.000314$), F1-score ($p = 0.000398$), and accuracy ($p = 0.000100$)—the p-values are all below the significance threshold ($\alpha = 0.05$). Therefore, **we reject the null hypothesis** across all metrics, confirming that BERT significantly outperforms VADER in sentiment classification on this dataset.

CONCLUSION & FUTURE WORK

We have compared the performance of VADER and BERT on a sentiment classification task and found that BERT consistently outperforms VADER across all sentiment classes, achieving significantly higher accuracy, precision, recall, and F1-score. BERT's ability to model more nuanced information allowed it to better distinguish between sentiments, while VADER struggled, especially with implicit or sarcastic expressions. A limitation of this work is that, although the training dataset is relatively balanced and sizable, it may still not cover the full variety of sentiment expressions found in real-world scenarios. Future work could involve fine-tuning BERT on larger, more diverse datasets to improve generalization. Performance could be further enhanced by implementing more advanced models such as RoBERTa. Exploring hybrid models that combine rule-based and deep learning approaches is another promising direction for more effectively handling implicit and sarcastic sentiment.

TOPIC CLASSIFICATION: MULTINOMINAL NB

INTRODUCTION

In the field of natural language processing, topics provide abstract categories of the main idea in a given text. Topic classification is used to computationally assign predefined labels to documents or texts, transforming unstructured content into a structured representation by categorizing it into relevant topics. One of the many approaches for topic classification involves supervised machine learning, where models learn from labeled data to categorize documents into predefined classes, emphasizing the role of lexical and semantic pattern recognition [17]. A proper topic classification can enhance applications like information retrieval and knowledge discovery, which makes it a critical component in NLP tasks [18]. In this project, a supervised ML model will be trained to classify given texts into 3 distinct thematic categories: sports, books, and movies.

DATA ACQUISITION & PREPROCESSING

The training dataset used for the topic classification in this project was obtained from Kaggle [19]. It consists of reviews, which are organized into 3 distinct textual categories— sports, book, and movie—mirroring the classes in the provided test set [19]. The training dataset has a bit over 120,000 instances in total. Due to the large volume of instances, it was later split into a training and validation set. An overview of the distribution of the instances over the topics before partitioning can be seen in Figure 3. Further details about the dataset split and usage will be discussed below.

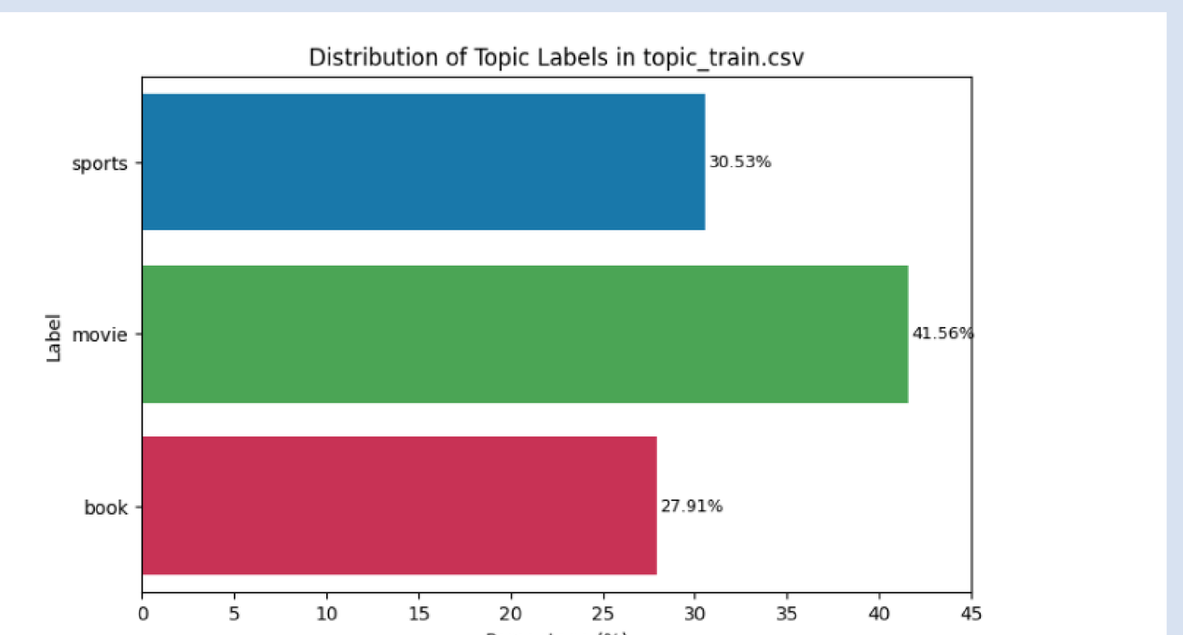

Figure 3. Distribution of Topic Labels in the Training Dataset before Cleaning

Before training our model, the textual data had to go through some preprocessing stages designed to enhance the classification model's performance. Firstly, stop-word removal was done using the NLTK library to eliminate frequently occurring English words "a," "the," and "of" since these words carry minimal semantic weight and often lead to unwanted noise during feature extraction [20]. Furthermore, stemming was also done with the Porter Stemmer from NLTK, removing several morphological variants of words to transform words into their singular root form, leading to lower lexical redundancy [20]. Moreover, non-English reviews, special characters, extra spaces, and numbers were also eliminated to ensure consistency [21]. Named entities carry a significant role in understanding and categorizing data, so it was ensured that they were preserved by using spaCy and kept in n-gram form in each review instance [22].

Additionally, reviews with less than 3 words, emojis and common emoticons, and symbols were discarded to reduce noise and maintain a good data quality. To take it a step further, duplicate reviews were dropped to increase the model reliability and reduce bias. In topic classification, duplicates can skew the distribution of topics, leading to overfitting, redundant data processing, and inaccuracies [23]. Lastly, the dataset was undersampled—meaning that instances from the majority class were deliberately removed in order to create a more balanced class distribution. This balance is crucial for training machine learning models, as imbalanced datasets can lead to biased classifiers that favor the majority class [24].

After all preprocessing steps, the dataset had a balanced structure with an equal number of instances for each label. The final distribution used for training and validation is presented in Table 7, providing a reliable foundation for the model training.

Table 7. Distribution of Topic Labels in the Training and Validation Sets

Dataset	Labels	Count	Total
Training Set	Book	23,895	71,685
	Sports	23,895	
	Movie	23,895	
Validation Set	Book	5,974	17,922
	Sports	5,974	
	Movie	5,974	

METHODOLOGY

Following the data cleaning and preprocessing stages, the next step was feature engineering and model configuration to build a robust topic classification model. For feature extraction, a TF-IDF vectorization method was used to convert the textual data (the reviews) into numerical features. The model uses TF-IDF for topic classification because it effectively identifies the importance of terms within a document relative to the entire text corpus. TF-IDF helps in distinguishing important and distinctive words by assigning higher weights to terms that are frequent in a specific document but rare across other documents. This ensures that the model focuses on words that are meaningful for topic classification [20]. In our case, the TF-IDF vectorizer was configured with a maximum feature limit of 5000 for transforming the reviews into a numerical feature matrix. The resulting sparse matrix was then converted into a Pandas data frame and concatenated with the corresponding labels to make the final dataset for the model training.

The balanced dataset was partitioned into training (80%) and validation (20%) subsets using stratified sampling, which led to the label distribution shown in Table 7 above. Stratified splits expose the model to a representative sample of all classes during training, reducing overfitting to specific classes and improving its ability to generalize to unseen data [25].

The Multinomial Naive Bayes (MNB) classifier was selected as the model for topic classification because it achieved the highest overall accuracy of 91.8% in a study by Rahman and Akter—outperforming Decision Tree (79.6%) and K-NN (82.6%) at their best settings [20]. Additionally, MNB offers a good balance between training and testing time, and its straightforward approach using prior and posterior probabilities simplifies the classification process [20]. The TF-IDF features, generated during feature engineering, served as the input for the model.

To optimize the performance of our Naive Bayes classifier on the task, we conducted hyperparameter tuning using Optuna. The objective was to finetune the *alpha* and *fit_prior* parameters [26] of the MultinomialNB model to maximize classification accuracy. We defined a search space where *alpha* was sampled on a logarithmic scale between 0.001 and 10.0, and *fit_prior* was a binary choice (True or False). To address potential overfitting, the optimization process involved computing accuracy on the validation set after training with a certain set of values on the training set. Optuna was run for 100 trials and the optimal values found were *alpha* of 7.430739850967651 and *fit_prior* set to True.

Trained with these parameters, the model achieved an overall accuracy of 97.28% on the validation set. The full classification report is shown in Table 8. The precision metric indicates that the model rarely assigns an incorrect topic to a review, while the high recall values demonstrate its ability to capture most instances of each topic. The F1-scores, balancing these two measures, confirm that the classifier performs consistently across all topics. This best-performing model was also subsequently evaluated on the test set.

For evaluation, the test set was loaded with only the required columns ("sentence" and "topic") because the other columns were not related to the topic classification task. The instances were preprocessed using the same function as the training data. Then the previously saved TF-IDF vectorizer and the optimized Multinomial Naive Bayes model were called using Joblib, and the preprocessed test reviews were transformed into the TF-IDF feature space. These features were then converted into a Pandas data frame to align with the model's input format. The true topic labels were extracted from the test set before the optimized model predicted labels for the reviews. Then a classification report was produced to showcase the model's performance. Additionally, the misclassified reviews with their proper labels were identified for further analysis. The results of these evaluations will be discussed in the next section.

Table 8. Multinomial Naive Bayes Classification Performance on the Validation Set

Class	Precision (%)	Recall (%)	F1-Score (%)	Number of Instances
Book	98.90	94.59	97.60	5,974
Sports	93.47	99.65	96.46	5,974
Movie	99.85	97.59	98.70	5,974
Macro Average	97.40	97.28	97.29	17,922
Weighted Average	97.40	97.28	97.29	17,922
Accuracy (%)			97.28	17,922

RESULTS & DISCUSSION

The performance of our TF-IDF Naive Bayes classifier for topic classification is evaluated using both quantitative metrics and qualitative analysis of the reviews. The quantitative evaluation on the test set is presented in Table 9, while Table 10 showcases the full (unpreprocessed) text of the misclassified reviews. These analyses provide a comprehensive view of the classifier's strengths and limitations.

The Multinomial Naive Bayes classifier achieved an overall accuracy of 88.89%. Notably, the "book" category achieved a perfect precision and recall (both 100%), resulting in a perfect F1-score of 100%. This suggests that all book-related instances were correctly identified. On the other hand, the "sports" category showed a high precision (100%) but the lowest recall (66.67%) and F1-score (80.00%), suggesting that the model is highly confident with no false positives but misses several of the true sports-related examples. In contrast, the "movie" category demonstrated high recall (100%) but low precision (75.00%), resulting in a balanced F1-score of 85.71%. These scores indicate that the model identified all movie instances correctly, but also overpredicted the category, leading to some false positive. The macro and weighted average F1-scores of 88.57% indicate consistent and high performance across classes. The variations across categories are likely due to topic-specific linguistic patterns and the inherent ambiguity of certain reviews.

Table 9. Multinomial Naive Bayes Classification Performance on the Test Set

Class	Precision (%)	Recall (%)	F1-Score (%)	Number of Instances
Book	100.00	100.00	100.00	6
Sports	100.00	66.67	80.00	6
Movie	75.00	100.00	85.71	6
Macro Average	91.67	88.89	88.57	18
Weighted Average	91.67	88.89	88.57	18
Accuracy (%)			88.89	18

In addition to the overall metrics, a detailed review of misclassified examples provides insights into the challenges faced by the model. The following table (Table 10) presents the instances where the predicted topic did not match the true label. The first example, which discusses the score of the game and the team's performance, was misclassified as "movie" rather than "sports." This error suggests that the model may sometimes mix sports-related terminology with features common in movie reviews, possibly due to overlapping vocabulary in the descriptions of dramatic events. This highlights the challenge of capturing paradigmatic context, where the meaning of words is dependent on the broader context of the conversation [27]. The second misclassification— a sentence implicitly referring to football—shows the model once again assigning the "movie" label to a review that should have been categorized as "sports," hinting at potential ambiguities in phrasing or contextual cues. The model might be missing subtle relationships between words, which can be addressed through the use of word embeddings to better capture semantic meaning [27]. The misclassifications underscore the difficulty of capturing subtle differences in brief or ambiguous reviews where the language used is generic. This is particularly relevant since even state-of-the-art classifiers struggle with short, context-poor comments [27].

Table 10. Multinomial Naive Bayes Misclassified Instances

Text	Topic	Predicted
It's still 0-0 so far, so way too early to tell - both teams trying their hardest, but maybe it won't be enough?	Sports	Movie
I don't get the appeal at all, it's just a couple guys kicking a ball around at the end of the day.	Sports	Movie

While the TF-IDF Naive Bayes classifier demonstrates strong performance on the validation set as discussed in the previous section, the performance on the test set is noticeably lower. This discrepancy is partly because of the small sample size of the test set and possibly the increased ambiguity in the test reviews. The overall accuracy of 88.89% on the test set, combined with the detailed misclassification analysis, indicates that the model is effective in general but still prone to errors in cases where the language is ambiguous or overlapping between topics. Furthermore, the differences in performance across the topics highlight areas for potential improvement.

In conclusion, the quantitative and qualitative evaluations together provide a balanced view of the model's performance. The insights from this analysis can inform future work, particularly in reducing the model's dependence on surface-level lexical features and improving its ability to capture contextual cues in topic classification problems.

CONCLUSION & FUTURE WORK

In conclusion, our study demonstrates that a TF-IDF Naive Bayes classifier is an highly effective approach for topic classification, achieving robust performance and high overall accuracy on the test set. The model's precision, recall, and F1-scores across the categories demonstrates that it is able to accurately identify the key features associated with sports, books, and movies. As a result, it successfully assigns the correct topics to almost all review instances, though there remains room for improvement. Notably, the drop in performance from the validation set to the test set highlights challenges related to knowledge transfer, the handling of ambiguous or context-poor reviews, and variations in linguistic patterns across topics.

Future work could focus on exploring ensemble methods that incorporate attention mechanisms and pre-trained word embeddings to better capture paradigmatic context and address the challenges of short, ambiguous reviews [27]. These efforts are aimed at developing a more robust and generalizable topic classification system for real-world applications.

Work Division of Group 8

Student	Coding	Analysis	Poster
Joana Petkova (2757809)	Found all three training datasets, made template notebook for the code, wrote the code for NERC, and reviewed the other code	Made the tables and wrote the text in the NERC part of the poster, and reviewed all other poster sections	Arranged the text and tables in the NERC section and helped design the color scheme
Evelina Lune (2757761)	Wrote the code for Sentiment data pre-processing, BERT implementation, and statistical analysis, and reviewed the other code	Contributed to all parts of sentiment analysis section, and reviewed all other poster sections	Made the overall poster design and contributed to the sentiment section
Lazar Gyoshev (2782742)	Implemented VADER, distribution analysis visualization functions, and Optuna hyperparameter tuning for topic classification, and reviewed the other code	Contributed to Data Acquisition & Preprocessing, Methodology, Results & Discussion, and Conclusion sections of sentiment analysis, and reviewed all other poster sections	Contributed to the sentiment section, and ensured the references, tables and figures numbers are consistent in the poster
Mahbod Tajdini (2756234)	Made the whole code for topic classification, except the Optuna function, tried LLDA for topic classification and chose between the models, reviewed the other code	Wrote the whole part about topic classification, except one paragraph related to Optuna, and reviewed all other poster sections	Made the topic classification section, checked for consistency in table design

References

[1] A. Goyal, V. Gupta, and M. Kumar, "Recent Named Entity Recognition and Classification techniques: A systematic review," Computer Science Review, vol. 29, pp. 21–43, Aug. 2018, doi: https://doi.org/10.1016/j.cosrev.2018.06.001.
[2] TNER, "Ontonotes 5," Huggingface.co, Sep. 26, 2022. https://huggingface.co/datasets/tner/ontonotes5 (accessed Mar. 21, 2025).
[3] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: the 90% Solution," ACL Anthology, pp. 57–60, Jun. 2006, Accessed: Mar. 21, 2025. [Online]. Available: https://aclanthology.org/N06-2015/.
[4] R. Weischedel et al., "OntoNotes Release 5.0," LDC2013T19, Oct. 2013, doi: https://doi.org/10.35111/xmhb-2b84.
[5] A. Walia, "Annotated Corpus for Named Entity Recognition," Kaggle.com, 2017. https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus (accessed Mar. 29, 2025).
[6] M. E. Rosadi, P. N. Andono, Muljono, and A. Z. Fanani, "A Comprehensive Evaluation of Machine Learning Classifiers in Named Entity Recognition," 2024 International Seminar on Application for Technology of Information and Communication (iSemantic), pp. 489–495, Sep. 2024, doi: https://doi.org/10.1109/isemantic63362.2024.10762517.
[7] X. Dai and H. Adel, "An Analysis of Simple Data Augmentation for Named Entity Recognition," International Conference on Computational Linguistics, pp. 3861–3867, Dec. 2020, doi: https://doi.org/10.18653/v1/2020.coling-main.343.
[8] Y. Mejova, "Sentiment Analysis: An Overview," University of Iowa, Computer Science Department, vol. 5, pp. 1–34, 2009. https://www.researchgate.net/profile/Yelena-Mejova/publication/264840229_Sentiment_Analysis_An_Overview/links/590ad68e0f7e9b1d0823eff2/Sentiment-Analysis-An-Overview.pdf
[9] S. Saha, M. Imran, Md. Motinur Rahman, and Z. Hasan, "VADER vs. BERT: A Comparative Performance Analysis for Sentiment on Coronavirus Outbreak," pp. 371–385, Jan. 2023, doi: https://doi.org/10.1007/978-3-031-34619-4_30.
[10] S. Parvez, "multiclass-sentiment-analysis-dataset," Huggingface.co, Jun. 25, 2023. https://huggingface.co/datasets/Sp1786/multiclass-sentiment-analysis-dataset
[11] E. Haddi, X. Liu and Y. Shi, "The Role of Text Pre-processing in Sentiment Analysis," Procedia Computer Science, vol. 17, pp. 26–32, 2013. https://www.sciencedirect.com/science/article/pii/S1877050913001385?utm_source=chatgpt.com
[12] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1, May 2014, Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550
[13] R. Srivastava, P. K. Bharti, and P. Verma, "Comparative Analysis of Lexicon and Machine Learning Approach for Sentiment Analysis," International Journal of Advanced Computer Science and Applications, vol. 13, no. 3, 2022, doi: https://doi.org/10.14569/ijacsa.2022.0130312. https://www.academia.edu/download/3243118/CompsYelenaMejova.pdf.
[14] M. V. Koroteev, "BERT: A review of applications in natural language processing and understanding," arXiv preprint arXiv:2103.11943, 2021. https://arxiv.org/abs/2103.11943
[15] A. Rogers, O. Kovaleva, and A. Rumshisky, "A Primer in BERTology: What We Know About How BERT Works," Trans. Assoc. Comput. Linguist., vol. 8, pp. 842–866, 2020, doi: https://doi.org/10.1162/tacl_a_00349
[16] S. Saha, M. I. H. Showrov, M. Rahman, and M. Majumder, "VADER vs. BERT: A Comparative Performance Analysis for Sentiment on Coronavirus Outbreak," in Proceedings, vol., pp. 371–385, Jun. 2023. https://www.researchgate.net/publication/371481205_VADER_vs_BERT_A_Comparative_Performance_Analysis_for_Sentiment_on_Coronavirus_Outbreak
[17] H. Allam, L. Makubvure, B. Gyamfi, K. N. Graham, and K. Akinwolere, "Text classification: How machine learning is revolutionizing text categorization," Information, vol. 16, no. 2, p. 130, 2025, doi: https://doi.org/10.3390/info16020130.
[18] M. T. Pilehvar, J. Camacho-Collados, R. Navigli, and N. Collier, "Towards a seamless integration of word senses into downstream NLP applications," ACL Anthology, pp. 1857–1869, Jul. 2017, Accessed: Mar. 22, 2025. [Online]. Available: https://aclanthology.org/P17-1170/
[19] A. Ferencz, "Topic Analysis Dataset: Sport, Book, Movie," Kaggle.com, [Online]. Available: https://www.kaggle.com/datasets/aronferencz/topic-analysis-dataset-sportbookmovie (accessed Mar. 23, 2025).
[20] M. A. Rahman and Y. A. Akter, "Topic Classification from Text Using Decision Tree, K-NN and Multinomial Naïve Bayes," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1–4, doi: https://doi.org/10.1109/ICASERT.2019.8934502.
[21] D. D. M. Pamplona, "Topic Identification and Classification of GooglePlay Store Reviews," The Philippine Statistician, vol. 71, no. 2, pp. 37–49, 2022. Presented at the 15th National Convention on Statistics, Oct. 3–5, 2022, Quezon City.
[22] Microsoft, "Sensitive information types named entities," Microsoft Learn, Mar. 2024. [Online]. Available: https://learn.microsoft.com/en-us/purview/sit-named-entities-learn [Accessed: Mar. 23, 2025].
[23] A. Kumar, M. Goel, P. Kumar, and A. Singh, "A Systematic Review on Machine Learning and Deep Learning Approaches for Mapping Crop Type Classification," Agriculture, vol. 14, no. 1, p. 135, Jan. 2024, doi: https://doi.org/10.3390/agriculture14010135.
[24] B. Hammer, E. Virgili, and F. Bilotta, "Evidence-based literature review: De-duplication a cornerstone for quality," World J. Methodol., vol. 13, no. 5, pp. 390–398, Dec. 2023, doi: https://doi.org/10.5662/wjm.v13.i5.390. PMCID: PMC10789108.
[25] Nustat, "Stratified splitting," STAT303-3 class notes, [Online]. Available: https://nustat.github.io/STAT303-3-class-notes/Stratified%20splitting.html. [Accessed: March 29, 2025].
[26] S. Srivastava, N. Bala, A. Gupta, Baddam Deepthi Priya, S. Kumar, and A. Raj, "Optimization of Sentiment Analysis Models Using Bayesian Hyperparameter Tuning," pp. 1–6, Dec. 2024, doi: https://doi.org/10.1109/icaiqsa64000.2024.10882347.
[27] B. van Aken, J. Risch, R. Krestel, and A. Löser, "Challenges for Toxic Comment Classification: An In-Depth Error Analysis," ACL Anthology, pp. 33–42, Oct. 2018, doi: https://doi.org/10.18653/v1/w18-5105.

LINK TO CODE: click here