Task 2: Climate Sentiment

2.1 Evaluation of the Methods Implemented in Part 1

Modifications to Naïve Bayes Classification

This study examined targeted improvements to a Naïve Bayes text classifier. Through systematic preprocessing enhancements, the model's performance increased substantially from 70.5% to 79.5% accuracy.

The successful modifications included lemmatization to normalize inflectional word forms, combined unigram and bigram feature extraction to capture both individual terms and contextual phrases, elimination of stopwords to reduce noise, and careful feature limitation (min-df=2, max-features=3000) to mitigate both underfitting and overfitting risks.

Notably, certain approaches proved counterproductive. Implementing trigrams increased computational demands without corresponding accuracy benefits. Similarly, TF-IDF transformation degraded performance due to the creation of overly sparse feature vectors, particularly with bigram representation.

These findings demonstrate that strategic preprocessing decisions significantly impact classification performance, with simpler approaches sometimes outperforming more complex alternatives.

Comparison of Results

Our experiment results indicate significant differences in performance among the three classification techniques applied. The modified Naïve Bayes classifier significantly performed much better than the two neural techniques with an accuracy rate of 79.5% compared to 55% for feedforward neural network and 59% for BERT-tiny as shown in Figure 1.

Naïve Bayes improved performance is attributed to effective preprocessing techniques and domain-specific feature engineering on the properties. The neural methods exhibited underwhelming performances. BERT-tiny was performance-constrained by its low capacity (2 layers, 128-dimensional embeddings), which was too low to capture complex linguistic patterns. The feedforward neural network suffered from architectural and hyperparameter choice constraints, particularly its high learning rate(5e-5). The interpretation are summarized in Table 1.

Error Analysis: Naïve Bayes incorrectly labelled phrases like "great opportunity" as risks and BERT-tiny misclassified "transition risks from policy changes" as neutral content. Naïve Bayes struggled with insidious language whereas BERT-tiny struggled with longer texts. This suggest that conventional methods are able to perform well when proper feature engineering is applied, neural approaches require more sophisticated structures, domain adaptation, and hyperparameter optimization.

Future Directions: Some of the significant improvements include using domain-specific models like ClimateBERT to solve domain shift, using novel architectures (LSTMs, transformers) to handle context nicely, optimizing hyperparameters through techniques like grid search, and model ensemble (e.g., Naïve Bayes + transformers) for leveraging their strengths and attaining improved performance.

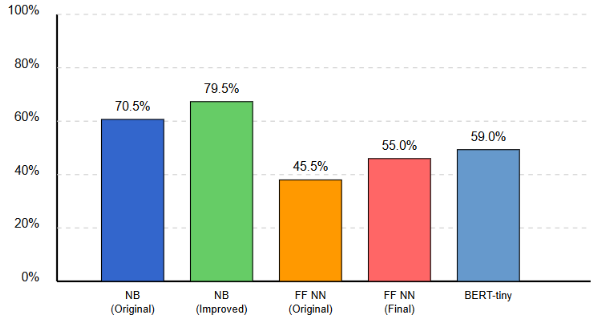


Figure 1: Classification Model Accuracy Comparison

Method	Strengths	Weaknesses
Naïve Bayes (Improved)	Fast, interpretable, handles small data	Assumes feature independence
Feed forward NN	Flexible architecture	Shallow layers, poor hyperparameter tuning
BERT-tiny	Transfer learning, context-aware	Limited model capacity, small hidden size

Table 1: Model, Strengths, and Weaknesses

2.2 Topic Analysis of Climate-Related Risks and Opportunities

Method, Motivation, Limitation

Method: For this analysis I've employed Latent Dirichlet Allocation (LDA) with Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to identify topics in climate-risk and opportunity texts. **Motivation for Choosing LDA:**

- Unsupervised Learning Approach: LDA is well-suited for exploratory analysis when we don't have predefined topics. It discovers patterns in text data without requiring labeled examples.
- Probabilistic Foundation: LDA models documents as mixtures of topics and topics as distributions over words, which aligns well with how natural language is structured.
- Interpretability: Unlike some black-box models, LDA produces human-readable topics represented by clusters of related words, facilitating qualitative analysis.
- Separation of Concerns: By separating the analysis of risk and opportunity texts, we can directly compare the distinct themes that emerge within each category.
- TF-IDF Enhancement: Using TF-IDF weighting (rather than simple count vectorization) helps prioritize words that are important within documents but not too common across the entire corpus, improving topic distinctiveness.

Limitations of LDA: It struggles with short texts and requires predefined topic numbers, which may not reflect natural topic structure. It lacks semantic understanding, relies on a bag-of-words approach that ignores context, and often produces topics that are not conceptually coherent for human interpretation.

Comparison of Different Approaches

Two variants of the LDA method to determine which is best for this dataset has been executed. Variation 1 uses only individual words (unigrams) and a compact 5-topic structure. It provides a basic overview by highlighting core vocabulary and key concepts with minimal contextual depth. Variation 2 incorporates both single words (unigrams) and two-word phrases (bigrams), expanding the model to 8 topics for a more detailed analysis. It improves the capture of compound ideas and technical terms.

Results

The figures below shows the visualization of 5 topics & 8 topics identified by Basic LDA and Advanced LDA respectively. Each subplot represents a different topic with its top five keywords and their relative importance. Figure 2 & 3 shows LDA for Climate Risk, while Figure 4 & 5 shows LDA for Climate Opportunity Topics.

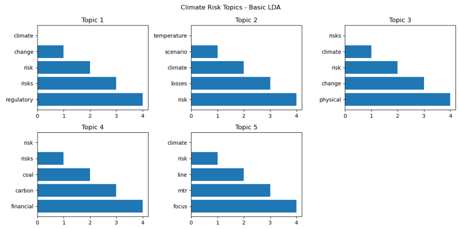


Figure 2: LDA Basic Risk(unigram 5 topics)

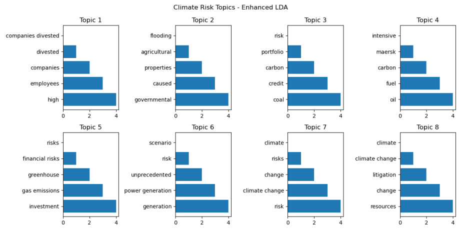


Figure 3: LDA Advance Risk(Bigram 8 topics)

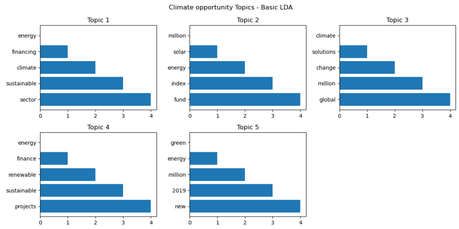


Figure 4: LDA Basic Opportunity(unigram 5 topics)

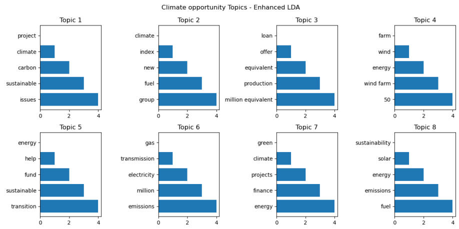


Figure 5: LDA Advance Opportunity(Bigram 8 topics)

Interpretation and Limitations

The risk topics reflect several key dimensions of climate-related corporate concerns while the opportunity topics reveal several key areas where organizations see potential benefits.

Risk: Some of the most notable climate-related corporate risks that have been determined are financial and business exposures, supply chain disruption, and carbon transition risks. The advanced model offers greater clarity regarding physical climate impacts (e.g., sea level increase) and specifically allocates agricultural risks, which were not covered in the base model.

Opportunity: Firms identify renewable energy, new market development, and green product development as significant climate-related opportunities. Both models prioritize resource efficiency, while the enriched model notably highlights climate policy, incentives, and green finance opportunities.

Limitations: Choosing 5 or 8 topics was to some extent random. There is a little topic overlap. Even with bigrams, the model is missing considerable contextual information and nuance in climate dialogue. The analysis lacks external verification of the goodness of topics. Human expert review or matching against external taxonomies would add more assurance to the results.

In conclusion, the enhanced LDA model with 8 topics and bigrams provides more accurate information on individual climate risks and opportunities while the two models both identify the major thematic areas correctly. The method illustrates strong contrasts between risk talk (interested in impacts, regulation, and carbon exposure) and opportunity talk (emphasizing energy transition, finance, and project development).

Task 3: Named Entity Recognition on Twitter

3.1 Design of Sequence Tagger

Method, Strengths, Limitations

Method: For this sequence tagging task,a fine-tuned BERT-based model for Named Entity Recognition (NER) on Twitter data was implemented . Specifically, the bert-base-cased architecture was utilized and adapted for token classification through the AutoModelForTokenClassification class from the Hugging Face Transformers library.

Strengths: Robust contextual understanding, transfer-learning capacity, and consistent tokenization. By maintaining rich semantic nuance between words through context-sensitive embeddings, it properly identifyies entities in text. Through pre-training over vast corpora, the model learns to generalize well even in situations where scarce, domain-specific data (e.g., tweets) exist. Lastly, its WordPiece sub-word tokenizer nicely deals with out-of-vocabulary items—social media slang, abbreviations, and typos—providing consistent representation of informal language.

Limitations: Computationally intensive and costly to implement at large scale. In addition, sub-word tokenization introduces complexity to sequence labeling tasks: aligning the original NER labels to the split tokens with accuracy can be tedious and error-prone, thus making it difficult to implement and evaluate the model.

Alignment of tokens with tags

The model uses BERT's tokenizer which uses WordPiece tokenization and therefore creates a mismatch between the original words and the tokens resulting from it. This is problematic in sequence tagging tasks where each token needs a corresponding label. To get rid of this, tokenize_and_align_labels function was used which:

- Tokenizes the input with the `is_split_into_words=True` parameter that keeps track of the original word boundaries.
- Maps every token to its corresponding word using `word_ids()` to maintain the relationship between tokens and original words.
- Handles sub-words either by assigning the same label to all sub-tokens of the word when label_all_tokens=True and assigning the label to the first sub-token and -100 to the rest when `label_all_tokens=False`. It also handles special tokens (e.g., [CLS], [SEP])

Example entity span

The dataset uses the BIO (Beginning, Inside, Outside) tagging scheme to encode entity spans. Looking at a random sample from the dataset [Blake, 's, Jerusalem, ', reserved, for, homosexuals, ', http://tgr.ph/k5UU7g] is encoded as: "Blake" → B-PER (Beginning of Person), "'s" → O (Not part of an entity), "Jerusalem" → B-LOC (Beginning of Location), "reserved, for, homosexuals, ', http://tgr.ph/k5UU7g" → O (Not part of an entity).

Model's Features

- Contextual word embeddings: Deep bidirectional transformers provide rich contextual representations capturing semantic and syntactic information.
- Case information: Using the cased version of BERT (bert-base-cased) preserves capitalization.
- Sub-word information: WordPiece tokenization allows the model to handle out-of-vocabulary words by breaking them into meaningful sub-units.
- Special token markers: BERT's [CLS] and [SEP] tokens provide sentence boundary information.

These features were chosen because NER in Twitter data presents unique challenges. It has informal language and non-standard spelling having limited context due to the short tweets. Variety of entity types specific to social media content and lack of consistent capitalization or formatting.

I hypothesize that these features will allow the model to capture the context necessary for entity recognition more effectively. It can handle the noisy and conversational nature of Twitter text, recognizing entities even when they appear in unconventional forms. It can also learn domain-specific entity patterns through fine-tuning.

3.2 Evaluation, Interpretation and Discussion

Performance Metrics

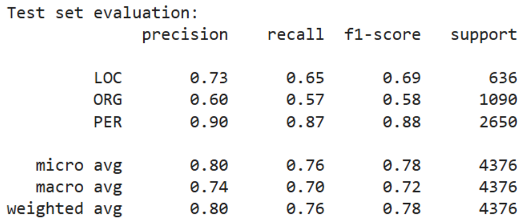
Performance Metrics Selection was done using seqeval package to compute entity-level metrics rather than token-level metrics. Because in NER tasks, correctly identifying complete entity spans is more important than individual token classifications. In addition, the classification report metrics use precision, recall, and F1 score for each type of entity.

Limitations of chosen metrics are they don't account for partial matches (e.g., identifying "Tim" but missing "Cook" in "Tim Cook") and treat all entity types equally, whereas some might be more important than others depending on the application. They don't specifically measure the model's ability to recognize entity boundaries versus entity types.

Testing Procedure

The dataset was already split into training, validation, and test sets through the TNER/BTC dataset from Hugging Face. The procedure followed was the model was trained on the training set for 3 epochs with a learning rate of 2e-5 and batch size of 16. While training, the model was evaluated on the validation set after each epoch to track progress and prevent overfitting. Final evaluation was done on the test set to assess performance. This approach ensured an unbiased evaluation of the model's ability to recognize named entities in unseen tweets, which is critical for real-world applications

Result



```
Test set evaluation:
              precision    recall  f1-score   support

         LOC       0.73      0.65      0.69       636
         ORG       0.60      0.57      0.58      1090
         PER       0.90      0.87      0.88      2650

   micro avg       0.80      0.76      0.78      4376
   macro avg       0.74      0.70      0.72      4376
weighted avg       0.80      0.76      0.78      4376
```

Figure 6: Output of NER task

Error Analysis and Improvement

The error analysis showed imprecise entity boundaries, particularly for multi-word organizations or locations, ambiguity in the heterogeneous MISC category, and lack of entities requiring more context. Slangy language common on Twitter also led to the failure to identify entities. Suggestions include the use of domain-specific gazetteers, enhancing the MISC category, document-level context modeling, and preprocessing slangy text before processing.

Further enhancement include use of BERT with Conditional Random Fields (CRF) for better sequence modeling, data augmentation to enhance entity mention diversity, and domain-specific pre-training with Twitter data. Adversarial training can also enhance the model's robustness against noisy and informal input. Overall, though the model is robust, these particular enhancements would make it considerably more feasible for practical use.

Reference

[1] Hugging Face tutorial https://huggingface.co/learn/llm-course/chapter1/1

[2] Token Classificiation Tutorial https://colab.research.google.com/github/huggingface/notebooks/blob/master/examples/token_classification.ipynb#scrollTo=vc0BSBLIIrJQ