EMATM0067 - Text Analytics Coursework

Anonymous submission

1. Task 2

1.1. 2.1 Evaluate Methods in Part 1

Improved Naïve Bayes (Task 1.1d)

In Task 1.1d, I improved the baseline Naïve Bayes classifier by replacing `CountVectorizer` with `TfidfVectorizer`, using a larger ngram range (1, 10), and applying a custom tokenizer to remove stopwords and punctuation. Limiting the vocabulary size to 10,000 reduced overfitting. These changes enhanced the model's ability to identify informative patterns and contextual cues. I also experimented larger feature sets (e.g., 50,000), but the performance dropped due to overfitting. The final setup improved the classifier's accuracy from 73.5% (1.1b) to 78.5%.

Comparison and Interpretation

Table 1 and Figure 1 summarize the performance of all three models. BERT-tiny achieved the highest accuracy (80%), followed by improved Naïve Bayes (78.5%) and the modified neural network (77%).

Model	Precision	Recall	Accuracy
Naïve Bayes (1.1b)	0.833	0.747	0.785
Neural Network (1.2)	0.787	0.752	0.770
Tiny BERT (1.3c)	**0.795**	**0.807**	**0.800**

Table 1: Comparison of precision, recall, and accuracy across models.

Naïve Bayes (1.1d): This linear model benefited from TF-IDF and n-grams but struggled to capture deeper semantics. It often mislabeled **opportunity (2)** as **neutral (1)**, especially in texts where opportunity was implied subtly.

Misclassified Text Example:

> "[...] an assessment [...] of physical risks covering the consequences of climate change [...] on the assets of Group clients."

Neural Network (1.2): A 3-layer BiLSTM with dropout and early stopping reached 77%.

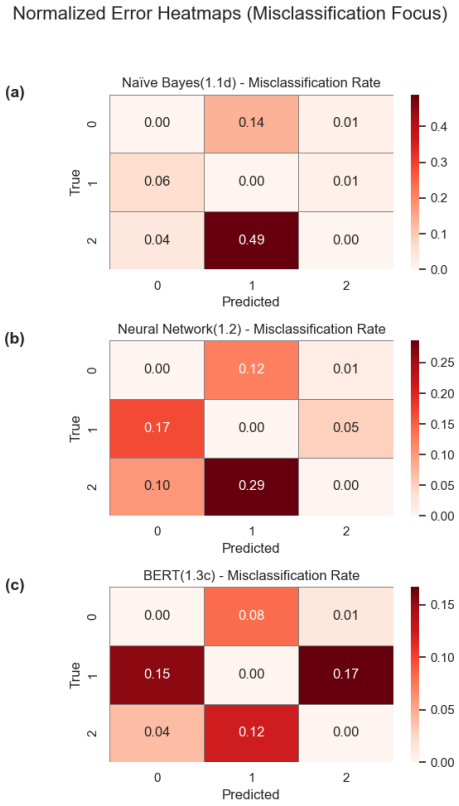

Figure 1: Normalized misclassification error heatmaps for: (a) Naïve Bayes, (b) Neural Network, and (c) BERT.

However, the model lacked contextualized semantics, frequently confusing **opportunity (2)** with **neutral (1)** in disaster-related texts.

Misclassified Text Example:

> "[...] climate change will increasingly impact our own operations [...]. For example, in 2018, the impact of natural disasters was significant [...] in the U.S."

BERT-tiny (1.3c): The best performance came from fine-tuning BERT-tiny, which leveraged transfer learning and captured context effectively. However, it sometimes over-predicted **opportunity (2)** in **neutral (1)** texts due to action-heavy language.

Misclassified Text Example:

> "invested €4.3 million [...] reduced energy consumption by 300 million MJ."

Future Improvements: To improve the neural model, integrating pretrained embeddings such as RoBERTa would enhance semantic understanding. For BERT, exploring deeper variants (e.g., `bert-base`) or better regularization could further boost performance. For Naïve Bayes, feature engineering via domain-specific keywords may help capture subtle class differences.

1.2. 2.2 Topic Modeling Analysis

Method Selection To identify climate-related risks and opportunities, I applied Latent Dirichlet Allocation (LDA) (Blei et al., 2003) using Gensim. LDA is an unsupervised probabilistic model that uncovers latent thematic structures from unlabelled text, making it ideal for exploratory analysis of climate disclosures where predefined categories are absent.

Comparative Approaches Since the LDA method needs the number of topics to be set in advance, I tried grid search in topic numbers using Bag of Words (BoW) and TF-IDF to pre-process the input text, and compared the **Coherence Score** (Röder et al., 2015) of the two approaches.

Topic Number	BoW	TF-IDF
3	0.454	0.507
6	**0.568**	0.415
8	0.484	0.498

Table 2: Coherence scores for different topic numbers using BoW and TF-IDF.

In Table 2, it is obvious that the BoW method with 6 topic numbers gets the highest coherence score (0.568), showing that the BoW method works better for this dataset. TF-IDF may excessively downweight certain important terms, making it difficult for LDA to effectively capture the underlying topic structure. Therefore, I will choose the BoW to do further fine-tuning.

I adjusted the hyperparameters of the BoW-LDA model by grid search. After that, the result is shown in Table 3[1].

[1] In the "Score" column, the left value represents the Coherence Score, and the right value represents the Perplexity

Table 3: Comparison between Default and Optimized LDA Models (BoW)

Model	Topic	Alpha	Eta	Passes	Score
Default	6	symmetric	None	10	0.567/-7.38
Optimized	6	0.5	0.01	20	0.597/-7.69

Here, the number of topics remains 6. The alpha increases to 0.5 encourages a more focused topic distribution per document, introducing a low eta value of 0.01, which promotes topic sparsity and clearer topic-word associations, and increasing the passes to 20 allows the model more iterations to converge on a stable and meaningful topic structure

Results In Table 3, the optimized LDA model demonstrates a notable improvement in topic quality and model performance compared to the default configuration. Specifically, the **coherence score** increased from 0.567 to 0.597, indicating that the optimized model produces more semantically consistent and interpretable topics. The **Perplexity** (Blei et al., 2003) slightly decreased from -7.38 to -7.69, suggesting a better generalization capability of the model on unseen data.

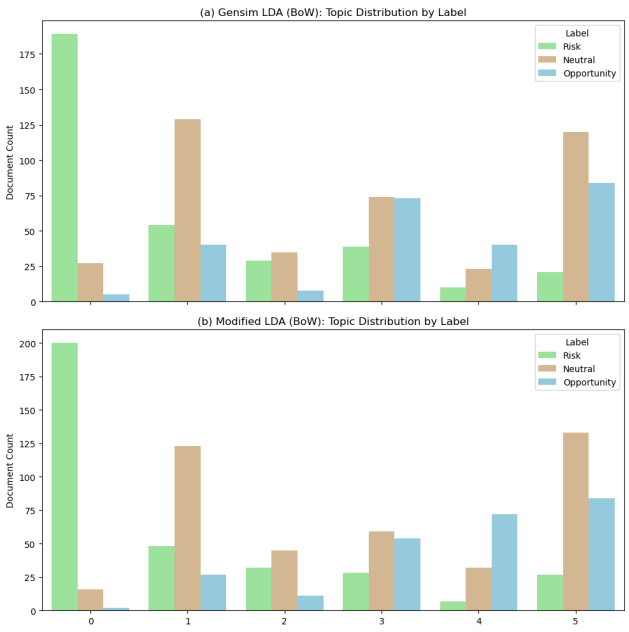

Figure 2: Topic-label distributions for BoW-LDA (top) and Modified-LDA (bottom).

For **Risk-related topics**:

Topic 0 focuses on physical climate risks, with keywords like *risk, change, climate, im-*

Table 4: Topic Keywords Comparison and Primary Label between BoW and Optimized BoW

Topic	BoW Keywords	Modified BoW Keywords	Primary Label
0	risk, change, climate, impact, physical, weather	risk, change, climate, impact, physical, weather	Risk
1	risk, climate, environmental, management, governance	risk, climate, environmental, management, governance, investment, opportunities	Neutral / Risk
2	climate, risk, carbon, impact, financial	climate, risk, portfolio, project, financial, carbon	Risk
3	climate, energy, finance, sustainable, transition, support	climate, energy, sustainable, finance, support, fund	Neutral / Opportunity
4	green, bond, finance, carbon, fund, climate	green, finance, bond, energy, carbon, fund, million	Opportunity
5	emissions, energy, carbon, reduce, power	emissions, energy, carbon, renewable, scope	Neutral / Opportunity

pact, weather. It appears in the highest number of "Risk"-labeled documents, and the Optimized model (b) shows a slight increase in its recognition over the baseline (a), improving its ability to identify physical climate risks.

Topic 1 addresses climate risk management and ESG concerns, with a governance-focused perspective reflected in keywords. The Optimized model slightly reduces the number of "Opportunity" documents, further supporting this focus on risk.

Topic 2 emphasizes the financial impacts of climate risks, particularly in investment portfolios, with keywords like *portfolio, project, financial*. The modified BoW model captures these effects.

For **Opportunity-related Topics**:

Topic 3 highlights energy transition and sustainable finance, with keywords like *energy, finance, sustainable, support*. The Optimized model shows a greater focus on "Opportunity," improving its discrimination ability.

Topic 4 centers on green finance and investment opportunities, using terms like *green, bond, finance, million, fund*. The Optimized model reduces "Neutral" labels, focusing more on "Opportunity."

Topic 5 focuses on emissions reduction and renewable energy, with keywords like *renewable, scope*, reflecting low-carbon transition themes, which are emphasized more in the Optimized model.

Intepretation Topic 0 dominates the *Risk* category (approx. 180–200 documents) in the Optimized model, highlighting the focus on physical climate risk. Topics like Topic 1 and Topic 5 show label-specific concentration, while Topic 3 is more balanced between *Neutral* and *Opportunity*. The optimized LDA model improves topic-label associations, with Topic 4 showing more "Opportunity" documents, and Topic 5 becoming more focused on specific terms like *renewable* and *scope*.

Limitations The Optimized model has limitations, including semantic overlap between topics (e.g., *climate, risk, energy*), which reduces topic separability. The Bag-of-Words approach also misses word order and context. Furthermore, some topics, such as Topic 3, show unclear categorization boundaries with a balanced distribution across labels.

2. Task 3

2.1. Task 3.1 Design a Sequence Tagger

2.1.1. Method Selection

Inspired by Lab 2, I first applied a Conditional Random Field (CRF) (Sutton and McCallum, 2010) model for Named Entity Recognition (NER) using `sklearn_crfsuite`. Although the model achieved an overall F1 score of 0.92, performance on most entity labels—except "O"—was poor.

To improve the performance, I explored pretrained language models as introduced in Lab 5. Then, I selected the `bert-base-cased` (Devlin et al., 2019) pretrained model and tokenizer, along with a custom function to align tokens and labels. This approach raised the

Table 5: Comparison of F1-Scores for NER Models on Broad Twitter Corpus(Derczynski et al., 2016)

Model / F1-Score	O	B-PER	I-PER	B-ORG	I-ORG	B-LOC	I-LOC	Weighted Avg
CRF	0.97	0.76	0.66	0.37	0.40	0.58	0.49	0.92
Bert-base-cased	0.97	0.84	0.86	0.61	0.62	0.72	0.71	0.94
Modified-Bert-base	0.98	0.85	0.90	0.62	0.65	0.73	0.73	0.95
BERTweet-base	0.99	0.89	0.90	0.70	0.68	0.75	0.79	0.96

overall F1 score to 0.94, with much better performance across all entity types.

Further enhancements included a linear learning rate scheduler and 500 warmup steps, improving stability and increasing the F1 score to 0.95. However, ORG-related labels still underperformed, likely due to domain mismatch, as `bert-base-cased` is not optimized for Twitter data.

Finally, I switch to `BERTweet-base` model (Nguyen et al., 2020), which is pretrained on Twitter, significantly improved all label scores, achieving the best overall F1 score of 0.96.[2]

2.1.2. Method Description and Analysis

In this task, I employed the BERTweet model, a Transformer-based architecture following RoBERTa, which is pretrained specifically on social media texts such as tweets. By fine-tuning this model on the NER task, it adapts well to the noisy and informal nature of Twitter language. Compared to general-purpose models, BERTweet exhibits superior domain adaptation, particularly in handling slang (e.g., *lol*), emojis, and misspellings that are prevalent in tweets. Additionally, it utilizes a WordPiece tokenizer, which mitigates the out-of-vocabulary problem by breaking down rare or compound words like *#COVID19* into manageable subword units.

However, BERTweet also comes with notable limitations. It is computationally expensive and has a relatively long training time. More importantly, due to the use of subword tokenization, token-label misalignment becomes a critical issue. For instance, words such as "New York" might be split into subword units like `["New", "Yor", "k"]`. To handle this, I implemented a token-label alignment strategy. Here is an example of an entity span before and after using the strategy:

Tokens:
[Think, you, call, ..., Gateshead, 's, ..., Shaw, 's, only, touch, ...]

Tag Labels (before alignment):
`[O, O, O, ..., B-LOC, O, ..., B-PER, O, O, O, ...]`

Tag Labels (after alignment):
`[-100, 0, 0, ..., 5, -100, ..., 1, -100, 0, 0, ...]`

Extracted Entities:
`[('Gateshead', LOC), ('Shaw', PER)]`

Tokens that don't correspond to original words (e.g., special tokens or padding) are assigned "-100", while only the first subword of tokenized words retains the original NER tag; subsequent subwords are assigned "-100" to avoid affecting loss computation. Token sequences are padded dynamically, and label sequences are padded with "-100" for alignment.

For hyperparameters, I allow the tokenizer to adjust `max_length` dynamically, reducing memory usage at the cost of minor padding noise. I used `DataCollatorForTokenClassification` for batch alignment, set the learning rate to 2×10^{-5} for fine-tuning, and chose a batch size of 16 for a balance between speed and memory. After testing different epoch counts, 3 epochs proved sufficient for convergence. BERTweet's domain-specific tokenizer handles social media entities like "@User", enhancing label recognition.

2.2. Task 3.2 Evaluation

`classification_report` from `sklearn` provides both F1 scores and accuracy as evaluation metrics. It not only shows the model's overall performance but also provides detailed

[2]All the data above is in Table 5, and the data were gathered from classification reports

metrics for each individual label. While accuracy is a commonly used indicator, F1 score is suitable for the Broad Twitter Corpus (BTC) dataset, which suffers from a class imbalance.

However, the size of "O" label vastly exceeds that of other entity types. This class imbalance can lead to inflated evaluation metrics, especially for accuracy and even F1 score, as the model can achieve high performance simply by focusing on the dominant "O" class.

During the testing procedure, the `train` split of the dataset was used to train and tune the model. Evaluation was performed on the `validation` split after each training epoch, and the best-performing checkpoint was selected accordingly. After completion of training, the final evaluation uses the `test` split to generate a classification report and visualization plots.

Table 6: Classification Report

Class	Precision	Recall	F1-Score	Support
O	0.98	0.99	0.99	30326
B-PER	0.93	0.85	0.89	2650
I-PER	0.93	0.87	0.90	269
B-ORG	0.73	0.67	0.70	1090
I-ORG	0.64	0.73	0.68	246
B-LOC	0.81	0.70	0.75	636
I-LOC	0.83	0.76	0.79	208
Accuracy		0.96		35425
Macro Avg	0.84	0.80	0.81	35425
Weighted Avg	0.96	0.96	0.96	35425

As shown in Table 6, most entity labels achieved F1 scores around 0.70. The "O" label had an extremely high F1 score (0.99), likely due to its overwhelming number of instances (30,326), which led the model to prioritize this class disproportionately. In contrast, the "I-ORG" label had the lowest F1 score (0.68), which may due to its small training size (246), making it difficult for the model to learn effective representations.

Figure 3 further supports these observations. Most entity labels demonstrated high prediction accuracy, but the "B-ORG" class experienced significant confusion, with an accuracy of only 0.66. Notably, 17% of "B-ORG" instances were misclassified as "O", and 12% as "B-PER".

Overall, the model performed best in recognizing person names, achieving high F1 scores for both "B-PER" (0.89) and "I-PER" (0.90), which were also reflected by high accuracy in the confusion matrix. In contrast, performance

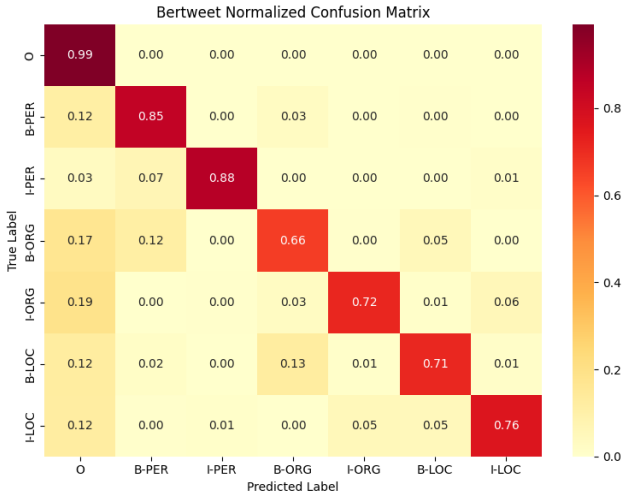

Figure 3: Bertweet Model: Normalized Confusion Matrix

on organization names was relatively poor, particularly for "B-ORG", with an F1 score of 0.70 and only 66% accuracy in the confusion matrix.

Here is a typical misclassification example:

Tokens: `[MORNING]`; **Id:** `301`
True: `B-ORG` ———> **Predict:** `B-PER`

As the most frequent error type: the model predicted "B-PER" instead of the correct label "B-ORG". In this example, the token `MORNING` appeared alone, with no surrounding context. Since `MORNING` could potentially refer to either a person or an organization, the lack of contextual information likely caused the misclassification.

To reduce such errors, one solution would be to increase the `max_length` parameter (e.g., to 256) to retain more contextual information. Additionally, incorporating contextual features, such as suffixes like `Inc.`, could help differentiate between person and organization entities.

To enhance the model's performance, further improvements could be achieved by increasing the number of training epochs and introducing early stopping to better capture the underlying patterns in the data. Due to the overwhelming number of "O" labels, class imbalance remains a critical issue. Assigning specific weights for class, such as reducing the weight for "O" while increasing weights for underperforming classes like "B-ORG", could help balance the learning process and improve overall performance on less represented classes.

3. Bibliographical References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA. Association for Computing Machinery.

Charles Sutton and Andrew McCallum. 2010. An introduction to conditional random fields.