University of Bristol – MSc Data Science (PGT)
EMATM0067: Text Analytics Coursework Report

Student ID: 2623567

2. Task 2: Climate Sentiment – Report (38%)

2.1 Evaluation of Methods

2.1.1 Modifications- Naïve Bayes Classifier (Task 1.1d):

In task 1.1d, to improve the naïve Bayes classifier, we performed the pre-processing steps in the NLP pipeline to reduce the noise and improve the text's semantic quality [9]. We implemented a custom tokeniser to lemmatise the words, which removes the affixes such as "emissions" → "emission" and gives the root word [1]. We used the `WordNetLemmatizer()` instead of the `nltk` lemmatiser to add more cleaning and preprocessing steps to improve generalisation, and have a high-quality text [1]. We removed the stopwords using the `CountVectorizer()` to eliminate the tokens which does not hold much value for our classification, such as "the" and "and" [3]. Used the n-gram range to include both the unigrams and bigrams by using `ngram_range=(1, 2)` to capture more sentiment-rich content from the classifier [9]. The final step was to lowercase the data to maintain consistency and prevent noise [1][5][9].

From the improvements here in the classifier, the model improved from the base model accuracy of 69.5% to 75.5%, reflecting the clean and semantically-rich features from the pre-processing stages. Therefore, we can achieve that while more features can bring richer information, they must be balanced with model simplicity and stability, especially for small and medium-sized datasets [3][5].

2.2.2 Comparing the Models

Task 1.2: In Task 1.2, we made modifications to the neural network classifier on the same training dataset and made a better choice of the parameters and architecture. We used a bi-directional LSTM (BiLSTM) [9] architecture with pretrained GloVe embeddings to classify the data into the classes or labels. Added a Padding of fixed sequence_length = 80 and converted to Pytorch Dataloader with batch_size = 32. We used the pretrained GloVe embeddings "glove-twitter-100". We then defined the BiLSTM Layer with 128 hidden units text classifier, and used a Dropout of 0.3 to prevent overfitting. We trained the model with 15 epochs using the Adam optimiser. After training, the final validation Accuracy obtained was 68.0%. [1][4][5][9]

Before the above modifications, I used the feedforward neural network (FFNN) [7] using GloVe embeddings and got an accuracy of about 74%. Though the Accuracy was high in comparison to the BiLSTM model lacked the capacity to model the text tokens or syntactic dependencies and missed the contextual clues that sequential models can capture.[4][5][9][7]

Task 1.3 a,b,c: In Task 1.3a–c, I applied transfer learning by fine-tuning the BERT-Tiny model (prajjwal1/bert-tiny). In 1.3a, we used the CLS token vectors representation as the document embedding. In 1.3b, we used the cosine similarity between the embedding of the chosen document and every document in the validation set. In 1.3c, we implemented the model to build the classifier and batch_size = 16 and 15 epochs and achieved an accuracy of 0.77%. The highest accuracy with contextualised embeddings and self-attention, BERT effectively captured the phrase-level and syntactic nuance more effectively than BiLSTM or traditional models.

Comparison of Results and Interpretation

Model	Accuracy (%)	Key Features Used
Naïve Bayes (baseline)	69.5	Bigrams, no preprocessing
Naïve Bayes (improved)	75.5	Unigrams + Bigrams, lemmatisation, stopword removal
Neural Network (BiLSTM+GloVe)	68	GloVe embeddings, BiLSTM, dropout
Feedforward NN (GloVe)	74	GloVe embeddings, FFNN
Transformer (TinyBERT)	77	Pretrained contextualised embeddings

Fig. 1: Accuracy comparison of the models.

The improved Naïve Bayes classifier included techniques such as lemmatisation, n-grams combinations (unigrams+bigrams), and stopword removal.

The BiLSTM model, after implementation of GloVe embeddings and sequential learning, underperformed due to its sensitivity to sequence length, data sparsity, and the need for extensive hyperparameter tuning, which misclassified the class "opportunity". The TinyBERT transformer model performed with the highest accuracy of 77% by using contextualised embeddings through self-attention and transfer learning, resulting in effective feature extraction. It had the best balance of precision and generalisation across all three classes, in comparison to the BiLSTM in recognising "opportunity" and

”risk” text sentences. Its performance is delivering the best sentiment classification tasks involving corporate climate disclosures among the Neural network and naïve bayes classifiers [2][4][5][9]. **Misclassified Example** The (Example 6) is labelled as Neutral but predicted as Opportunity. This likely occurred due to overlapping vocabulary (e.g., “carbon price”, “incentives”) in the text, which is most likely to be classified as opportunity. [1][5].

Conclusion and Future Improvements

In this Task, we explored techniques such as Naïve Bayes, NLP pipeline pre-processing steps, deep neural networks like BiLSTM and transformer-based BERT model. We performed the sentiment classification, feature engineering, and contextual embeddings, which include compositional and contextual semantics, sequence modelling, and transfer learning. For future improvements, we can incorporate coherence in topic modelling over bag-of-words on the BERT embeddings. Incorporating ensemble models, hyperparameter fine-tuning, modifying the sequence number and batch size and an increase in the genism vocabulary dimensions and sentence segmentation and further preprocessing steps can yield good results in classifying climate disclosure documents.

2.2 Topic Modelling of Risks and Opportunities (25%)

In task 2.2, the objective is to use ClimateBERT’s climate sentiment dataset and explore the topics related to the climate–related risks and opportunities.

2.2.1 Methodology for Identifying Topics: To identify the themes of climate-related risks and opportunities, we implemented the Latent Dirichlet Allocation (LDA), which is a topic modelling method. LDA is the most common approach for the classification of documents. LDA considers the data documents as a mixture of topics and is used to classify the document as the probability of each of the labels or topics which can be assigned to it. We used LDA in the training dataset, which has 800 climate-related documents/ disclosures.

We implemented various preprocessing steps to ensure our documents can be classified and identified by topic. Further, we created a dictionary of the tokens, which resulted in 4614 unique tokens and converted the dictionary into a Bag-of-words Representation, which is a vector representation of the dictionary tokens.

```
Number of unique tokens: 4614
Dictionary<4614 unique tokens: ['ability', 'aimed', 'blackrock', 'bu
Bag of Words vector: [(0, 1), (1, 1), (2, 2), (3, 1), (4, 1), (5, 1)
```

Fig. 2: Tokens, Dictionary, and BoW representation

Now we have fitted a three-topic LDA for the topics – “Risk”, “Opportunity”, and “Neutral”. LDA was trained for efficiency using Gensim’s LdaMulticore, and dominant topics were extracted per document based on the highest probability.

For the task, we implemented the LDA model to label the risk and opportunity, so we assigned the classes or labels using the BoW representation. LDA identifies by assuming the topic mixtures are static across all the documents, which may not fully capture dynamic trends [5]. Bag-of-words input ignores word order and contextual semantics [9].

To explore the topic modelling in an unseen dataset, we used the validation dataset, which has 200 documents, and performed the same pre-processing steps and applied the LDA and displayed the 5 documents’ topic probabilities and dominant topic as below. To analyse whether the topic modelling is accurately predicting the labels for the documents, we printed a random index document, raw text, Bag of words representation and the topic distribution probability. Below is the graph of the 6 documents.

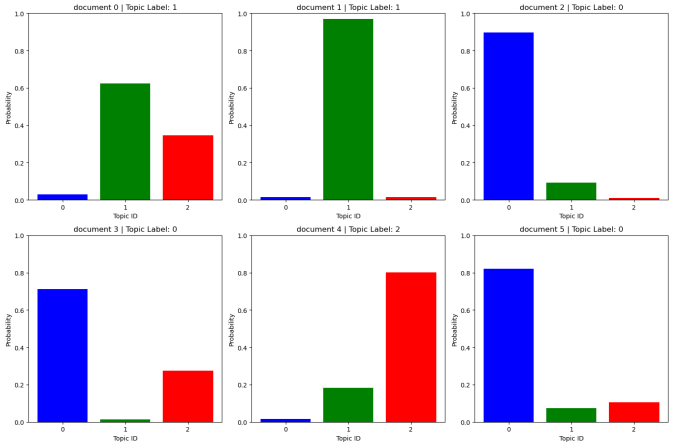

Fig. 3: Topic Distribution of 6 documents

We also visualised the Average topic distribution of the labelled classes documents.

2.2.2 Comparison of Two Variants: **LDA with TF-IDF Vectors**

In the Latent Dirichlet Allocation (LDA) model using the Bow representation, the topic distribution is based on raw term frequencies. To ensure the modelling is robust and accurate, we performed TF-IDF vectorisation and Cosine similarity. TF-IDF is a measure of how related or dependent a document is among all the documents.

We chose 3 random documents by index in the validation dataset and fitted the TF-IDF model on the training

Bow represented words and converted the validation tokens to Bow vectors and applied the TF-IDF in the 3 documents to get the sparse vectors, and further converted the sparse TF-IDF vectors to dense formats to calculate the cosine similarity.

Limitations of the TF-IDF variation are that TF-IDF vectors capture term importance but do not model latent topics directly.

Some documents (e.g., Doc 1 vs Doc 3) had low cosine similarity (0.3622), confirming thematic separation. LDA topic distributions (dense vectors) showed much higher semantic similarity (Doc 1 vs Doc 2 0.9833), validating the topic model's accuracy.

Hierarchical Dirichlet Process (HDP)

A hierarchical Dirichlet process (HDP) models a collection of random probability distributions, assigning one distribution to each group while also introducing a shared global distribution that links them together.

For this task, we used the HDP to identify the topic structures with manually fixed topic labelling numbers. It can capture highly fragmented topics without assigning a number of topics, but due to our task, it is slightly challenging to interpret the topics, "risk" and "opportunity".

2.2.3 Results and Visualisation: Following the topic modelling methodologies using Latent Dirichlet Allocation (LDA), TF-IDF variation, and Hierarchical Dirichlet Process (HDP), we present the consolidated results:

1) LDA Topic Assignment (Training Dataset)

Using the Bag-of-Words (BoW) representation, we applied LDA on 800 climate-related documents. The extracted dominant topics per document show a clear classification into Risk and Opportunity labels.

	text	tokens
0	In July 2020, BlackRock provided comments to t...	[july, blackrock, provided, comment, departmen...
1	Climate change strategy Climate risks the Comm...	[climate, change, strategy, climate, risk, com...
2	In order to achieve Environmental Future Visio...	[order, achieve, environmental, future, vision...
3	In addition to providing the facility with ste...	[addition, providing, facility, steam, needed,...
4	Paper and waste: the BBVAsinplastico project (...	[paper, waste, bbvasinplastico, project, http,...

Fig. 4: text and cleaned tokens

The topics were assigned to the documents, and also the dominant topic's probability percentage for the sample of 5 documents in the train dataset.

Document NO.	Dominant Topic	Topic Probability	Text	Topic Label
0	1	0.986788	In July 2020, BlackRock provided comments to t...	Risk
1	1	0.984337	Climate change strategy Climate risks the Comm...	Risk
2	0	0.738583	In order to achieve Environmental Future Visio...	Opportunity
3	1	0.596280	In addition to providing the facility with ste...	Risk
4	0	0.981066	Paper and waste: the BBVAsinplastico project (...	Opportunity

Fig. 5: Probability percentage of the documents topic

In training the data, the proportion of the number of documents per Topic is plotted below:

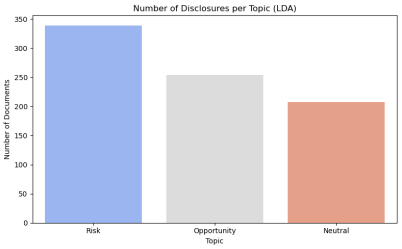

Fig. 6: Number of disclosures per topic (LDA)

Topic Distribution in Validation Dataset: We applied the trained LDA model to the 200-document validation set and the unseen dataset, which labels 84 documents as Risk, 84 as Opportunity and 32 as neutral. The dominant topics distribution for random samples showed that the Risk documents are predominantly assigned to Topic 1, and then Opportunity documents are mostly assigned to Topic 0. Followed by Neutral documents, scattered but minimal in proportion.

1) **TF-IDF and Cosine Similarity Analysis**

We employed TF-IDF weighting to negate the bias of frequent words and compared three random validation documents. The Cosine Similarity Matrix between documents showed: Low similarity ($\sim$0.3622) between dissimilar topics. High similarity ($\sim$0.9833) between documents of similar topics. This confirms that the LDA model semantically separates Risk and Opportunity themes.

1) **HDP- LDA Variation**

Using the Hierarchical Dirichlet Process (HDP), topics were generated with 3 topics. HDP captured more fragmented subtopics across the corpus. "Climate risks", "carbon initiatives", and "investment impact" topics appeared naturally. But clean Risk or Opportunity labels required manual inspection due to topic fragmentation.

2.2.4 Conclusion: The LDA model was able to separate documents that talked about climate-related opportunities and climate-related risks from one another with regard to dominant keywords and probability distributions. TF-IDF made topics clearer by prioritising less frequent words, allowing LDA to focus on words. HDP showed the advantage of flexible, automatic topic identification, finding imputed sub-themes without pre-specifying a topic number, but requiring hand interpretation. Hence, LDA, TF-IDF, and HDP combination provided the multi-aspect description of climate sentiment-topics.

3. Task 3: Named Entity Recognition on Twitter (30%)

For Task 3, we are implementing the named entity recognition of Twitter social media to get information on

the Public's opinion and events by extracting information on people, organisations, and locations by training and testing the NER tagger with the Broad Twitter Corpus (BTC) dataset.

3.1 Sequence Tagger Design and Features (17%)

a. Chosen Model, Strengths and Limitations: As the main and baseline model, we have implemented the Conditional Random Fields (CRF) model, which is a conditional probabilistic discriminative model for sequence tagging or labelling [6][9]. The CRF is a log-linear model commonly used in Natural Language processing, Feature engineering with CRF POS tagger and Named Entity recognition [6].

Strengths: Conditional Random Fields is ideally the best sequence tagger for an observation sequence because of the BTC dataset with unstructured data to identify the people, organisations and location. It can make structured predictions with named entities and POS tagging, and flexible feature engineering. The CRF often outperforms the Deep neural networks and is very effective in small or medium Datasets.

Limitation: The limitation of this model is that it can extract the rich, overlapping features, but it does require extensive manual feature engineering.

We explored the BiLSTM + CRF sequence tagging model to address the limitations of the CRF model as well as increase the performance through improvements to get an in-depth analysis. BiLSTM CRF model is generally considered a very effective model for NLP tasks as it captures the contextual features from the input sequence, and with the CRF Layer on top, it will give more accurate label predictions.

BiLSTM is a Recurrent neural network; it has two series of LSTM, one which covers the text embeddings in the forward direction and another is modelled above the first series, which handles the tokens from the backwards direction [4]. In this model, in each sentence, the dependency between the neighbouring adjacent tokens can be captured much better and the CRF layer is then applied on top to model label dependencies and enforce valid BIO tag sequences.

Strengths and Limitations: BiLSTM efficiently learns features such as syntax, word shape, and context-rich tokens with minimal feature engineering processes due to its bidirectional approach. The combination with CRF improves the model by predicting the named entities in the unstructured data.

The limitations of BiLSTM models require larger datasets and more training time compared to feature-based CRF, which performs well with small datasets.

In addition to the CRF and BiLSTM+CRF models, we also fine-tuned a TinyBERT model (prajjwal1/bert-tiny) for token classification in order to compare performance with transformer-based contextual embeddings.

TinyBERT is a condensed version of the complete BERT architecture, but with significantly fewer parameters while still maintaining reasonable performance, allowing more efficient inference and training. We initially attempted to fine-tune the whole BERT model, but training was slow and computationally expensive. TinyBERT, however, was light and efficient and therefore appropriate for low-resource environments.

Strengths: TinyBERT includes that it is able to learn long-range token dependencies, and semantic and syntactic features are learned automatically through pretraining and fine-tuning.

Limitations: But with its smaller model capacity, TinyBERT might be unable to capture certain fine-grained linguistic cues versus the full BERT model and hence can have an effect on performance in more challenging or fuzzy entity spans.

b. Tokenizer Alignment: For the CRF and BiLSTM+CRF modelling, no tokeniser alignment was necessary as it is modelled on word-level tokens provided by the BTC dataset. Whereas in TinyBERT, which uses WordPiece tokenisation, alignment was necessary. We assigned the original BIO tag to the first subword of each token and masked subsequent subwords with -100 during training, ensuring that loss and evaluation only considered complete original tokens.

c. Example Entity Span Encoding:

["It", "'s", "like", "this", "Hunico", "/", "TedDiBiase", "match", ..., "Cole", "&", "Josh", "did", ..., "wwe"],

In the above list of tokens, the BIO tagging results: "Hunico" is labelled as B-PER (beginning of a person entity) and "Cole" and "Josh" are also correctly identified with B-PER. For the token "wwe" it is tagged as B-ORG, marking the beginning of an organisation name. All other non-entity tokens are assigned the O (Outside) label.

This example helps us understand how the entities are identified in a BIO tag for our task to recognise the person, locations and the organisation.

Another example of entity span encoding from the BTC dataset in the CRF model for multi-token entities. We print one such entity BIO tag, which resulted in -

”New York Times” is annotated as B-ORG, I-ORG, I-ORG for [“new”, “York”, “times”], and ”Barack Obama” as B-PER, I-PER for [“Barack”, “Obama”]. This shows us how important the BIO tagging which in capturing and recognising the tokens with semantic meaning.

d. Features Used and Hypotheses: The feature used in the CRF model are POS tagging for extracting the word classes and Feature Engineering [1]. POS tagging processes a sequence of words and defines a part of speech tag to each token or word, which is very useful to understand the meaning (eg, Noun, Verb, Adverb, Adjective etc) of the tokens [1]. We performed several token preprocessing steps, such as Stopword removal and lemmatisation, which reduces the words to their root form using the WordNetLemmatizer() were also performed. Instead of removing the emojis, we converted them to word tokens using the emoji library. Feature extraction for the CRF is one of the important NLP steps to extract the contextual features. We performed the Lowercasing of each token, removal of the suffixes and sentence boundary indicators. These methods help to get the refined tokens with meaningful and lexically rich quality content.

To improve the accuracy of CRF model, we incorporated the Feature engineering steps in the BiLSTM + CRF model, which uses word embedding using GloVe 100-Dimensional token vectorisation to analyse the semantic similarity. Character-level vocabulary extraction. The LSTM layers of BiLSTM perform the sequence labelling, and CRF is used for prediction by using the Viterbi decoding algorithm.

We further explored the Tinybert model, which helps to fine-tune the data to learn rich features, including semantic meaning using self-attention, token dependencies, and subword-level information. Tinybert will be an effective way of modelling large datasets with highly noisy data, such as the Twitter data.

3.2 Evaluation and Discussion (13%)

a. Performance Metrics and Limitations: For the evaluation of the models, we used Precision, Recall, F1-Score, and Token-level Accuracy as the performance metrics. Precision allows us to know how accurately correct are the predicted named entities are, and Recall gives us the proportion of actual positive entities identified by the model. It is the best practice to have a balanced view of precision and recall; the F1 Score is the harmonic mean of the precision and recall. We additionally calculated the token-level accuracy of the classified tokens, which also lets us understand how accurate is the model’s performance. [8][9] These metrics are common and best practices in NER evaluation, and in particular, the F1-score is considered for the performance measure as it balances both precision and recall. [8][9]

Token-level accuracy can be misleading because non-entity tokens are dominant and noise in the data. Therefore, during the classification and model evaluation, we have also printed excluding the non-entity tokens.

CRF Classification Report (excluding 'O'):

	precision	recall	f1-score	support
LOC	0.830	0.386	0.527	844
ORG	0.683	0.258	0.375	1336
PER	0.814	0.811	0.812	2919
micro avg	0.798	0.596	0.682	5099
macro avg	0.775	0.485	0.571	5099
weighted avg	0.782	0.596	0.650	5099

Token-level Accuracy: 0.9355

Fig. 7: Performance metrics of CRF model

Classification Report BiLSTM + CRF:

	precision	recall	f1-score	support
LOC	0.65	0.35	0.46	636
ORG	0.44	0.27	0.33	1090
PER	0.77	0.86	0.81	2650
micro avg	0.70	0.64	0.67	4376
macro avg	0.62	0.50	0.53	4376
weighted avg	0.67	0.64	0.64	4376

Token-level Accuracy: 0.9360

Fig. 8: Performance metrics of BiLSTM + CRF model

Classification Report TinyBERT:

	precision	recall	f1-score	support
LOC	0.790	0.325	0.461	636
ORG	0.604	0.281	0.383	1090
PER	0.797	0.859	0.827	2650
micro avg	0.769	0.637	0.697	4376
macro avg	0.730	0.488	0.557	4376
weighted avg	0.748	0.637	0.663	4376

Fig. 9: Performance metrics of TinyBERT model

b. Dataset Splits: The Broad Twitter Corpus (BTC) dataset is pre-split into: Training set (∼70%), Validation Set (∼15%) and the

We used these splits as follows: Models were trained on the training set. Validation set was used to monitor the model’s loss, tune hyperparameters (learning rate, batch size, number of epochs). After training, the final models were evaluated on the test set, which remained unseen during training and validation.

All evaluations were conducted only on the test set to ensure fair comparison across models.

c. Results : According to the bar plot, the CRF model has the best performance on LOC and ORG entities, benefiting from explicit feature engineering, while TinyBERT performed well with the highest F1-score on PER entities (82.7%), leveraging transformer-based contextual embeddings.

BiLSTM + CRF performed well, especially for person entities, but lagged for location and organisation labels in comparison to the other two models.

Overall, CRF remains the most robust model for structured entity types in noisy Twitter data, while TinyBERT shows promise for adapting to person name recognition.

BiLSTM + CRF acts as an effective middle ground, improving over basic CRF in some cases while maintaining relatively low computational cost compared to transformers.

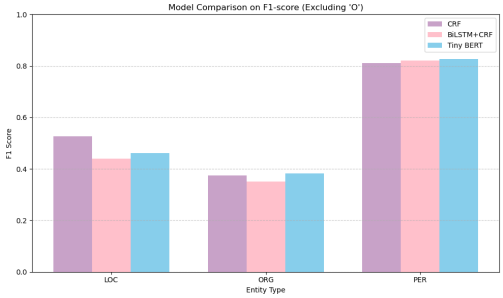

Fig. 10: Performance metrics comparison

d. Misclassification Analysis and Future Improvements: In the CRF model: **Tokens:** ['morning', 'met', 'Senators', 'Inabo', 'Senior', 'from', 'Palau', 'discuss', 'role', 'Chair', 'Public', 'Works'] **True Labels:** ['PER', 'PER', 'LOC']

Predicted Labels: ['ORG', 'O', 'LOC']

Similarly, in the BiLSTM+CRF model, the misclassification where "Senators" was labelled as ORG and "Inabo" was missed. Whereas in the TinyBERT model, both the tokens "Senators" and "Inabo" were predicted as Other entities and "Palau" was incorrectly labelled as PER instead of LOC. The common errors in all the models are the misclassification between person (PER) and organisation (ORG) entities due to the other non-entities noise in the Twitter data.

Future Improvement: In the CRF model to get more semantic relations between tokens, we can use contextual word embeddings in CRF features. This would combine a typical CRF structure with additional word representations, better enriching entity recognition. For BiLSTM+CRF model, we can improve pre-trained social media embeddings such as GloVe-Twitter by increasing dimensions. And for TinyBERT we can use bigger transformer models such as BERTweet, which are better suited for noisy Twitter text.

REFERENCES

[1] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analysing Text with the Natural Language Toolkit*, O'Reilly Media, 2009.

[2] A. Rogers, O. Kovaleva, and A. Rumshisky, "A Primer in BERTology: What We Know About How BERT Works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020.

[3] P. Goyal, S. Pandey, and K. Jain, *Deep Learning for Natural Language Processing: Creating Neural Networks with Python*, Apress, 2018.

[4] F. Chollet, *Deep Learning with Python*, 2nd ed., Manning Publications, 2021.

[5] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. draft, Stanford University, 2023.

[6] K. Balog, *Entity-Oriented Search*, Springer, 2018.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[8] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed., O'Reilly Media, 2019.

[9] J. Eisenstein, *Introduction to Natural Language Processing*, MIT Press, 2019.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[11] J. Silge and D. Robinson, *Text Mining with R: A Tidy Approach*, O'Reilly Media, 2017.

[12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[13] D. M. Blei, "Probabilistic Topic Models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[14] Y. Goldberg, *Neural Network Methods for Natural Language Processing*, Morgan & Claypool Publishers, 2017.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint*, arXiv:1301.3781, 2013.

[16] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,"*arXiv preprint*, arXiv:1810.04816, 2019.