UNIVERSITY of HOUSTON

EDS 6397 – Natural Language Processing

BBC News Article Classification

Submitted by

MADHUKUMAR GOPAL - 2354612

Under the guidance of

Dr. Nima Ekhtari, Ph.D.

Cullen College of Engineering

University of Houston -Main Campus

Table of Contents

1. Introduction

1.1 Problem Statement

With so much information available online, it's important to classify news articles accurately so people can easily find what they're looking for. This project focuses on categorizing BBC news articles into five groups: Business, Entertainment, Politics, Sport, and Tech. The goal is to build a system that organizes articles better, makes them easier to browse, and helps users quickly access relevant information, even when some articles might have overlapped or similar topics.

1.2 Literature Review

Text classification is an important task in NLP, and recent advancements in machine learning have made it more accurate and efficient. This study compares three models: DistilBERT, ALBERT, and an RNN with a Feed-Forward Neural Network (FFN) classification head.

DistilBERT is a smaller and faster version of BERT [1]. It keeps about 97% of BERT's accuracy but uses fewer resources by employing a method called knowledge distillation (Fig.4). In this process, a smaller "student" model learns from a larger "teacher" model. By removing unnecessary components like the Next Sentence Prediction (NSP) task and focusing on Masked Language Modeling (MLM), DistilBERT's simpler design (Fig.3) making it suitable for datasets like BBC news articles.

ALBERT improves efficiency by reducing the number of parameters through techniques such as parameter sharing and factorized embeddings [2]. Instead of NSP, it uses Sentence Order Prediction (SOP), which helps better understand the flow of sentences. Although ALBERT (Fig.5) is pre-trained on large datasets, it often needs fine-tuning on smaller datasets to give the best results, making it a strong option for complex text classification tasks.

RNNs are built to process sequential data by maintaining a hidden state that evolves over time. In text classification, this is generally done with Word2Vec embeddings to convert words into dense vector representations. An FFN head using dense layers to map the output of RNN to class probabilities is used to predict the category for each input [3]. However, RNNs (Fig.6) have some challenges, like handling long-term dependencies, vanishing gradients, and slower processing compared to transformer-based models. These limitations can affect their performance when dealing with overlapping categories, such as "Tech" and "Business."

1.3 Objective

This project tries to do a comparison among three different models: DistilBERT, ALBERT, and RNN with the Feed Forward Neural Network (FFN) head for classifying the BBC news articles. Evaluation of precision, recall, and F1 scores over the five categories gives the strengths and weaknesses of each model.

2. Dataset Used

2.1 Dataset Overview

The dataset used for this project is the BBC News Articles dataset which is available publicly on Kaggle [4]. It contains 2,225 news articles categorized into five distinct classes: Business, Entertainment, Politics, Sport and Tech.

2.2 Key Features of the Dataset

- Total Records: 2,225 articles.
- Categories: Articles are evenly distributed among five categories (Fig.1):
    - Business
    - Entertainment
    - Politics
    - Sport
    - Tech
- Text Characteristics (Fig.2):
    - Average word count: approximately 384 words per article.
    - Average text length: around 2,264 characters per article.

2.3 Preprocessing Steps

To prepare the dataset for the model, the following preprocessing techniques were applied:

- **Special Character Removal:** All the unwanted symbols and extra spaces are removed to have clean text input.
- **Numeric Normalization:** Numerical values normalization (For Example: "5M" becomes "5 million").
- **Financial Unit Conversion:** Ensured consistency in financial terms and formats for uniformity.
- **Custom Preprocessing:** The specific text cleaning steps followed in order to make the RNN model learn better from sequential data.
- **Embedding Preparation:** Pretrained Word2Vec embeddings were utilized for the RNN-based model.

2.4 Exploratory Data Analysis (EDA)

1. **Category Distribution**: The dataset is balanced across all five categories, ensuring no significant class imbalance (Fig.1)

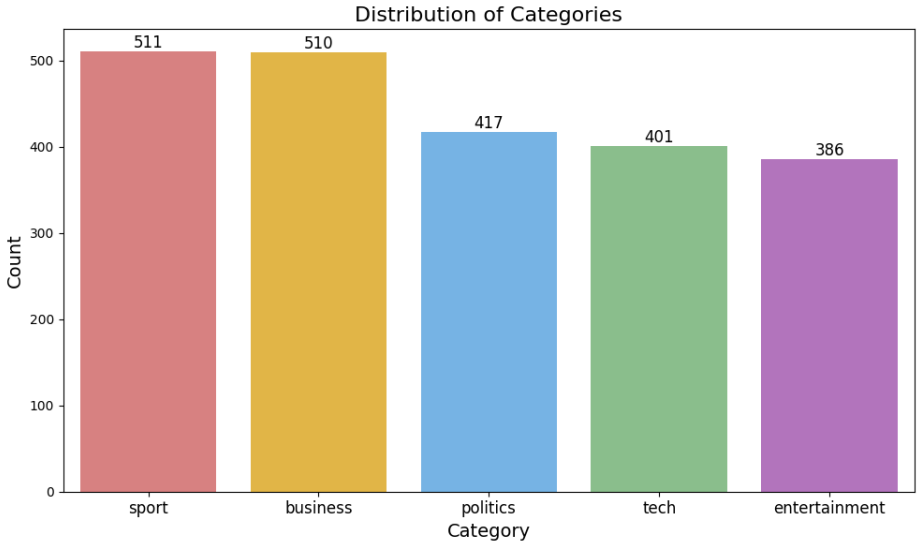

Fig.1. Distribution of Categories

2. **Text Length Distribution:**

- Most articles typically fall within the range of 200 to 400 words. (Fig.2)
- The distribution of text lengths is consistent across categories which supports a fair evaluation of all models.

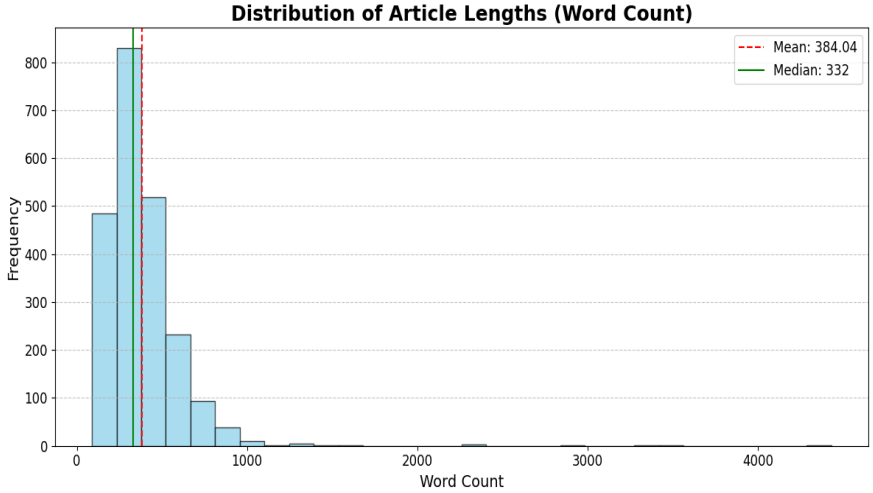

Fig.2. Distribution of Article Length

The dataset is diverse and well balanced which makes it an ideal option for testing and evaluating text classification models.

3. Methodology

In this project, we focus on implementing and evaluating three models: DistilBERT, ALBERT, and RNN combined with a Feed Forward Neural Network (FFN) for classification. These models categorize BBC news articles into five categories: Business, Entertainment, Politics, Sport and Tech.

3.1 Workflow

The overall workflow consists of the following steps:

3.1.1 Input and Preprocessing:

- Input data (BBC news articles) were preprocessed to clean the text and prepare it for model training.
- Preprocessing steps include removing special characters, normalizing numeric values, and applying Word2Vec embeddings for the RNN model.

3.1.2 Pre-trained Models:

- **DistilBERT:** A pre-trained model downloaded from Hugging Face [5] was fine-tuned on the dataset. DistilBERT simplifies the architecture of BERT and focuses on Masked Language Modeling (MLM) for better efficiency.
- **ALBERT:** This is the pre-trained ALBERT model, also from Hugging Face [6], optimized with parameter-sharing and Sentence Order Prediction (SOP).
- **Spacy:** This pretrained model (en_core_web_lg) [7] is used for tokenization. That offers an efficient and effective way to prepare the text for modeling: proper tokenization ensures clean input.
- **Word2Vec for RNN:** Pre-trained Word2Vec embeddings [3] were utilized in order to obtain dense vector representations for words in the RNN model. This embedding were trained on a large corpus which enriched the RNN by capturing semantic relationships.

3.1.3 Model Evaluation:

- Metrics such as precision, recall, and F1-score are calculated for each category to compare model performance.

3.2 Model Architectures

3.2.1 DistilBERT:

In DistilBERT, the pre-trained layers are kept unchanged to retain the language knowledge that model already learned. A classification layer, which is a fully connected dense layer, is added on top. This new layer is fine-tuned using the BBC dataset to predict the right category of each article. By freezing the pre-trained layers, the model efficiently reuses what it learned earlier and focuses on improving the classification head for this specific task. This method reduces computational costs and helps the model adapt to the dataset without needing to retrain the entire network.

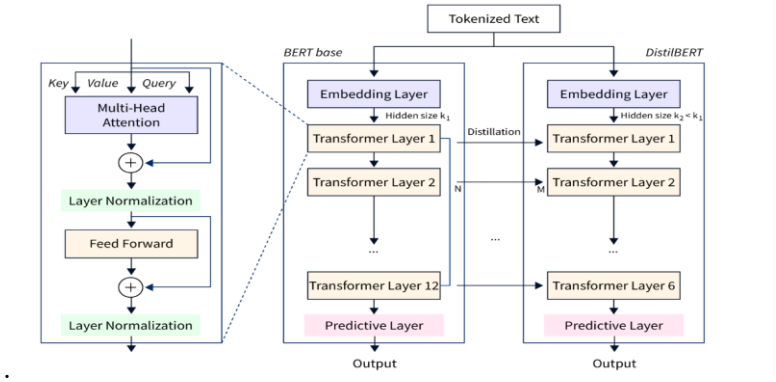

Fig.3. The architecture of the DistilBERT model

This diagram (Fig.3) explains how DistilBERT is created from BERT using a method called knowledge distillation. DistilBERT reduces the number of transformer layers from 12 to 6 while keeping most of BERT's accuracy by learning from the larger BERT model. It removes extra tasks like Next Sentence Prediction (NSP), making it faster and easier to use. This design helps the model focus on specific tasks while using fewer computing resources.

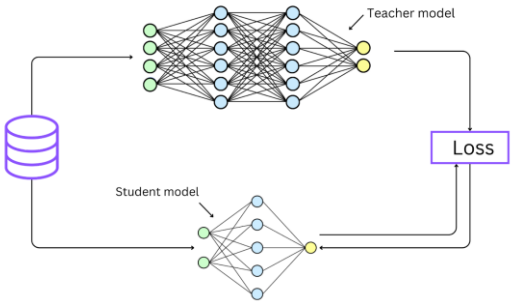

Fig.4. Knowledge Distillation

DistilBERT is trained to act like a smaller version of the larger BERT model using a method called knowledge distillation (Fig.4). This process helps the smaller model learn to make predictions similar to the teacher's model while using fewer parameters. After trying couple of **hyperparameters**, it seemed the

best setup included a learning rate of (0.00005), a batch size of (16) for training and (64) for evaluation, and (30) epochs.

3.2.2 ALBERT:

The pre-trained transformer layers in ALBERT are kept frozen to retain the language understanding it learned during pretraining. A classification head, made of a fully connected dense layer is added on top and fine-tuned using the BBC dataset to predict the correct category for each article. Freezing the pre-trained layers allows the model efficiently reuse its pretraining, while the classification head is adjusted for this specific task. ALBERT also uses parameter sharing and factorized embeddings to save memory and is fine-tuned with Sentence Order Prediction (SOP) for better sentence understanding.

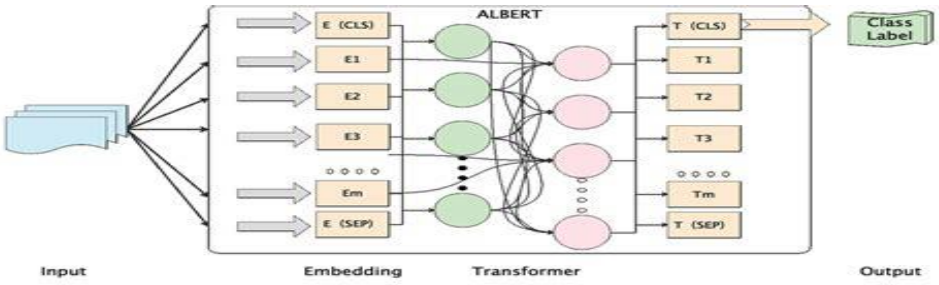

Fig.5. The architecture of the ALBERT model

The input text is split into tokens, processed through an embedding layer, and then passed through multiple transformer layers with shared parameters. The output is sent to a classification head to predict the category label (Fig. 5).

For ALBERT, hyperparameter tuning was done and the best configuration included a learning rate of (0.0005), a batch size of (16) for both training and evaluation, and (50) epochs.

3.2.3 RNN with FFN Classification Head:

RNNs are great at handling sequences like sentences or paragraphs because they keep track of context through their hidden states. These hidden states act as a memory, helping the RNN understand the relationships within the sequence. The FFN head uses the RNN's output, applies weights on it and then predicts the final class label.

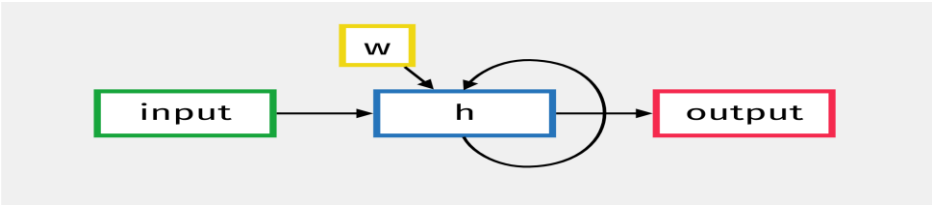

Fig.6. RNN model

The RNN (Fig.6) processes data (green) step by step. At each step, it updates its hidden state (blue), which acts as memory to store information from the past. The weights (yellow) determine how the hidden state and output are updated. The output (red) at each step reflects the previous context, and the recurring loop ensures connections between steps are retained.

The model starts with a pre-trained Word2Vec embedding layer that converts text into dense vectors, keeping the embeddings fixed. It includes two Simple RNN layers, each with 64 hidden units, to handle the sequential data. Then the RNN output is passed to a dense layer with 128 units, followed by a dropout layer to prevent overfitting. Finally, a dense layer predicts the output categories. The whole model has 29,315,045 total parameters, with only 9,771,681 being trainable, focusing on optimizing the classifier head (Fig.7).

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 3963, 300)	9,731,100
simple_rnn (SimpleRNN)	(None, 3963, 64)	23,360
simple_rnn_1 (SimpleRNN)	(None, 64)	8,256
dense (Dense)	(None, 128)	8,320
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 5)	645

Total params: 29,315,045 (111.83 MB)
Trainable params: 9,771,681 (37.28 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 19,543,364 (74.55 MB)

Fig.7.Simple RNN model Summary

The embedding layer uses a pre-trained Word2Vec matrix with 32,437 words, each represented by a 300-dimensional vector. The model includes two Simple RNN layers, each with 64 hidden units, followed by a dense layer with 128 units. After experimenting with various **hyperparameters**, the best configuration was found with a learning rate of 0.001, a dropout rate of 0.3, a batch size of 32, and 10 epochs. The Adam optimizer was chosen along with categorical cross-entropy as the loss function.

4. Results

The performance of the three models: DistilBERT, ALBERT, and RNN with a Feed-Forward Neural Network (FFN) classification head was evaluated using precision, recall, and F1-score for each category (Fig.7.). The results highlight the strengths and weaknesses of each model in classifying BBC news articles.

Category	Precision (DistilBERT)	Recall (DistilBERT)	F1-Score (DistilBERT)	Precision (ALBERT)	Recall (ALBERT)	F1-Score (ALBERT)	Precision (RNN)	Recall (RNN)	F1-Score (RNN)
Business	0.94	0.94	0.94	0.77	0.77	0.77	0.77	0.8	0.78
Entertainment	0.99	0.97	0.98	0.81	0.83	0.82	0.91	0.83	0.87
Politics	0.93	0.96	0.95	0.77	0.86	0.81	0.59	0.71	0.64
Sport	0.97	0.98	0.98	0.95	0.9	0.92	0.87	0.79	0.83
Tech	0.99	0.95	0.97	0.77	0.71	0.74	0.78	0.72	0.75
Accuracy	**0.96**			**0.82**			**0.78**		

Fig.7. Comparison of precision, recall, and F1-scores for DistilBERT, ALBERT, and RNN with classifier head models across five categories

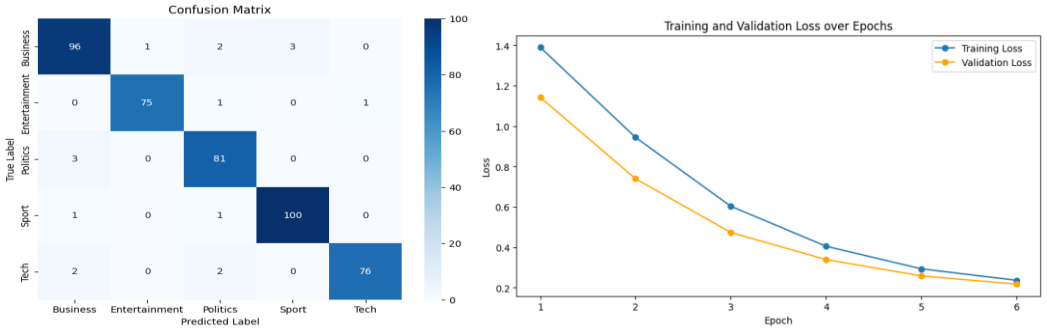

Fig.8.Confusion matrix for DistilBERT Model **Fig.9. Training and validation Loss over Epochs (DistilBERT)**

- Diagonal entries (correct classifications) dominate, indicating excellent classification performance (Fig.8)
- Around Epoch 5-6, the losses stabilize, indicating that the model has nearly converged, and further training might not significantly improve performance (Fig.9)

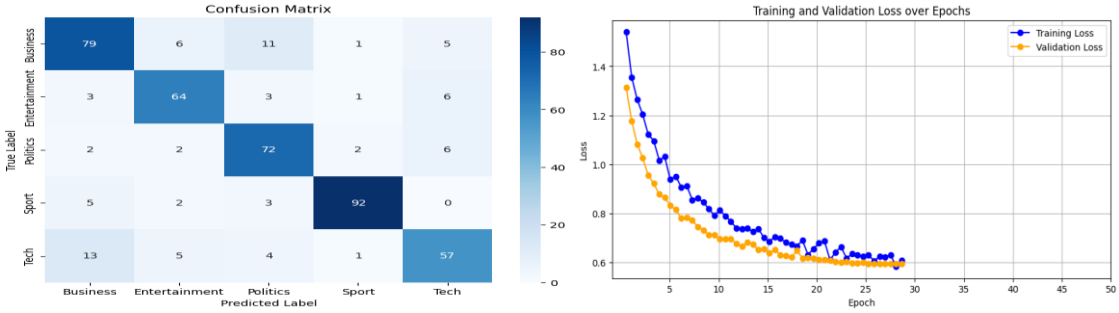

Fig.10.Confusion matrix for ALBERT Model **Fig.11. Loss over Epochs (ALBERT)**

- **Business vs. Politics:** Misclassification due to overlapping topics like economic policies.
- **Tech vs. Business:** Confusion caused by articles on technology companies.
- **Entertainment vs. Tech:** Ambiguity arises from media-tech innovations.
- **Early Stopping at Epoch 25:** Training stopped because validation loss stopped improving, preventing overfitting and saving time (Fig.11)

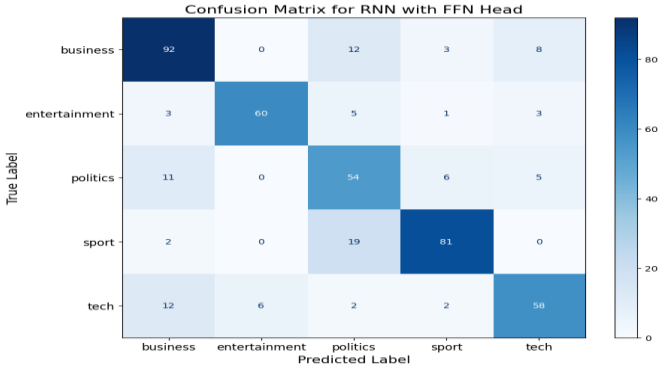

Fig.12.Confusion matrix for RNN with FFN Head

- **Best Performing (Sport):** Clear and distinct language resulted in accurate classification.
- **Politics Misclassification:** Confused with Business due to overlapping trends in economic discussions and governance topics.
- **Tech Misclassification:** Misclassified as Business because of shared themes in technology industries and market innovations.

5. Discussion

5.1 Performance Overview

5.1.1 DistilBERT

This model performed best with high precision, recall, and F1 scores on topics such as "Entertainment" (Precision: 0.99, Recall: 0.97, F1: 0.98) and "Sport" (Precision: 0.97, Recall: 0.98, F1: 0.98). (Fig.7.). It also performed slightly lower on overlapping topics such as "Tech" (Precision: 0.99, Recall: 0.95, F1: 0.97) and "Business" (Precision: 0.94, Recall: 0.94, F1: 0.94).

5.1.2 ALBERT

ALBERT performed better (Fig.7). It had good results in "Sport" (Precision: 0.95, Recall: 0.90, F1: 0.92) but struggled with overlapping categories like "Tech" (Precision: 0.77, Recall: 0.71, F1: 0.74) and "Business" (Precision: 0.77, Recall: 0.77, F1: 0.77).

5.1.3 RNN with FFN

The RNN model performed well in "Entertainment" (Precision: 0.91, Recall: 0.83, F1: 0.87) but struggled with "Politics" (Precision: 0.59, Recall: 0.71, F1: 0.64) and "Tech" (Precision: 0.78, Recall: 0.72, F1: 0.75). This is most likely because it relied on static embeddings and had limited ability to handle complex relationships (Fig.7).

5.2 Significance of Results

DistilBERT had the best F1 scores, with 0.98 for "Entertainment" and "Sport," proving that it can understand context very well. ALBERT also performed well but struggled with overlapping categories like "Tech," scoring 0.74, though it achieved an F1 score of 0.92 for "Sport." The RNN model did fine with simpler categories like "Entertainment" (0.87) but found it hard to handle more complex ones like "Politics" (0.64). These results suggest that transformer models work best for complex tasks, while RNNs are better suited for simpler ones.

5.3 Challenges

Limited training data made it difficult for ALBERT to perform at its best, as it is designed for larger datasets. Transformer models like DistilBERT and ALBERT showed high accuracy but required significant computational resources for fine-tuning. Word2Vec embeddings lacked context, which limited the ability to capture detailed features. RNNs faced difficulties in handling long-term dependencies and understanding

deeper context in the data. The major challenge was hyperparameter tuning for ALBERT, which involved trial and error and required several experiments to optimize its performance.

5.4 Recommendations

To enhance ALBERT's performance on the BBC dataset, unfreezing its last two layers could help the model fine-tune better for small datasets by capturing more specific patterns. For the RNN, using contextual embeddings like Glove instead of Word2Vec can provide richer context and improve accuracy, especially for overlapping categories like "Tech" and "Business."

6. Conclusion

In this project, we evaluated three models: DistilBERT, ALBERT, and RNN with an FFN head for classifying BBC news articles into five categories: Business, Entertainment, Politics, Sport, and Tech.

DistilBERT performed the best, handling context-rich categories efficiently. ALBERT came second, being computationally efficient but less effective at handling overlapping categories due to its parameter-sharing design, which reduces representation power, especially for smaller datasets. Lastly, the RNN did well on simpler categories but struggled with long-range dependencies and complex relationships in the text. These results highlight that transformer-based models like DistilBERT and ALBERT are better suited for handling complex tasks, whereas RNNs work well for simpler applications.

7. References

[1] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[2] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.

[3] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781

[4] J. Ferretti, "BBC News Articles Dataset," Kaggle. https://www.kaggle.com/datasets/jacopoferretti/bbc-articles-dataset

[5] Hugging Face, "DistilBERT Model," [Online]. Available: https://huggingface.co/distilbert-base-uncased.

[6] Hugging Face, "ALBERT Model," [Online]. Available: https://huggingface.co/albert-base-v2.

[7] M. Honnibal and I. Montani, "spaCy: Industrial-strength Natural Language Processing in Python," Explosion AI, 2020. [Online]. Available: https://spacy.io