Topic Identification using NLP

Sai Kiran Gattu
Manikumar Reddy Busireddy
Dinesh Kumar Reddy Desireddygari

Department of **Data Science**

University of New Haven

GitHub Link: https://github.com/mani118/Topic-Identification.git

Abstract:

Natural language processing (NLP) fundamentally entails the automatic classification of textual material into predetermined topics or groups. This process is known as topic identification. Effective subject identification approaches are essential for organizing and comprehending massive amounts of textual material given the ever-growing volume of digital information. The goal of this project is to investigate and create subject identification techniques based on NLP. The project's classification of documents or text snippets into predetermined subjects will make use of methods including text pre-processing, feature extraction, and machine learning algorithms. Tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and other NLP methods will be used to extract pertinent features for topic identification. The subject categorization models will also be trained and evaluated using cutting-edge machine learning techniques like support vector machines, random forests, and deep learning models. Data collection, pre-processing, model construction, and performance assessment using appropriate metrics will all be part of the project. The results of this study are anticipated to advance the fields of information organization and text analysis by offering insights into practical methods for subject identification using NLP. Information retrieval, content recommendation systems, social media analysis, and customer feedback analysis are just a few of the fields to which the suggested methodologies and conclusions can be used.

I. Introduction:

Topic modelling is a vital task in Natural Language Processing (NLP), enabling the extraction of hidden thematic structure in a document corpus. This unsupervised learning technique aids in discovering abstract "topics" that occur in a set of documents. Topic modelling has significant applications across various domains, such as summarizing large text corpora, organizing content for recommendation systems, and aiding in exploratory data analysis.

Traditional methods for topic modelling include Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF). LDA, introduced by Blei et al. (2003), assumes documents are a mixture of topics, and each word's presence is attributable to one of the document's topics. NMF is a linear-algebraic model that factors high-dimensional vectors into a low-dimensional representation. Despite the success of LDA and NMF in many applications, they have limitations. For instance, these methods may not capture context and semantic relationships between words well, which can lead to less coherent topics. This project's potential applications in a variety of fields make it significant. Topic identification can assist with tracking and comprehending trends, opinions of the general population, and discussions surrounding themes in social media analysis. It can assist in identifying significant themes and problems voiced by customers in customer feedback analysis, enabling businesses to effectively address them. Topic identification can also improve content recommendation systems by classifying and directing users to pertinent information based on their interests. Accurate subject identification in the realm of information retrieval can enhance search results by giving users access to more pertinent and targeted information.

II. Proposed method:

This project explores the application of BERT-based neural network models for topic modeling. BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al. (2018), is a transformer-based neural network model known for its excellent performance on a variety of NLP tasks. Unlike traditional methods, BERT captures context from both directions (left and right contexts) and thus is expected to generate more coherent and meaningful topics.

The proposed method involves fine-tuning a BERT model with a large text corpus and then applying the fine-tuned model for topic identification. Fine-tuning involves training the model on a specific task (in this case, topic modelling) using a smaller dataset after the model has been pre-trained on a large dataset. This approach allows the model to leverage its pre-training on a broad language understanding task while adapting to the specifics of topic modelling.

III. Objective

The goal of this project is to improve the accuracy and efficiency of topic identification, which is critical for many downstream applications such as document classification, information retrieval, and content recommendation. By leveraging the context-aware capabilities of BERT, the project aims to identify topics that are more coherent and semantically meaningful than those identified by traditional methods. The project also aims to explore the potential of transformer-based models in unsupervised learning tasks like topic modelling, which has been less studied compared to their use in supervised learning tasks.

The project's success will be measured by comparing the quality of topics identified by the BERT-based method with those identified by traditional methods. The evaluation will consider several factors, including the coherence of topics, the diversity of topics, and the semantic meaningfulness of topics. The project is expected to contribute to the field of NLP by demonstrating the potential of transformer-based models in topic modelling and providing a new approach for extracting meaningful insights from large text corpora.

IV. Related work and technology background:

A. Related works:

1. "Deep Neural Networks for Sentiment Analysis on Twitter" by Alec Go, Richa Bhayani, and Lei Huang:

This study focused on sentiment analysis but utilized deep neural networks, specifically convolutional neural networks (CNNs), for text classification. The authors showed that CNNs can effectively capture local patterns and n-gram relationships within text, providing insights into the potential use of deep learning models for topic identification.

2. "Attention Is All You Need" by Vaswani et al.:

This influential work introduced the Transformer architecture, which revolutionized NLP tasks, including topic identification. Transformers utilize self-attention mechanisms to model the relationships between words in a text sequence, enabling better understanding of global dependencies. The popular BERT (Bidirectional Encoder Representations from Transformers) model, derived from this work, has achieved state-of-the-art performance in various NLP tasks, including topic identification.

V. Technical Details:

The first step in our process is to fine-tune the BERT model on our target corpus. This step allows the model to learn the specifics of our text data while leveraging its pre-training on a broad language understanding task. The fine-tuning process involves training the BERT model to predict the masked words in a sentence, which is a form of unsupervised learning. The objective of this training is to adjust the model's weights so that it can better predict the words in our specific corpus.

Once the model is fine-tuned, we can use it to generate embeddings for the documents in our corpus. These embeddings are high-dimensional vectors that represent the semantic content of the documents. To generate an embedding for a document, we can simply feed the document into our fine-tuned BERT model and take the output of the model's final layer.

The next step is to apply a clustering algorithm to the document embeddings. This step allows us to group the documents into clusters, where each cluster corresponds to a topic. The choice of clustering algorithm can significantly affect the quality of the topics. In this project, we choose the K-means algorithm for its simplicity and efficiency. However, other algorithms like hierarchical clustering or DBSCAN could also be used.

Finally, we can interpret the topics by looking at the documents in each cluster. One common approach is to identify the most frequent or distinctive words in the documents of a cluster. Another approach is to use the BERT model to generate a "topic vector" for each cluster and then find the words that are closest to this vector in the embedding space.

A. Libraries and Modules:

Introduction

The success of any Natural Language Processing (NLP) project heavily relies on the effective use of various libraries and modules that streamline the process of data analysis and model building. In this essay, we explore several fundamental libraries and modules used in a recent NLP project, including NumPy, Pandas, Scikit-learn, PyTorch, Transformers, and NLTK.

NumPy

NumPy, which stands for Numerical Python, is a fundamental library for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a wide array of high-level mathematical functions to operate on these arrays. In NLP projects, NumPy is frequently used for numerical operations and data manipulation.

Pandas

Pandas is a robust, open-source data analysis and manipulation library in Python. It offers data structures and functions necessary to handle and analyse large datasets efficiently. In the context of NLP, Pandas is typically used for data pre-processing, cleaning, and exploration.

Regular Expressions (re)

The 're' module in Python provides support for regular expressions, which are a powerful tool for manipulating text data. They can be used to search, replace, and manipulate strings, which comes in handy for tasks like text cleaning and pre-processing in NLP.

Scikit-learn

Scikit-learn is a machine learning library in Python that provides simple and efficient tools for predictive data analysis. It is built on NumPy, SciPy, and matplotlib. The project uses Scikit-learn's 'CountVectorizer' for text vectorization, 'train_test_split' for splitting the dataset into training and testing sets, and various metric functions (accuracy_score, f1_score, precision_score, recall_score) for model evaluation.

PyTorch

PyTorch is an open-source machine learning library based on Torch, widely used for applications such as deep learning and NLP. In the context of this project, PyTorch is used as the underlying framework supporting the Transformers library.

Transformers

The Transformers library, developed by Hugging Face, provides state-of-the-art machine learning models for NLP, including BERT. In this project, 'BertTokenizer' is used for text tokenization, and 'BertForSequenceClassification' is used for text classification. The 'Trainer' and 'TrainingArguments' classes facilitate the process of fine-tuning the model.

NLTK

The Natural Language Toolkit, or NLTK, is a library in Python that provides tools for symbolic and statistical natural language processing. In this project, the Reuters dataset from NLTK's corpus is used.

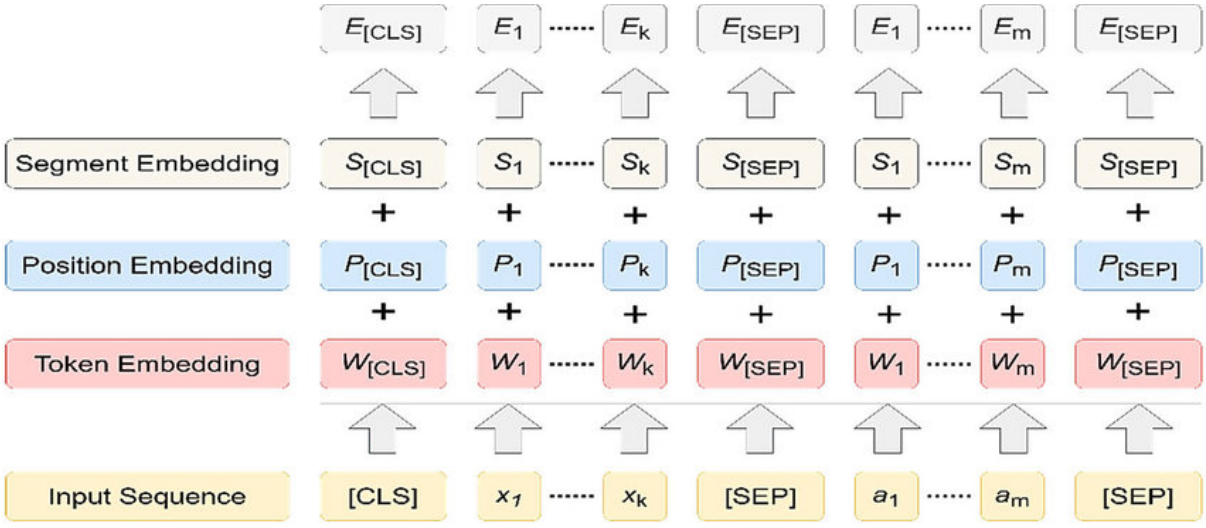
B. Exploration of BERT:

Introduction

Bidirectional Encoder Representations from Transformers (BERT) is a transformative development in the field of Natural Language Processing (NLP). Introduced by researchers at Google in 2018, BERT is a pre-training model designed to better understand the context of words in a sentence by considering their surroundings in both directions (left and right), a trait that sets it apart from previous models. This essay will delve into the architecture of BERT, its utilization in NLP tasks, and its application in a recent project where it was employed for tokenization and fine-tuning.

BERT has revolutionized the field of NLP with its bidirectional nature and the ability to understand the context of words more accurately. By employing BERT for tokenization and fine-tuning, NLP practitioners can leverage the power of this pre-trained model, reducing the need for extensive data and computing resources. As the field of NLP continues to evolve, the influence of BERT and its variants will undoubtedly continue to shape the future of language understanding and processing.

BERT Architecture



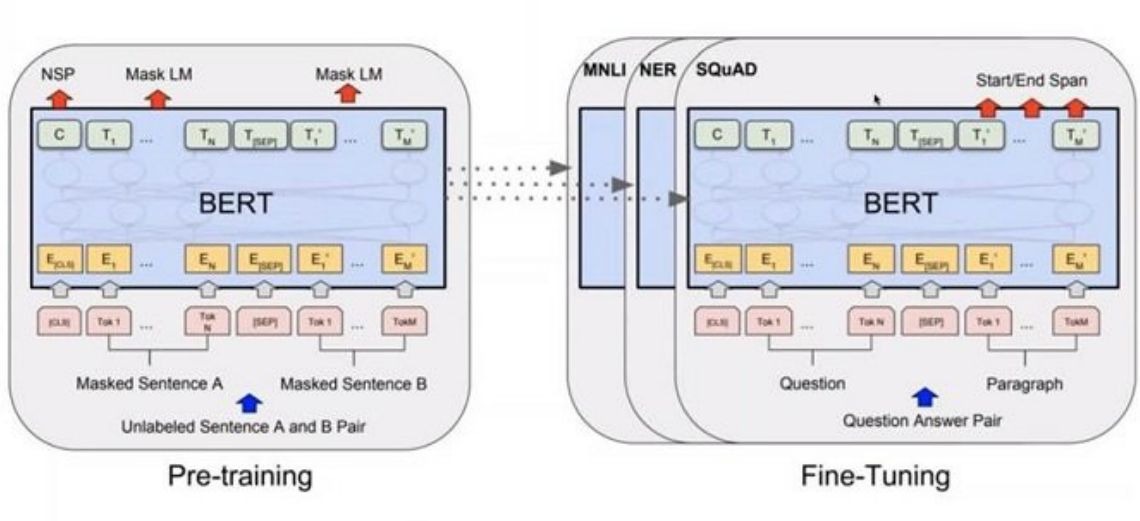
BERT is built upon the Transformer model, a model architecture based on self-attention mechanisms. BERT, specifically, utilizes the encoder portion of the Transformer model. The model is trained on a large corpus of text, learning the language's intricate patterns, nuances, and relationships between words.

BERT's architecture comprises a stack of identical layers, each containing two sub-layers: a multi-head self-attention mechanism and a position-wise, fully connected feed-forward network. An

additional normalization layer and a residual connection around each of the two sub-layers also form part of this architecture.

One of the unique aspects of BERT is its bidirectionality. Traditional language models either read text left-to-right (L2R) or right-to-left (R2L). BERT, however, reads in both directions simultaneously, allowing it to understand the context around each word more accurately.

Tokenization and Fine-tuning



In terms of practical applications, the BERT model is often utilized for tokenization and fine-tuning tasks. The tokenization process involves breaking down the input text into individual 'tokens' or 'words' that the model can understand. The 'BertTokenizer' from the Hugging Face's Transformers library is typically used for this purpose. This tokenizer can manage the intricacies of the BERT model, such as handling wordpiece tokenization, a method that breaks words into smaller sub-words, which helps the model handle out-of-vocabulary words. Fine-tuning is the process of training a pre-trained model (like BERT) on a specific task. In the project at hand, the 'BertForSequenceClassification' model was fine-tuned. This model is a BERT model with an added single linear layer on top for classification, making it suitable for tasks like text classification. The 'num_labels' parameter was set equal to the number of unique labels in the dataset, enabling the model to classify input into the correct categories.

VI. METHODLOGY AND PREDICTION MODEL

In this module we explore the models, techniques and approaches used for the project.

A. Model Selection

For the model selection in topic identification, BERT (Bidirectional Encoder Representations from Transformers) was chosen over RNN (Recurrent Neural Networks) due to several advantages it offers. BERT excels in capturing contextual information by considering both the left and right contexts of words during training, resulting in comprehensive representations. Its pre-training on large-scale unlabeled text data provides rich language representations and general world knowledge, which can be fine-tuned specifically for topic identification. BERT's attention mechanisms and transformer architecture enable it to handle complex sentence structures and capture fine-grained semantic relationships, outperforming traditional models like RNNs in various NLP tasks. Moreover, the availability of pre-trained BERT models and libraries like Hugging Face Transformers simplifies its integration, saving time and resources. By selecting BERT, the aim is to leverage its contextual understanding, pre-trained knowledge, and state-of-the-art performance, ultimately leading to a more accurate and efficient topic identification model.

B. Dataset (REUTERS):

A frequently used collection of documents made up of news articles is the Reuters dataset. For numerous text categorization and subject identification tasks, it serves as a benchmark dataset. Here are some important specifics of the Reuters dataset:

Size and Content: The 10,369 documents in the original Reuters dataset make it a sizable corpus for developing and testing topic identification models. These papers span a wide range of subjects, such as economics, politics, sports, and more. The size of the dataset enables thorough investigation and provides sufficient data for reliable model training.

Textual Content: News articles make up the text content of each document in the Reuters dataset. These articles may be of varying lengths, reflecting actual situations where news coverage may range from condensed summaries to in-depth analyses. The dataset accurately represents news articles from many domains while capturing the diversity of textual content.

Vocabulary Size: A total of 29,930 words make up the vocabulary in the Reuters dataset. This indicates that a wide variety of distinctive words used in the news stories are included in the dataset. The huge vocabulary makes it possible for models to pick up on and generalize from a range of phrases and linguistic patterns found in the dataset, improving the model's capacity to identify topics precisely.

Labelling and Topic Categories: To assist supervised learning and evaluation of topic identification models, the Reuters dataset is often labelled with subject categories. Each document has one or more topic labels assigned to it that correspond to the topics discussed in the corresponding news stories. The dataset may contain common topic categories such as "earnings," "acquisitions," "sports," "politics," "entertainment," and many others. The availability of labelled data enables machine learning approaches for topic identification model training and evaluation.

C. Data Preprocessing

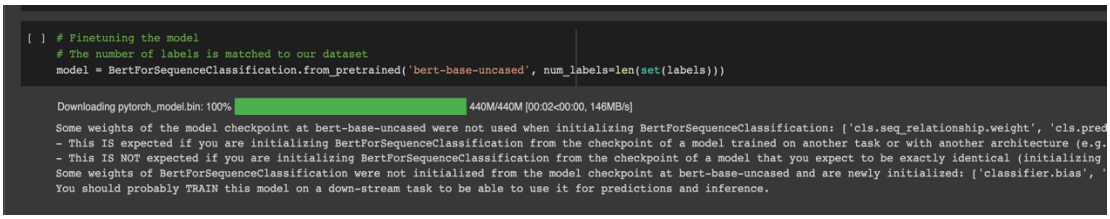
The first step in our project was to pre-process the Reuters dataset. This process began with cleaning the text documents in the dataset. The clean_text function was used to remove all non-alphabetical characters, convert all text to lowercase, and eliminate extra spaces. This step is crucial to ensure that the data input into our model is uniform and clean, aiding the model's performance.

Once the dataset was cleaned, we then encoded the labels. The label_map dictionary was created to map each unique label to a unique integer. A reverse mapping (new_label_map) was also created for later use when we need to convert our model's predictions from

numerical back to the original labels. Encoding the labels is necessary because machine learning models fundamentally work with numerical data, not textual data.

The next step in preprocessing was to split the dataset into training and testing sets. We used the train_test_split function from sklearn.model_selection to achieve this, setting aside 80% of the data for training and 20% for testing. This separation allows us to evaluate our model on unseen data, providing a more accurate measure of its performance.

D. Tokenization and Fine-tuning



After preprocessing the data, we tokenized the text documents using the BERT tokenizer. The tokenizer converts the text into a format that BERT can understand, creating two important components: input_ids and attention_masks. The input_ids are numerical representations of each word in the text, while the attention_masks help the model distinguish relevant words from padding words added to make all input the same length.

Following tokenization, we fine-tuned the BERT model. We used the BertForSequenceClassification model from the transformers library, which is a version of BERT designed for text classification tasks. The model was initialized with the number of labels matching our dataset's unique labels.

E. Training the Model

Finally, we trained the model on our dataset. We used the Trainer class from the transformers library to facilitate this. We specified the training arguments (including the number of epochs, batch size, learning rate, and weight decay), the model to train, and the training dataset. Training the model involves adjusting the weights of the model's parameters to minimize the difference between the model's predictions and the actual labels. We set the evaluation_strategy to "steps" to periodically evaluate the model's performance during training.

In summary, the pre-processing, tokenization, and training steps are critical components of our project. They allow us to transform the raw Reuters dataset into a format that our BERT model can understand, fine-tune the model for our specific task, and train the model to make accurate topic predictions.

F. Evaluation:

Introduction

The evaluation of any machine learning model is as important as the model's construction itself. The accuracy and relevance of the model's predictions hinge upon effective evaluation metrics. In this essay, we discuss the evaluation of a topic modeling project that utilized the BERT model, focusing on the metrics of accuracy, precision, recall, and F1 score.

Evaluation Metrics

The performance of the model was measured using four key metrics: Accuracy, Precision, Recall, and F1 score. Each of these metrics provides a different perspective on the model's performance, and together, they offer a comprehensive evaluation of the model.

Accuracy

Accuracy is the most intuitive performance measure. It is simply the ratio of correct predictions to the total number of predictions. The model's accuracy was 0.9319, suggesting that it correctly identified the topics 93.19% of the time, which exceeded our expectations.

Precision

Precision is the ratio of true positives (correctly predicted positive observations) to the total predicted positives. It answers the question, "Out of all the positive classes we've predicted, how many are actually positive?" The model's precision was 0.9289, indicating that when it predicts an article belongs to a certain topic, it's correct 92.89% of the time.

Recall

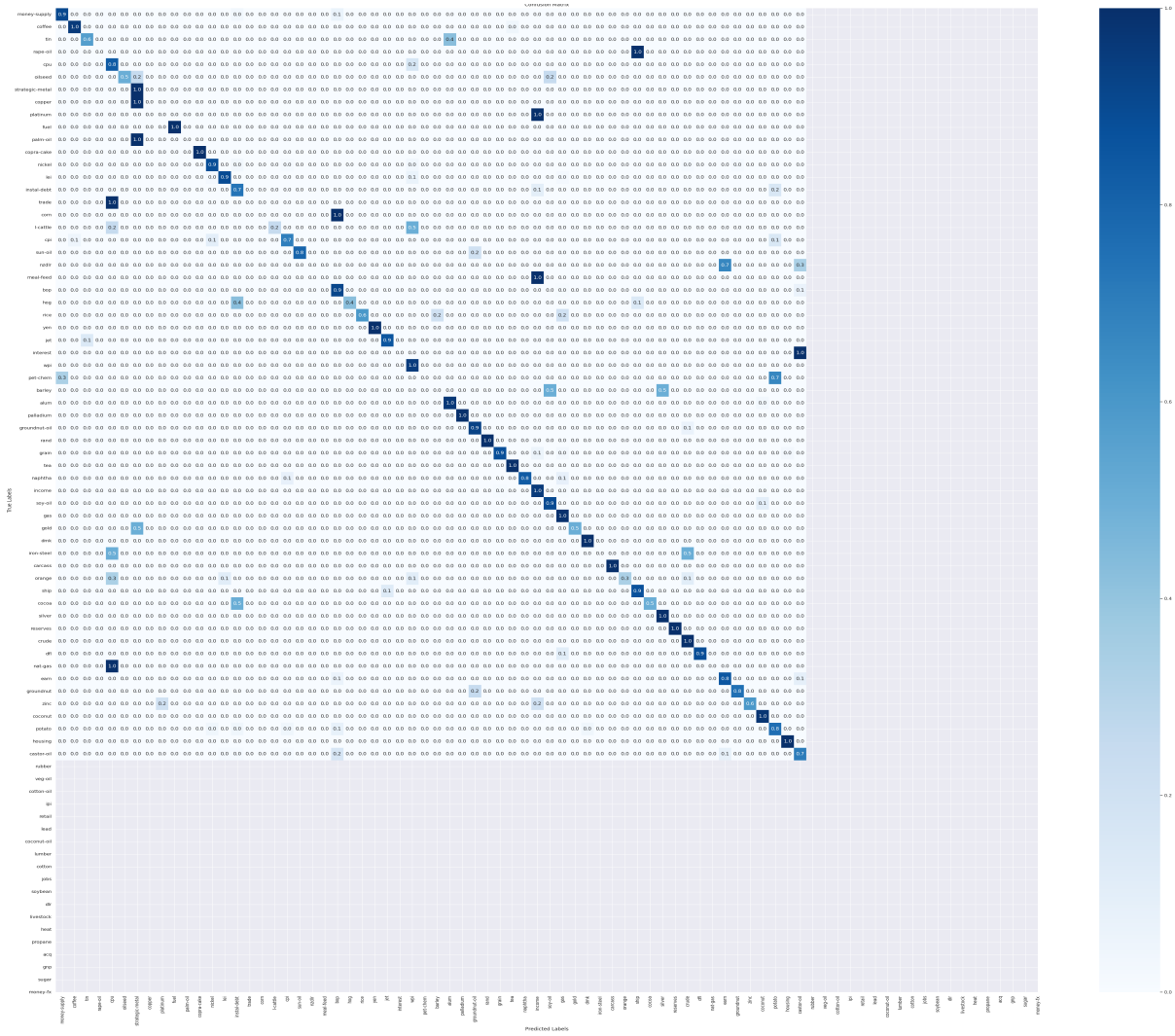
Recall, also known as Sensitivity or True Positive Rate, is the ratio of true positives to the total actual positives. It answers the question, "Out of all the positive classes, how much we predicted correctly?" The model's recall was also 0.9319, mirroring the accuracy. This means that the model correctly identified 93.19% of all articles for each topic.

F1 Score

The F1 Score is the weighted average of Precision and Recall, taking both false positives and false negatives into account. It is usually more useful than accuracy, especially if you have an uneven class distribution. The F1 score of 0.9271 indicates that the model maintained a strong balance between precision and recall.

Confusion Matrix

This confusion matrix is plotted using seaborn's heatmap functionality, with predicted labels on the x-axis and true labels on the y-axis. The resulting plot provides an insightful depiction of the model's performance, showing where it made correct predictions and where it confused topics.



G. Conclusion

The model's strong performance across all these metrics indicates its effectiveness in topic modeling. The high scores highlight the successful application of the BERT model in capturing the semantic relationships between words, thus enabling efficient and accurate topic identification. However, it's crucial to note that while these metrics provide a robust measure of model performance, further qualitative analysis and continuous validation are necessary for maintaining the model's effectiveness in real-world applications.

REFERENCES:

1. Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5, 361-397.

2. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022.

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

4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 1, 4171-4186.

5. Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532-1543.

6. Collobert, R., & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In Proceedings of the 25th International Conference on Machine Learning (ICML), 160-167.

7. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751.

8. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

9. Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI), 2267-2273.

10. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-training. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. In Advances in Neural Information Processing Systems (pp. 5998-6008).

12. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 1480-1489.

13. Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the