Zero-Shot Classification with RoBERTa

Manan Jain, Alankrit Singh and Divyanshu Singh

1 Introduction

Zero-shot text classification is a challenging natural language processing task where models must classify text into categories that were not seen during training. This capability is crucial for real-world applications where new categories may emerge frequently, and fine-tuning models for each new category is impractical. One of the major use of Zero-shot classification is for annotating unlabeled data. Human annotation requires hiring experts or large crowdsourcing teams, which can be expensive, especially for large datasets or highly specialized domains. Annotating datasets manually can take weeks or even months, whereas zero-shot classification models can label data almost instantly. Also, human annotators may introduce biases or inconsistencies, particularly for subjective tasks. Zero-shot classification provides a more uniform labeling approach.

The transformer architecture, introduced by Vaswani et al. in 2017[1], revolutionized natural language processing with its self-attention mechanism. Building upon this foundation, RoBERTa (Robustly Optimized BERT Approach) (Liu et al. 2019[2]) improved the original BERT(Bidirectional Encoder Representations from Transformers) model (Devlin et al. 2019[3]) through careful hyperparameter tuning and training methodology modifications. Key improvements include:

- `Training on Larger Datasets`: BERT was trained on the BooksCorpus (800M words) and English Wikipedia (2.5B words), which are relatively small datasets by modern NLP standards. RoBERTa uses significantly larger datasets, like, CommonCrawl News (63M articles, 76GB), OpenWebText (38GB) among others which allows it to generalize better and learn richer representations of language.
- `Longer training with larger batches`: The original BERT model was trained for 1 million steps with a batch size of 256. RoBERTa trains for 10x longer than BERT, with larger batch sizes (8,000) and more training steps (500K to 1M steps with larger data and batches). The extended training time ensures the model fully converges and leverages the larger datasets.
- `Removal of the Next Sentence Prediction objective`: BERT uses Next Sentence Prediction (NSP) as one of its pretraining objectives, where the model predicts whether one sentence logically follows another. This task was intended to help BERT understand relationships between sentences but was found to contribute minimally to performance.
- `Dynamic masking patterns`: During pretraining, BERT masks 15% of the input tokens at random. However, the masking is static—each training example always has the same tokens masked, limiting the model's ability to learn diverse patterns. RoBERTa employs dynamic masking, where the masking pattern is changed each time the same input is seen during training.

These modifications resulted in significant performance improvements across various natural language understanding tasks.

2 Problem Definition and Objectives

2.1 Problem Definition

This project addresses the following key challenges:

1. Model Setup: Implementation of a pre-trained RoBERTa model for zero-shot classification using the Hugging Face transformers library.

2. Data Processing: Preparation and preprocessing of the AG News dataset, which contains news articles across four categories: World, Sports, Business, and Technology.

3. Zero-shot Classification: Development of an effective classification system without task-specific fine-tuning.

4. Performance Analysis: Comprehensive evaluation of the model's classification capabilities.

2.2 Dataset Description

The AG News dataset is a collection of news articles from more than 2000 news sources, categorized into four classes: World, Sports, Business, and Technology. Each article contains a title and a description. The dataset includes:

- Training set: 120,000 samples (30,000 per class)
- Test set: 7,600 samples (1,900 per class)

2.3 Objectives and Evaluation Metrics

Our primary objectives include optimizing the following evaluation metrics:

1. Overall Accuracy: The proportion of correctly classified instances across all categories.

2. Per-class Metrics:

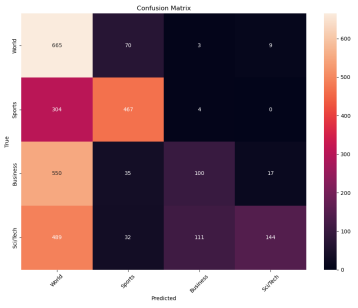
- Precision: The ratio of true positive predictions to the total positive predictions for each class, indicating the model's ability to avoid false positives.
- Recall: The ratio of true positive predictions to the total actual positives for each class, showing the model's ability to find all positive instances.
- F1-Score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
- Support: The number of samples in each class, helping understand the class distribution.

3. Confusion Matrix: It is a performance measurement tool used in classification tasks. It compares the actual labels of the dataset (ground truth) with the predictions made by the model.

3 Methodology

3.1 Initial Approach: RoBERTa Large Model with AG News Dataset

Our first approach involved prompting the pre-trained `roberta-large-mnli` model, which we imported from the Hugging Face transformers library with the labels World, Sports, Business and Technology.



(a) Confusion Matrix

```
45.86666666666667
Candidate labels: ['World', 'Sports', 'Business', 'Sci/Tech']

Overall Accuracy: 45.87%

Per-class Performance:
              precision    recall  f1-score   support

    Business       0.46      0.14      0.22       702
     Sci/Tech      0.85      0.19      0.30       776
      Sports       0.77      0.60      0.68       775
       World       0.33      0.89      0.48       747

    accuracy                           0.46      3000
   macro avg       0.60      0.46      0.42      3000
weighted avg       0.61      0.46      0.42      3000

Most Confident Mistakes:
```

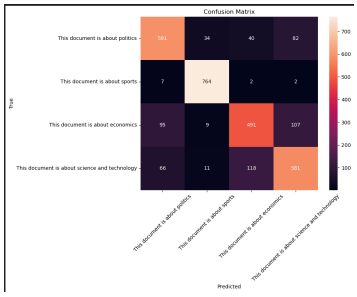
(b) Statistics

Figure 1: Default Prompts

The initial results, using the default labels corresponding to the AG News dataset, yielded the above performance scores. Although the model provided reasonable results, we aimed to improve the classification accuracy and overall performance by refining the prompts used during classification.

3.2 Manual Human Prompting

Building upon the initial results, we explored the potential of improving it's performance through manual human prompting. We hypothesized that by providing better-structured prompts, we could guide the model toward more accurate predictions. We experimented with several different prompt formulations, adjusting them to better align with the task and dataset.



(a) Confusion Matrix

```
80.9
Candidate labels: ['This document is about politics', 'This document is about sports', 'This document is about economics', 'This document is about science and technology']

Overall Accuracy: 80.98%

Per-class Performance:
                                precision    recall  f1-score   support

          This document is about economics       0.75      0.70      0.73       702
           This document is about politics       0.78      0.79      0.78       747
This document is about science and technology       0.75      0.75      0.75       776
            This document is about sports       0.93      0.99      0.96       775

                                  accuracy                           0.81      3000
                                 macro avg       0.80      0.81      0.81      3000
                              weighted avg       0.81      0.81      0.81      3000

Most Confident Mistakes:
```

(b) Statistics

Figure 2: Human Prompts

After a series of attempts, we identified a set of effective prompts that significantly improved the model's performance. The refined prompts yielded the above results. These improvements inspired us to extend our approach even further by utilizing advanced generative models to help enhance the prompts iteratively, which leads us to the next section.

3.3 Prompting RoBERTa using Generative Large Language Models

To refine our prompting process for RoBERTa, we conducted an analytical study using generative large language models (LLMs). These models were chosen for their diverse architectures, parameter scales, and training methodologies, each contributing unique strengths to our iterative prompt optimization process. Below, we provide detailed technical descriptions of the models:

3.3.1 Generative Models Overview

Gemma2 (9B parameters): Gemma2 is a lightweight generative language model from the Gemma family, designed for state-of-the-art performance in compact architectures (Google Team 2024[4]). It employs advanced transformer techniques such as interleaving local-global attentions (Beltagy et al. 2020[5]) and group-query attention (Ainslie et al. 2023[6]). Trained using knowledge distillation (Hinton et al. 2015[7]) instead of next-token prediction, Gemma2 delivers performance competitive with models 2–3× larger, making it highly efficient for generating prompt variations and enabling rapid experimentation.

Qwen2.5 (32B parameters): Qwen2.5, developed by Alibaba Cloud, is a 32-billion-parameter language model that advances the Qwen2 (Alibaba Team 2024[8]) series with significant architectural and training improvements. It utilizes an enhanced transformer architecture with multi-query attention (Shazeer et al. 2019[9]) for efficient decoding and rotary position embeddings (Su et al. 2023[10]) for better representation of sequential data. Training incorporated a mixture of objectives, including next-token prediction and instruction tuning, enabling superior generalization across tasks.

The model was trained on a diverse dataset, including high-quality curated texts from multiple domains, to ensure robustness and adaptability. Its large parameter size and optimized architecture make Qwen2.5 particularly effective for complex reasoning tasks and generating high-quality outputs.

Nemotron (70B parameters): Nemotron, developed by NVIDIA, is a 70-billion-parameter generative language model based on the LLaMA 3.1 framework (Meta Team 2024[11]). Building on the foundation of LLaMA 3.1's sparse and dense mixture of experts (MoE) architecture, NVIDIA introduced significant enhancements tailored for high-performance computing and inference efficiency. This model was trained using RLAIF (specifically, REINFORCE), Llama-3.1-Nemotron-70B-Reward and HelpSteer2-Preference prompts on a Llama-3.1-70B-Instruct model as the initial policy.

These improvements include the integration of fused multi-head attention and tensor parallelism optimized for NVIDIA GPUs, enabling faster training and inference speeds at scale. Additionally, Nemotron leverages a more sophisticated data sampling strategy, prioritizing domain-specific content during pretraining to improve generalization on specialized tasks. The model was fine-tuned with NVIDIA's proprietary instruction-tuning framework to excel in contextual understanding and complex generation tasks.

Quantization of Models: To ensure efficient deployment and reduce computational overhead, we used quantized versions of all three generative models for this study. Quantization is a process that involves converting a model's parameters from a high-precision format (e.g., 32-bit floating point) to a lower-precision format (e.g., 8-bit integers). This process reduces the model's memory footprint and speeds up inference without significantly affecting its performance. We utilized Ollama python library to import and use the quantized version of all of these models.

In technical terms, quantization exploits the redundancy in the representation of neural network weights and activations. By approximating these values with lower precision, we achieve a significant reduction in computational resources, which is especially beneficial for larger models like `Nemotron`. Quantization also allows these models to run on hardware with limited memory, such as edge devices or GPUs with constrained resources.

3.3.2 Prompts to LLMs

Initial Prompt: We used the same initial for all 3 models, this prompt was inspired by our understanding gained by the manual prompting we did earlier. Special emphasis was put on asking the model to write the label prompts as assertions rather than questions, as having prompts as questions lead to significantly worse results for classification. The initial prompt gave each LLM full statistical information about how the default label prompts performed along with further context of most confident mistakes and common error patterns in a hope for it to use that information to generate better label prompts.

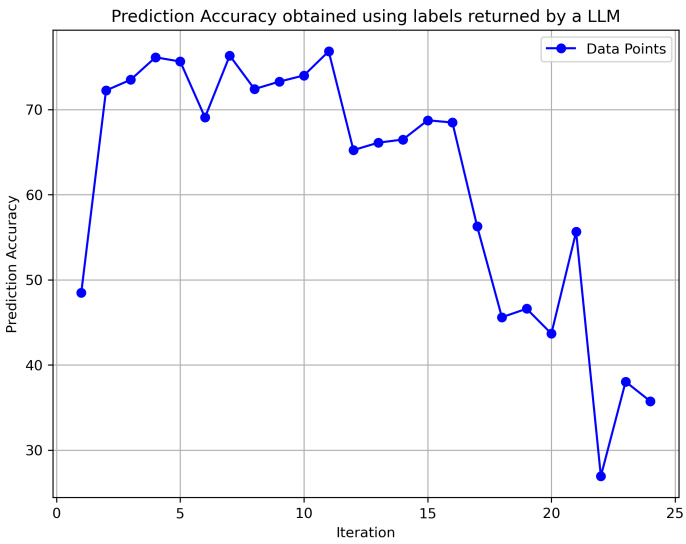
```
message = '''You are an expert prompt generator for a zero shot text classification task. I am using roberta-large-mnli to do zero shot classification on the ag news dataset.
I will first provide you with the truth labels corresponding to the 4 types of texts in the ag news dataset (in total the dataset has over one hundred thousand different texts).
The AG News dataset provides us with the following intial labels: 0: World, 1: Sports, 2: Business, 4: Sci/Tech
Then iteratively, I will provide you with the current prompts that have been given to the model and the corresponding accuracy of the model per label on the task.
I want you to provide better prompts for all the four labels so that the accuracy of the model improves. We will have many iterations of this process and I want you to iteratively give better prompts.
I will also give you the per class accuracy, the most confident mistakes, and common error patterns observed for the given labels. Infer whatever you want to from them and try to make your results better.(Follow the format exactly, dont miss any commas or inverted commas)
Your output should be of the format 0: \"prompt0\", 1: \"prompt1\", 2: \"prompt2\", 3: \"prompt3\" where the values of the dictionary correspond to the prompts that you generate. (The prompts aren't questions, they are assertions)
Do not generate any text other than these prompts in dictionary format.
Also if you are able to achieve precision of over 0.90 for any of the four labels then don't change the prompt for those at all.
This is my initial input to you, from the next message onwards, I will start giving the above mentioned data as the message, I won't provide any other information in the subsequent messages, so use the information I gave in this message to interpret the data provided.
Also, don't give any non alphanumeric characters in the output such as emojis or emoticons.
Overall Accuracy: 48.50%

Per-class Performance:
              precision    recall  f1-score   support

    Business       0.36      0.08      0.13       205
    Sci/Tech       0.91      0.21      0.34       253
      Sports       0.75      0.68      0.72       274
       World       0.35      0.85      0.50       268

    accuracy                           0.48      1000
   macro avg       0.60      0.46      0.42      1000
weighted avg       0.61      0.48      0.44      1000

Most Confident Mistakes:
                                                                                                                                   text
702                                          HP Unveils Cavalcade of Consumer Products (PC World) PC World - First TVs, new printers, long-lasting inks, and projectors are targeted\ at living room and office.
624  World #39;s smallest digital camera with zoom lens Come September, Japanese electronics giant Casio Computer will launch the world #39;s smallest digital camera with a zoom lens. Casio #39;s palm-sized Exilim camera is much smaller than others as, for the first time, it uses a ceramic lens.
468                                          Microsoft: Use Script to Block Windows XP SP2 Updates Microsoft has offered up yet another way for businesses to block the automatic update of Windows XP to the big-deal Service Pack 2 (SP2) upgrade.
821                                          We owe Athens an apology ATHENS -- The Games of the XXVIII Olympiad -- the great disaster that wasn #39;t -- come to an emotional end this afternoon and, really, the world owes Athens an apology.
963                 EU, Japan Win WTO Approval to Impose Duties on US (Update2) The European Union, Japan and Brazil won World Trade Organization backing to impose tariffs on US imports after Congress failed to end illegal corporate subsidies worth \$850 million since 2001.

    true_label predicted_label  confidence
702   Sci/Tech           World    0.952798
624   Sci/Tech           World    0.937181
468   Sci/Tech        Business    0.881427
821     Sports           World    0.867683
963   Business           World    0.826846


Common Error Patterns:
Business → World: 173
Sci/Tech → World: 160
Sports → World: 87
World → Sports: 36
Sci/Tech → Business: 28
Business → Sports: 14
Sci/Tech → Sports: 12
World → Sci/Tech: 3
Business → Sci/Tech: 2

(print only in the format described and print nothing else, don't forget the double quotations for the prompts)
    Output format: {0: "prompt0", 1: "prompt1", 2: "prompt2", 3: "prompt3"}

'''
```

Figure 3: Initial Prompt provided to LLMs

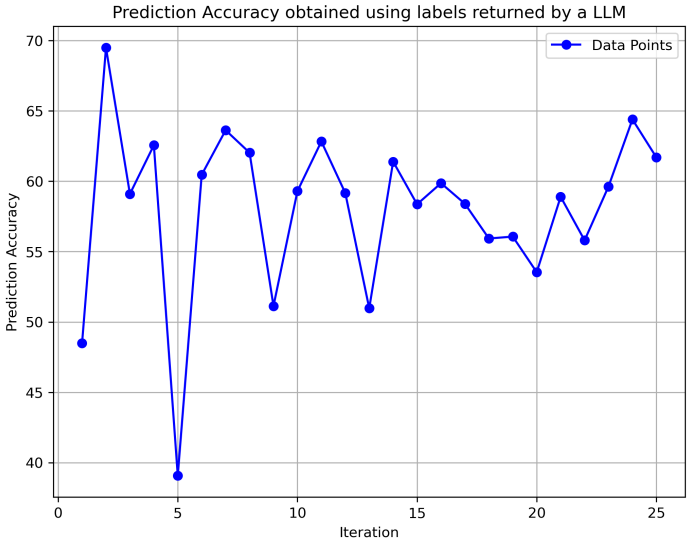
Subsequent Prompts: For this, we improved the prompts for different LLM models.

- **Gemma2**: The model received the recurring prompt, which contained the response error analysis based on the label prompts it provided in the last iteration. Unlike the initial prompt, this recurring prompt excluded the suggestion that label prompts should be assertions rather than questions.
- **Qwen-2.5**: This was different as it contained the suggestion to have assertions as label prompts repeatedly. Also, it reminded it of the original labels of the data so it didn't forget what the original labels were.
- **Nemotron**: This had a slight modification as it contained the suggestion that if it got a precision greater than 0.90 for any label, then to not change the label prompt of that label until all 4 labels got precision greater than 0.90.

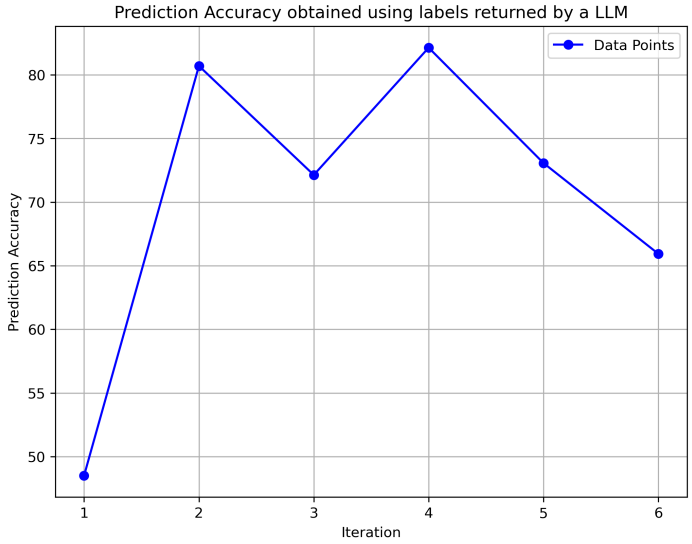
4 Results



(a) Gemma2



(b) Qwen2.5



(c) Nemotron

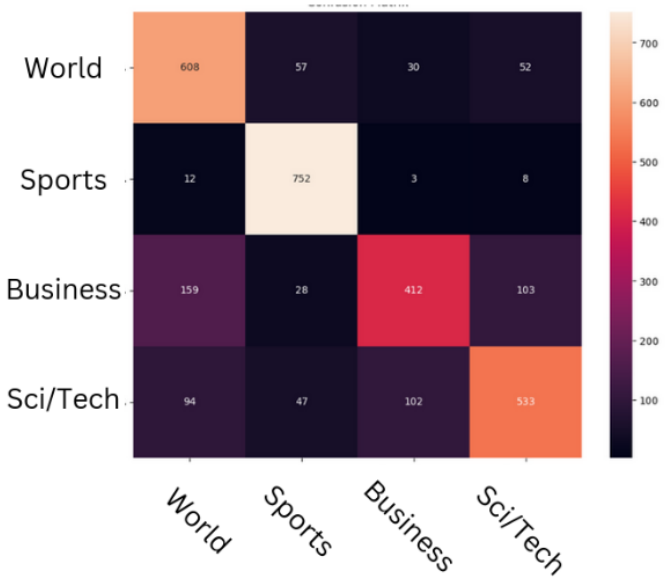
Figure 4: Accuracy trends of LLM Models

These plots reveal a few key aspects:

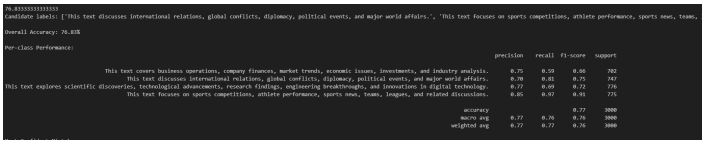
- **Memory/Context Length**: The plot for the run using Gemma2 as the label prompt generator suggests that the initial prompt message given to Gemma2 was pretty good(as can be seen by the immediate increase in accuracy). But after about 15 iterations it just collapses. We observed the label prompts it generated and found out that it started asking questions as labels, which we had observed earlier was a bad idea and due to this it collapsed on itself.
- **Reminders**: Qwen2.5's plot is very stable and mostly consistently gets an accuracy of above 60%. This was as expected because for Qwen we had put a recurring prompt of reminding it what the original labels were as well as to not put questions as labels, which it obeyed.
- **Nemotron**: This was the largest model of them all and got about the same prompt as Qwen except to keep good label prompts as they are. Due to shortage of resources we were only able to run 6 iterations of it, but on the 4th iteration itself it generated the best label prompt we found in our entire experiment.

Best Results obtained: We used the prompts given by each LLM that gave the highest overall accuracy and calculated the statistics and confusion matrices for those prompts

- Gemma2:



(a) Confusion Matrix

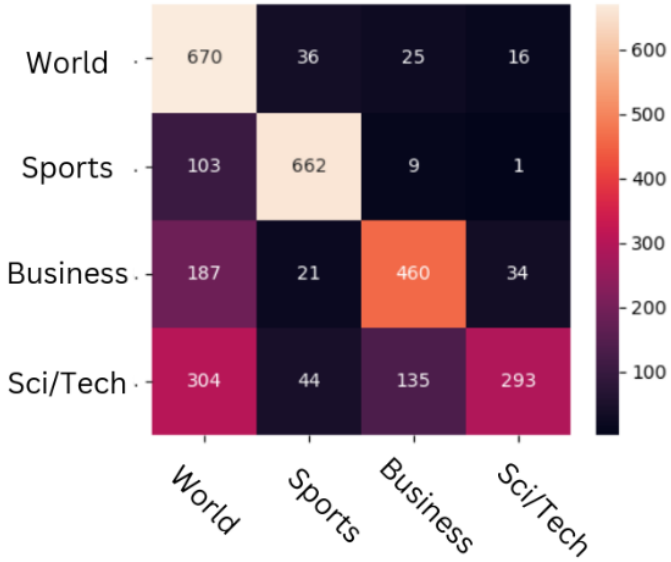


(b) Statistics

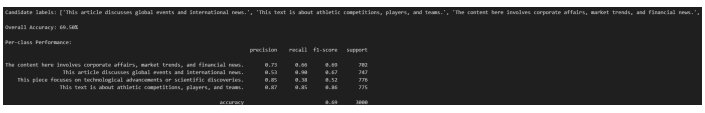
(c) Common Error Patterns

Figure 5: Gemma2's Best Run

- Qwen2.5



(a) Confusion Matrix

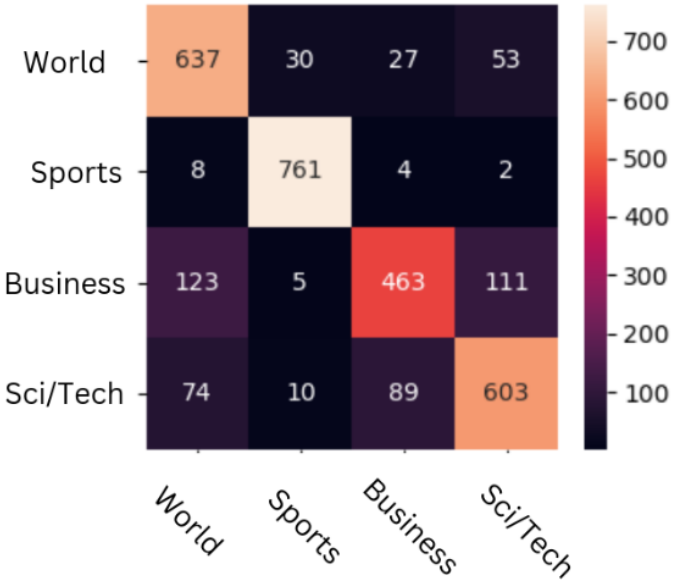


(b) Statistics

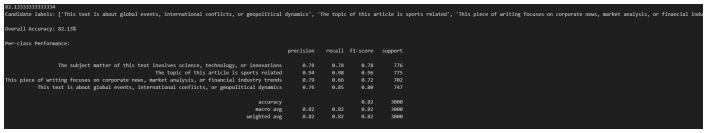
(c) Common Error Patterns

Figure 6: Qwen2.5's Best Run

- Nemotron:



(a) Confusion Matrix



(b) Statistics

(c) Common Error Patterns

Figure 7: Nemotron's Best Run

- **Precision:** It was observed that the model, when tested on news with the actual label as 'World,' had the worst precision for all 3 of these best prompts. This means that the classifier falsely classified many documents as 'World' news.
- **F1-score:** 'Business' news had the worst F1-score in 2 out of these 3 prompts, while 'World' news had the worst F1-score for one prompt. This suggests that the classifier struggled to balance precision and recall for 'Business' news, indicating potential confusion between 'Business' and other categories. Similarly, the poor F1-score for 'World' news in one prompt reflects difficulty in accurately capturing both true positives and minimizing false positives and false negatives.
- **Some Common Error Patterns:** 'Business' being classified as 'World' news was by far the most common error pattern which also contributed in 'World' news having a low precision score.
  Another common misclassification observed was 'Business' news being classified as 'Sci/Tech' news, these can be observed in the confusion matrix. Such common patterns are not suprising given the similarity in these.
- **Easiest class:** 'Sports' news was the easiest to classify class for most of these best prompts, this is understandable considering it's distinction from the other 3.

5 Conclusion

We have successfully showed that LLM chat-bots can be used to create accurate label prompts for zero shot classification to improve the ability of RoBERTa to classify texts on the AGNews dataset. Our study has showcased the increase in accuracy from 48.50% to 80.9% through manual prompting and to a best of 82.13% using LLMs as label prompt generators, which is a substantial increase. The best part about our method is that it is general, the algorithm wasn't fine-tuned for any dataset, we just changed the label prompts and got such accuracies.

Further work can be done by using these exact methods on different datasets and see if they retain such effectiveness. More tweaking can be done with the prompts to the label generator LLM which may lead to better label prompts for classification. We can fine-tune the model on news datasets to further increase efficiency of the classifier, as well as, fine-tune the label prompt generator LLMs on this kind of task, potentially using RLAIF or RLHF so that they may generate even better label prompts.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[4] G. Team, "Gemma 2: Improving open language models at a practical size," 2024. [Online]. Available: https://arxiv.org/abs/2408.00118

[5] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[6] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "Gqa: Training generalized multi-query transformer models from multi-head checkpoints," 2023. [Online]. Available: https://arxiv.org/abs/2305.13245

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[8] A. Y. et al., "Qwen2 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2407.10671

[9] N. Shazeer, "Fast transformer decoding: One write-head is all you need," 2019. [Online]. Available: https://arxiv.org/abs/1911.02150

[10] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," 2023. [Online]. Available: https://arxiv.org/abs/2104.09864

[11] A. D. et al., "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783