Semester Project for Advanced Topics in Machine Learning - Usecase 1

Few-shot or zero-shot learning paradigms using large language model

Aakash Khadka, Mohammadamin Madani, Karthikeyan Suresh, Amr Zayed, Max Neubaer

Motivation

As information grows, summaries are becoming the preferred way to consume text. With local LLMs like Gemma 3B, we can now generate summaries quickly but can these replace full documents for tasks like text classification?

This project compares three document representations:

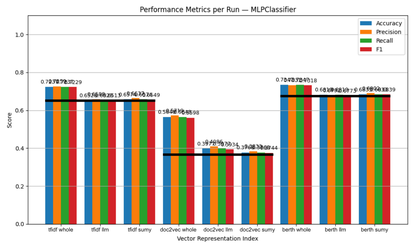
- TF-IDF and Doc2Vec from full text
- Gemma 3B summaries embedded via Sentence-BERT

We evaluate these on 20 Newsgroups and AG News datasets using SVM, MLP, and Random Forest classifiers. We also test how well LIME, SHAP, and the LLM itself explain predictions.

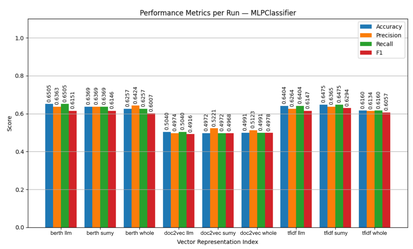
Our goal: assess the trade-offs between accuracy, efficiency, and explainability in summary-based vs. full-text pipelines.

Results

We split the 20 Newsgroups dataset into training and testing subsets. After training, we evaluated the performance of different classifiers and vectorizers. The results show that models perform best when trained and tested on the full dataset, rather than on its summarized versions.



When testing on a different dataset—the AG News dataset—we observed no significant performance difference between models using the full texts and those using summaries. However, overall performance was lower. This is expected, as the models were not trained on AG News and we had to map the 20 Newsgroup categories to the 4 classes used in AG News, introducing an additional layer of abstraction.



While testing XAI, we found LLM helped explain the model's decision really well in most of the cases.

Also, LLM seemed to be biased with the model's chosen category and created additional background to support it.

Some cases also helped us know that even when the model predicted class and LLM prediction were same, the important features had low impact, possibly due to weak feature extraction or poor data preprocessing.

```
2025-06-20 07:59:38,968 | INFO | ➡Ollama explanation with LIME values and its own classification decision:
2025-06-20 07:59:38,969 | INFO | **support**

The LIME explanation highlights key words ("bill," "clinton," "senanigan," "disclaim," "opinion," "express," "mine," "employ") which are strongly i
ndicative of a political discussion or commentary. The original LLM explanation states the text is about general conversation, which doesn't align
with the emphasis on political elements. The LIME explanation provides a contextualized understanding, thus supporting the classification of "talk.
politics.misc."
```

```
2025-06-20 08:01:24,235 | INFO | ➡Ollama explanation with LIME values and its own classification decision:
2025-06-20 08:01:24,236 | INFO | **contradict**

The LIME explanation suggests a focus on Christian theology and practices, while the original LLM emphasizes theological discussions and beliefs. T
he text's core is religious, but the explanation highlights a specific framework, which is not explicitly covered in the text. Therefore, the expla
nation contradicts the text's primary focus.
```

Process

Text preprocessing

Both 20 Newsgroups and AG News datasets were cleaned using two modes:

- Light cleaning: lowercasing, minimal formatting (for LLMs/BERT)
- Strict cleaning: stopword removal, stemming, and punctuation filtering (for TF-IDF/Doc2Vec)

Duplicates, empty entries, and short texts (<10 tokens) were filtered. Cleaned data was saved as CSV and split 80/20 for training and testing.

Summary generation

For the summarization, we decided to use Gemma3:1b as the LLM model and LSA as a NLP summarization algorithm. a problem we faced while generating LLM summarize is that it adds useless text such as "Here is the summarization" or "Let me know if you need anything else". To resolve this issue, we developed a function to clean the output from all the filler phrases. Another issue we face was the time per summarization especially for the LLM summarization. It would take ~2 seconds depending on the hardware resources used. With over 200,000 data entry, This would be a waste of time. Therefore we introduced multi-threading to parallelize the summarization.

Vectorization

We explored three vectorization techniques to represent text numerically: TF-IDF, Doc2Vec, and BERT. For Doc2Vec, we experimented with various hyperparameters (e.g., vector size, window, epochs) and selected the configuration that gave the best classification results. We used a pre-trained BERT-based Sentence Transformer (all-MiniLM-L6-v2) to generate vector representations of documents. This allowed us to compare how different text representations impact classification performance.

Training and evaluation

To obtain representative results, we used three different classifiers for training and evaluation: SVC, MLP, and Random Forest. As each classifier requires different hyperparameters, we performed grid search prior to training to identify the most effective configurations. Each classifier was trained on three versions of the 20 Newsgroups dataset: the full texts, LLM-generated summaries, and Sumy-generated summaries. Each version was vectorized using TF-IDF, Doc2Vec, and BERTopic, resulting in 9 distinct datasets. These were each used to train the three classifiers, producing a total of 27 models.

Although there are minor differences in performance metrics such as accuracy, precision, recall, and F1 score, overall trends are consistent across models. These are detailed in the Results section.

Additionally, the trained models were evaluated on the AG News dataset, which was preprocessed in the same way as the 20 Newsgroups data. To enable comparison, the 20 categories from the original dataset were mapped to the 4 categories used in AG News. The outcomes of this cross-dataset evaluation are also discussed in the Results section.

XAI

We selected the best performing model i.e MLP for the purpose of XAI analysis. SHAP was found to be infeasible for fast analysis due to its excessive processing requirements. Thus, LIME was only used for quantifying the model's performance. We used the gemma1:1b model to explain the model's prediction decision for the text and, supplied the llm's explanation with LIME values to check the agreement and disagreement.