Assignment 2

Centanni Samuele, Cotic Tomaz and **Lodi Mattia**
Master's Degree in Artificial Intelligence, University of Bologna
{ samuele.centanni, tomaz.cotic, mattia.lodi3 }@studio.unibo.it

Abstract

In this report we discuss and outline our work in completing the second assignment for the Natural Language Processing course. The main objective of the assignment is to use zero-shot and few-shot inference on a text classification task. We used two available large language models comparing their performance in both settings. We found out, empirically, that few-shot inference improves the performance of the system and the careful choice of the demonstrations to include pays a big role in the performance increase.

1 Introduction

The task at hand is to classify a given text into one of five classes: *not-sexist*, *animosity*, *prejudiced*, *threats*, and *derogation*. This classification is approached by prompting large language models in both zero-shot and few-shot settings.

We compare two instruction-tuned large language models available from HuggingFace: Mistral-7B-v0.3 and Llama-3.1-8B. Both models were evaluated under three experimental settings:

1. **Zero-shot inference.**
2. **Few-shot inference with random demonstrations.** Demonstrations are sampled randomly from the dataset and statically inserted into the prompt, i.e., the same demonstrations are used for all examples.
3. **Few-shot inference with similarity-based demonstrations.** For each text to classify, the N most similar demonstrations for each class are selected from the dataset using cosine similarity in the embedding space.

In few-shot experiments, the number of examples per class, N, was varied as $N \in \{2, 4, 6\}$.

The main findings of this study are:

- Adding demonstrations to the prompt improves the performance of both models.
- Selecting demonstrations based on similarity further enhances results in all cases.
- No consistent pattern emerges regarding the optimal number of demonstrations per class.

2 System description

As introduced earlier, we define three distinct inference systems, which are described in detail below:

1. **Zero-shot inference:** The text passages to be classified are directly inserted into the provided prompt, which is then fed to the large language models. The models response is then extracted with a custom function that doesn't perform any particular cleaning.
2. **Few-shot inference (A):** A function is first defined to randomly sample num_per_class examples from the demonstrations dataset for each class. These examples are then statically inserted into the prompt, independent of the text passage to be classified. Finally, the text field of the prompt is filled with the passage to classify, and the large language models answer is extracted as in the zero-shot setup.
3. **Few-shot inference (B):** A function is defined that selects demonstrations tailored to each text passage. Using the BGE-large-en-v1.5 sentence encoder, all demonstration examples are embedded in an off-line manner for efficiency. For each text passage once it is embedded, the num_per_class most similar examples per class are selected based on cosine similarity between the text embedding and the precomputed demonstrations' embeddings. The selected examples are added to the

prompt along with the text to classify. The resulting prompts are then fed to the LLMs to obtain their categorical answers.

3 Experimental setup and results

We conducted inference using two instruction-tuned large language models: Mistral-7B-v0.3 and Llama-3.1-8B. Both models were evaluated on two few-shot inference tasks, with the hyperparameter num_per_class set to $2, 4, 6$. To compute the similarity between examples and target texts, we employed the BGE-large-en-v1.5 sentence encoder.

All experiments were evaluated using the macro F1-score and the fail ratio. The results are summarized in Table 1. The fail ratio is the ratio between invalid and all responses, where an invalid answer is an answer without class, with two or more classes, or with an explanation. The prompt states we only want the category, so, providing the explanation means not following instructions.

Mode	Shots (per class)	Mistral F1	Mistral Fail	Llama F1	Llama Fail
zero-shot	0	0.3738	0.0066	0.4738	0.0633
random	2	0.4255	0.0633	0.4223	0
random	4	0.4970	0.0433	0.4010	0
random	6	0.5124	0.0266	0.3794	0
similarity	2	0.5068	0.0266	**0.5408**	0
similarity	4	**0.5511**	0.0066	0.5304	0
similarity	6	0.5452	0	0.5123	0

Table 1: Performance comparison across prompting modes and models. Best values for F1 are highlighted in bold.

4 Discussion

From the results shown in Table 1, we observe that the Mistral model benefits from the inclusion of demonstrations, even when they are selected randomly. In contrast, the Llama models performance can degrade if demonstrations are not chosen carefully. For both models, using a similarity-based selection strategy leads to a significant improvement in performance. This outcome is expected, as the model is provided with examples that are most similar to the target text. Upon inspecting several cases, we found that selecting the most similar demonstrations generally preserves the semantic meaning of the query (body weight, shooting, killing), although it may alter the tone of the sentence.

Moreover, although the Llama model performs better in the zero-shot setting, the Mistral model slightly outperforms it in similarity-based few-shot inference.

We also observe that adding more demonstrations does not necessarily lead to monotonic improvements in performance. In fact, the Llama model achieves its highest F1-score with $num\_per\_class = 2$, while the Mistral model reaches its peak at $num\_per\_class = 4$. This suggests that including too many demonstrations may introduce confusion, corrupting performance.

We have also looked at the best performing configurations for each model and inspected the per-class improvements from zero-shot inference. The classes that benefit the most from including demonstrations are threats and derogation.

An additional consideration can be done on the answers the models provide. Mistral in some cases doesn't output exactly a single category, but adds some additional reasoning or other classes resulting in invalid answers. Llama, in zero-shot only, answers (18 times) by letting the user know it can't annotate sexist content.

Finally, examining the confusion matrices across the different models and configurations, we observe that the *animosity* class consistently exhibits the lowest precision, ranging from 0.2 to 0.4. This indicates that the models frequently misclassify examples from other classes as *animosity*.

In contrast, the *threats* class achieves the highest precision, between 0.7 and 0.9, in the majority of experiments, suggesting that the models are more reliable at correctly identifying examples of this class. Slightly worse, but still better than other classes, is the not-sexist class, ranging from 0.47 to 0.82.

5 Conclusion

We investigated zero-shot and few-shot prompting with Mistral-7B-v0.3 and Llama-3.1-8B for a five-class text classification task. Our experiments show that few-shot inference improves performance over zero-shot, and that selecting demonstrations based on similarity further boosts results. The optimal number of demonstrations per class is not consistent, and too many examples can sometimes reduce performance.

Per-class analysis indicates that *threats* and *derogation* benefit most from demonstrations, while *animosity* remains challenging.

These findings underline the importance of demonstration inclusion and careful demonstration selection.