RAGE - Retrieval Augmented Generation and Election Algorithm for Few-shot Intent Classification

Soumadeep Saha

Indian Statistical Institute, Kolkata
soumadeep.saha97@gmail.com

Abstract

State of the art Large Language Models (LLMs) show immense promise in several language understanding tasks, however due to the unconstrained nature of generated text, application to classification tasks is not straightforward. The standard approach where LLM features would be further mapped into the set of labels with the aid of additional layers is computationally expensive and does not work well when the number of training samples is limited. To address this we propose the RAGE (Retrieval Augmented Generation and Election) algorithm that first clusters semantically close training examples and uses them to create a retrieval augmented generation prompt, and then uses consistency conditions and voting to arrive at the final answer. Our approach achieves top accuracy scores of 91.47% with just prompt tuning and **90.8% accuracy with no gradient updates**. This approach is computationally cheaper than gradient based methods, and naturally extends to ensembling on a family of models. Additionally, our approach also efficiently solves the unconstrained text to label mapping issue for the intent classification task and successfully leverages the vast language understanding capabilities of state of the art LLMs.

Method Description

Although LLMs excel at language understanding, the main challenge of using LLMs in the intent classification problem stems from the unconstrained nature of outputs they produce. Standard approaches of dealing with classification tasks such as this would involve mapping the output embeddings from a suitable LLM to the set of possible labels with additional parameter layers followed by fine-tuning. However, since the available number of labeled instances is rather limited (2248) and typical models have around 10^{10} trainable parameters, this approach runs the risk of over-fitting. The labels are extremely close semantically, and it would be unreasonable to expect good decision boundaries with only ~15 example instances on average supporting each label.

Few-shot prompting (Brown et al. 2020) has been demonstrated to perform close to state-of-the-art fine-tuned models in a plethora of tasks, and in particular performance improvements have been reported with increasing the number of provided examples. Although the mechanism by which few shot prompting leads to performance gains is not well understood and continues to be an active area of research, there are several promising theories. In the bayesian view of in-context learning (ICL) (Xie et al. 2022) the prompt provides evidence for the model to sharpen the posterior distribution over concepts, p(concept|prompt) and if p(concept|prompt) is concentrated on the relevant concept, the model has effectively "learned" the concept from the prompt. Thus we have -

$$P(output|prompt) = \int_{concept} p(output|concept, prompt)p(concept|prompt)d(concept)$$

So, if prompts are chosen such that p(concept|prompt) can be increased for the relevant concept p(output|prompt) should in principle also improve for the relevant output. To achieve this we employed a retrieval-augmented generation (RAG) strategy.

We employed a pre-trained Roberta-large (Liu et al. 2019) model[1] fine-tuned on the Amazon massive dataset (FitzGerald et al. 2022) to extract features for the provided "surprise" (training) and test datasets. These features were then used to find candidate "best-matches" from the training set using k-nearest neighbors. These k training examples alongside l randomly sampled entries from the surprise dataset were used to create RAG (k+l) shot prompts. The k-NN based approach with plurality voting already performs appreciably, and we observed an **accuracy of 84.15% with k=15**. This leads us to believe that this strategy can cluster based on semantic similarity, and can lead to improvement in the p(concept|prompt) term for the relevant concept in ICL.
These prompts[2] (see Table 1) were then fed into the LLM to generate several candidate completions of limited

[1] philschmid/habana-xlm-r-large-amazon-massive on hugging-face.

[2] The preamble can contain instructions or an introductory message, however since dropping it didn't result in performance degradation, we elected not to use a preamble. This is consistent with the literature where a large number of few shot prompts were shown to supersede the need for textual prompts.

sequence lengths by random sampling from the distribution. Although several LLMs were tried (Mistral, phi-1.5, etc.) best results were observed with the Llama-2 family of models (Touvron et al. 2023), and in particular the 13B and 70B models were used. It was observed that the LLMs always produced outputs of a certain format (see Table 1) where an intent description would be generated followed by a new line character and more LLM generated utterances (we truncated generation at 10 new tokens). This behavior is typical of LLMs and can be used to our advantage. We simply consider the generated text before the newline (\n) character to be the generated response.

Preamble (instructions)
Utterance : < training set utterance 1> *Intent : < utterance 1 label >* *Utterance : <training set utterance 2>* *Intent : <utterance 2 label>* . . . *Utterance : <training set utterance k+l>* *Intent : <utterance k+l label>*
Utterance : <test set utterance> **Intent : <eos>**
<LLM generated text>\nUtterance : <further LLM text>

Table 1 : Few-shot prompt formats used for retrieval-augmented generation and corresponding LLM outputs. k best matches from k-NN with l randomly sampled examples from the training set were used as few-shot prompts.

This still leaves one problem for us to solve - there is no guarantee that the LLM generated response is a valid intent label. We used two approaches to solve this issue motivated by observed behaviors. The first approach was motivated by the observation that sometimes the LLMs produced outputs that were subsets or supersets of actual label texts. For example a valid label is “rollover 401k” whereas we found that the LLMs labeled them as “401k” or “next song” as “play next song”. To exploit this phenomena we computed the F1-score of generated LLM output with every candidate label and picked the highest scoring candidate. However, this approach showed worse performance when compared to the other approach, and wasn’t investigated further.

The other, more successful approach is inspired from Chain-of-thought with Self-consistency (CoT-SC) (Wang et al. 2022) idea - where we procure several generations from an LLM with CoT prompts and select based on majority voting. Our implementation uses several samples from an LLM and adds to it the k-NN solution, and reduces the set based on whether the label exists in the set of potential labels followed by majority voting. This algorithm, dubbed **RAGE(Retrieval Augmented Generation and Election),** guarantees that the set of candidate labels always contains one label that is acceptable, so even after reduction, at least one label remains. So in the worst case the k-NN solution is chosen (which is already appreciable) and in the best case scenario we choose a label that an LLM predicted several times potentially avoiding noisy outputs.

This approach lends itself extremely well to ensembling. In practice we can choose n_1, n_2, … n_r samples from LLM_1, LLM_2, … LLM_r respectively, resulting in the final candidate set of size ($1+n_1+n_2$ + … n_r) followed by reduction which is guaranteed to contain at least one acceptable label. This is followed by choosing the most frequently occurring label from the reduced set. In our final ensemble we used Llama2-13B with n=3, Llama2-13B (prompt-tuned) with n=4 and Llama2-70B with n=6 for our final submission achieving an **accuracy of 91.47%** - a 7.32% improvement over the base kNN approach.

There are two competing factors at play with this approach, as on one hand we want all generated outputs to fall within the set of acceptable labels and on the other hand diversity of answers is desirable so that consensus voting is meaningful. Generation parameters like temperature and top-p (nucleus sampling) were adjusted to achieve these goals. We also attempted to use Q-LoRA (Dettmers et al. 2023) fine-tuning on the Llama-13B model, however this approach didn’t lead to much success and was dropped from the final submission. The full schematic algorithm can be found in Algorithm 1 and is illustrated in Figure 1.

Algorithm 1 : **RAGE**

```
Given LLM1, LLM2, … and n1, n2, …, l, k;
answers = [];
for x in test set do:
    get k best matching examples with kNN;
    candidate_labels = \\
                [most_frequent_knn_label];
    for each LLMj do (nj times):
        Sample l random training examples;
        Create a prompt with k best matching \\
           and l random training example pairs;
        y = LLMj(prompt, x);
        if y in acceptable_labels:
            candidate_labels.append(y);
    z= most frequent label in candidate_labels;
    answers.append(z);
return answers;
```

k,l were chosen such that the created prompt always fits in the context length. For the final submission we used k=60 and l=10. The examples in the few shot prompts were arranged on the basis of relevance to the test instance i.e.

the random examples were placed first followed by the kNN examples in order of closeness.[3]

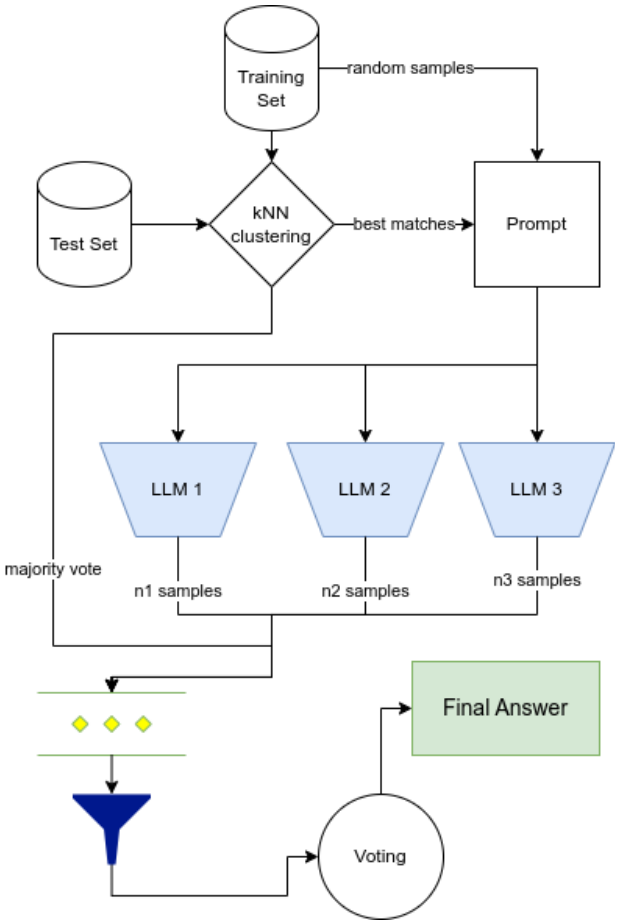

Figure 1 : Schematic diagram of the RAGE algorithm.

Experimental Results

Since the success of our method is contingent on LLMs being constrained to the set of acceptable labels. That is the first experiment we performed. In particular we looked at the variation in the percentage of acceptable labels generated with the number of examples provided in the few shot prompts. We found that with an increasing number of few shot examples the percentage of acceptable labels generated increased from **69.088% with 5 shot** to **80.57% with 60 shot examples** (see Figure 2). This experiment was performed using Llama2-13B (temperature = 0.5, n = 5 samples)[4]. Even better results were obtained with Llama2-70B **(84.2%)**, however due to the high cost of inference a comparative analysis with varied numbers of prompts was not performed.

Several other qualitative experiments were performed like performance with textual prompts, Q-LoRA, prompt tuning, varying the number of few shot examples and by varying the contribution of each model. These results are summarized in Table 2.

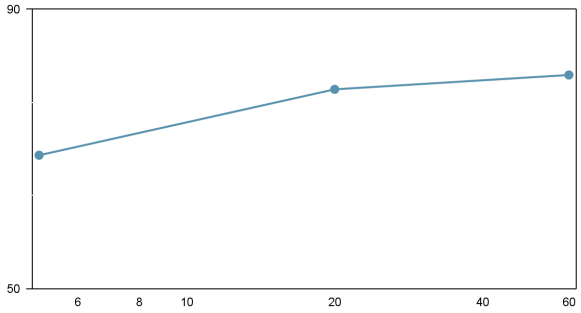

Figure 2 : Variation of percentage of acceptable labels with number of few-shot examples.

Method	Accuracy (%)
15-NN	84.15
+ Llama2-13B (50 shot)	88.18
15-NN + Llama2-13B + (textual & 50 shot)	86.83
15NN + Llama2-13B (60 shot)	89.40
+ Llama2-70B (60 shot)[5]	**90.81**
+ Llama2-13B with prompt tuning[6]	**91.47**
/ Llama2-13B Q-LoRA	91.08

Table 2 : Summary of results from various models employed.

Other models we considered were phi-1.5 (Li et al. 2023), Mistral-7B (Jiang A. Q. et al. 2023), etc. however these models produced a very low percentage of acceptable results with few-shot prompting, probably due to the small model sizes as demonstrated in the literature (Brown et al. 2020). Our experiments were carried out on two servers, the low parameter models were run on a single A6000 GPU and the 70B model was run on a server with two A100 80GB GPUs.

[3] Further details can be found in the github repository https://github.com/espressoVi/RAGE-LLM-IntentPrediction

[4] The number of random examples used in the prompt (l) = 1

[5] The 70B model got 5 votes and the 13B got 3.

[6] With 3, 6 and 4 votes each respectively.

Novelty

The main novelty of our approach lies in being able to successfully leverage the extraordinary language understanding capabilities of LLMs in this context. Mapping the free-form generation from language models into the limited set of labels poses a challenge, and straightforward approaches to map LLM outputs to the label space proved to be ineffective. Our consistency based approach suppresses noise and the RAG strategy improves LLMs ability to focus on relevant concepts. Further, our prompting technique increases the percentage of generations which are in the set of acceptable labels and our aggregation method ensures that even in the worst case scenario an acceptable is present in the set of candidate labels. Our method produces **extremely high accuracy scores of around 90.8% with absolutely no gradient updates** and generalizes well from the extremely limited training set examples.

Further, this approach has the added benefit of pointing out controversial test samples (see Table 3). If such a model were deployed in practice a human in the loop system could be devised, where if significant disagreement is present between the various models we can query an oracle for better intent elucidation.

Example	**Predictions**
Can you tell me what time the movie starts tomorrow? And what date is it playing	[date, time]
I'd appreciate it if you could send me a reminder for my upcoming PIN change deadline. Thank you	[pin change, reminder]
I need a reminder to schedule an appointment with my doctor.	[reminder, schedule meeting]

Table 3 : Examples from the test set with high variability of outputs produced by LLMs. LLMs are most confused when the choices between the labels are somewhat subjective.

There are two models used that have a big impact on the performance - the feature extraction model used for RAG and the LLM. We experimented with several feature extraction models (cartesinus/xlm-r-base) fine-tuned on Amazon massive and on other sentence classification tasks. The chosen model performs better than all other models we tested based on k-NN scores, which massively aids RAG.

The choice of LLMs were also thoroughly experimented with to make an optimal choice. We tried Mistral 7B, phi-1.5, Llama-13B, Llama2-13B-chat, Llama2-70B and Llama2-70B-chat and the instruction fine-tuned models were found to perform worse than the base model, because they tend to generate natural conversational texts which is contrary to expectations for this task. Computational limitations didn't allow us to experiment with larger models like Gopher or PaLM.

Conclusion

Although recent advances in large language models have revolutionized several disciplines their application to domain specific constrained classification tasks remains a challenge. Standard approaches are computationally expensive and do not perform well in low resource scenarios. To address this issue, we presented the RAGE (Retrieval Augmented Generation and Election) algorithm that achieves state of the art results on the intent classification task with only prompt tuning and shows very little performance degradation even without any gradient updates. Our method is computationally cheaper than fine-tuning and adept at generalizing from only ~15 examples per label. Further, it was observed that high variability in the intent labels produced by the models indicates genuine confusion which often cannot be parsed by human evaluators. This indicates that our method, when deployed in practice, can in principle use human feedback or query users for better intent elucidation.

References

Brown, T. B. et al. 2020. Language Models are Few-shot Learners. *Arxiv Preprint*. DOI: 10.48550/arXiv.2005.14165.

Dettmers, T. et al. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *Arxiv Preprint.* DOI: 10.48550/arXiv.2305.14314

FitzGerald, J. et al. 2022. MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages. *Arxiv Preprint.* DOI: 10.48550/arXiv.2204.08582

Jiang, A. Q. et al. 2023. Mistral 7B. *Arxiv Preprint.* DOI: 10.48550/arXiv.2310.06825

Li, Y. et al. 2023. Textbooks Are All You Need II: phi-1.5 technical report. *Arxiv Preprint.* DOI: 10.48550/arXiv.2305.14314

Liu, Y. et al. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Arxiv Preprint.* DOI: 10.48550/arXiv.1907.11692

Touvron, H. et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *Arxiv Preprint.* DOI: 10.48550/arXiv.2307.09288

Wang, X. et al. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *Arxiv Preprint.* DOI: 10.48550/arXiv.2203.11171

Xie S. M.; Raghunathan, A; Liang, P; and MaEngelmore, T. 2022. *International Conference on Learning Representations.*