Cross-Lingual Retrieval-Augmented In-Context Learning: Can Smaller Models Compete?

Rodrigo Nieto
Computer Science
Stanford University
rjnieto@stanford.edu

Abstract

Retrieval-augmented in-context learning has become a heavily researched field in order to solve knowledge-intensive tasks such as question answering. There has been existing work that has optimized interactions between frozen language models and retrieval models to approach open-domain question-answering tasks. However, few have experimented with smaller models and have tested the multilingual capabilities of their systems. In this paper, we modify the DEMONSTRATE-SEARCH-PREDICT (DSP) framework for the Alpaca model in order to produce results that are competitive to SoTA systems on English evaluations and are particularly impressive in Spanish and German. Our results deliver 137% relative gains on Spanish and 50% relative gains on German against a standard GPT-3 DSP baseline system. These results are remarkable for the Alpaca model and demonstrate how the small model could be leveraged for cross-lingual retrieval-augmented in-context learning tasks.

1 Introduction

Since the emergence of foundation models, research has only begun to understand the impressive zero-shot and few-shot capabilities that these models have on numerous downstream tasks. More specifically, advanced functionalities such as in-context learning have been researched with these models to solve knowledge-intensive problems. For instance, open-domain question answering has been a long-standing challenge because they require an immense amount of general world knowledge. This has led to the investigation of retrieval-augmented in-context learning which has seen rapid progress in very recent years. It has now become standard practice to pair a language model (LM) with a retrieval model (RM) to produce the most accurate, well-informed response. In this interaction, the RM will retrieve documents that are ideally relevant given the search query, while the LM will handle the output generation. Furthermore, new sophisticated interactions between the RM and the LM continuously getting discovered that can further performance gains (Liang et al., 2023).

Most modern LLMs' pre-training corpus is comprised of English text, with little influence from other languages. However, these models have been able to show some multilingual understanding (even in a zero-shot setting), despite the overall little understanding of their multilingual abilities and anecdotal evidence that their outputs are not as robust as their English outputs (Armengol-Estapé et al., 2021). Some research hypothesizes these potential capabilities may come from the model learning abstractions in their pre-training that generalize across multiple languages (Artetxe et al., 2020).

Therefore, we would like to test the efficacy of LLMs in the setting of cross-lingual retrieval-augmented in-context learning for open-domain question answering. In this paper, we will specifically be testing English, Spanish, and German, and show the potential failures on non-Latin script languages. Although, to our knowledge, this specific experiment (GPT *cross-lingual* in-context learning evaluation for question answering) has not been tested extensively, we would primarily like to perform this experiment on Alpaca 7B. Given that Alpaca is only a 7B parameter model compared to text-davinci-003's 175B parameters, it is very surprising that it can produce similar behaviors to GPT-3 despite being significantly cheaper to run (Taori et al., 2023). However, we still expect the results of Alpaca to be significantly lower than those of GPT-3 and therefore would like to find ways to optimize an Alpaca system with the usage of the DEMONSTRATE-SEARCH-PREDICT (DSP) framework (Liang et al., 2023). We will explore multiple technique variations that optimize performance for Alpaca to see if they can produce competitive results relative to GPT-3.

2 Prior Literature

2.1 In-Context Few-shot Cross-lingual Capabilities of LLMs

Although there is little research in the domain of cross-lingual retrieval-augmented in-context for open-domain question answering, the XRICL framework proposed by (Shi et al., 2022) was developed in order to solve cross-lingual retrieval-augmented in-context for cross-lingual Text-to-SQL semantic parsing. Text-to-SQL deals with converting natural language text to SQL code that could be executed by the computer, which involves semantic parsing, syntactic analysis, and logical reasoning. Therefore, their framework entails retrieving English references and global translation references for the target language in order to use chain-of-thought prompting for the inference stage. Their experiments show how they leverage Codex in their framework to retrieve and re-rank framework to display state-of-the-art performances on this task over a few datasets including two benchmarks that they developed which include questions in Chinese, Vietnamese, Farsi, and Hindi.

2.2 LLMs as Few-shot Multilingual Learners

In the paper authored by (Indra Winata et al., 2021), they test the multilingual capabilities of pre-trained models such as numerous GPT versions and T5 through few-shot in-context learning. The authors test these models only on languages with Latin-based alphabets: English, Spanish, French, and German. They test their model on 3 different datasets: SNIPS (evaluates spoken language understanding), MTOP (evaluates multilingual semantic parsing), and MultiNLU (evaluates textual entailment information). Their work showcases how the model performance in other languages dramatically increases as the models are given more samples. Furthermore, although the authors acknowledge the large performance gap between GPT and T5 models, they note that large gains can be made in the T5 models depending on the order of the samples that are shown to the T5 model, which is interesting information to consider when working with smaller models compared to GPTs. Finally, another informative note is that they recognize that they were not able to perform sophisticated work with some of the other language offerings in their tested datasets such as Chinese, Japanese, Hindi, and Turkish because their English-based tokenizers tend to fail on non-Latin characters (Indra Winata et al., 2021).

2.3 SoTA Retrieval-Augmented In-Context Learning Frameworks

(Liang et al., 2023) recently demonstrated how more sophisticated interactions between the LM and RM can lead to state-of-the-art results on open-domain question answering through retrieval-augmented in-context learning. In this paper, they introduce the DEMONSTRATE-SEARCH-PREDICT (DSP) framework, which is a multi-stage framework. Starting with the *demonstration stage*, a demonstration is a training example that was sampled to guide the LM to follow the desired behavior that the user is seeking. The *search stage* is meant to find passages of text in order to support the LM to produce factual responses, update world knowledge, and retrieve support knowledge for the query. Finally, the *predict stage* then uses the information gathered from the previous stages in order to generate an output given the original query. This framework steers away from the traditional retrieve-then-read or self-ask pipeline and consequently produces state-of-the-art in-context learning results on English-based question and answering datasets such as Open-SQuAD, HotPotQA, and QReCC.

3 Data

3.1 XQuAD

For our primary evaluation, we will be using XQuAD, which is a cross-lingual classification benchmark for question answering. The dataset is comprised of 240 paragraphs and 1190 question-answer pairs from the development SQuAD v1.1 translated into 10 languages that were translated by professional translators (Artetxe et al., 2020). Although the dataset offers 10 different languages to evaluate the efficacy of our model, we will primarily concentrate on 3 languages to narrow the scope of our analysis. The dataset's context passage and questions were translated by professional human translators from Gengo. The creators of the dataset showed in (Artetxe et al., 2020) that some promising results and intuition can be gathered from this dataset using mBERT.

3.2 SQuAD Train Split Translations

For the retrieval model, we will be retrieving demo passages from SQuAD's train split translations in each respective language. Because the XQuAD translations are from the SQuAD v1.1 validation set, we will ensure that the SQuAD translations used for the demos come from only the train splits. We will be using the following Spanish translation of the SQuAD train split that is available on Hugginface: https://huggingface.co/datasets/squad_es. It should be noted that this dataset is an automatic translation of SQuAD v1.1 and was not done by human professionals. Although it would be preferred to use translations done by humans, because I am a native speaker in Spanish I was able to sample 100 examples from the dataset to ensure that the translations were coherent and correct. For the German SQuAD train split translation, we will be using the following dataset that is available on Huggingface: https://huggingface.co/datasets/deepset/germanquad. This dataset combines the work of SQuAD's train split with some influence of the Natural Question dataset for open-domain QA. Their 13,722 questions were labeled by human annotators that were familiar with supervised learning, given a 2-hour workshop beforehand, and given rigorous labeling instructions (Möller et al., 2021). They also used computational linguistics and computer science students as well as crowd-sourced annotators for a three-way annotated dataset in order to have a more diversely annotated dataset (Möller et al., 2021).

4 Model

In this section, we will highlight the main framework components that our system will use.

4.1 DSP Framework

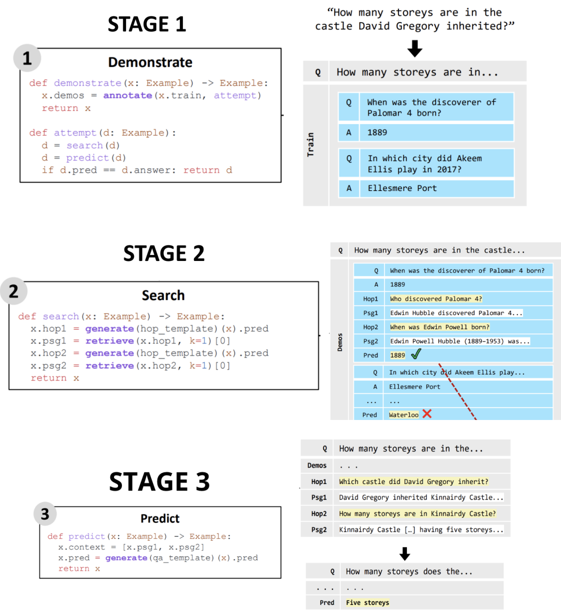
For our experiments and our baseline, we will be utilizing the DEMONSTRATE-SEARCH-PREDICT (DSP) framework as mentioned in Section 2.3. Considering that DSP is one of the SoTA frameworks for open-domain QA, and their system allows for flexible technique additions as well as LM configuration with LM abstract class, we decided it was an effective standard to use throughout our project. We can see in Figure 1 an example of a DSP program for multi-hop question answering that visualizes our previous explanations of the DSP framework.

4.2 Retrieval Model

For all model experiments, we will keep the retrieval model the same in order to emphasize the differences in the model performance. More specifically, we will use the ColBERTv2 model which is a state-of-the-art retriever that boasts zero-shot search quality as well as efficient search (Santhanam et al., 2022). ColBERTv2 is a late interaction model that produces multi-vector representations at the token level and decomposes relevance modeling into scalable computations for each and every token (Santhanam et al., 2022).

ColBERTv2 combines an aggressive residual compression mechanism with a denoised supervision strategy in order to both improve the quality of the retrieval and reduce the space footprint of the model (Santhanam et al., 2022). The authors of the paper have tested the model on numerous benchmarks and subtasks – such as BEIR Search Tasks, BEIR Semantic Relatedness Tasks, OOD Wikipedia Open QA, etc – and saw extremely promising and state-of-the-art performances in numerous sections.

Figure 1: An example of a vanilla DSP program for multi-hop question answering from (Liang et al., 2023). The demonstrate stage receives an input question and annotates the intermediate transformations of the 2 training examples. The search stage then receives supporting information over two retrieval hops. Finally, the predict stage uses the demonstrations and the supporting information to answer the input question.



4.3 Alpaca 7B

Alpaca 7B is an instruction-following model that was fine-tuned from Meta's LLaMA 7B parameter model (Taori et al., 2023). The creators of Alpaca trained it on 52K instruction-following demonstrations that were generated in the style of self-instruct through text-davinci-003. More specifically, they fine-tuned the LLaMA models using HuggingFace's training framework (Taori et al., 2023). The result of Alpaca has it display similar behaviors to GPT-3, despite it being a significantly smaller model (175B parameters to 7B parameters) (Taori et al., 2023). Despite this, Alpaca is still very susceptible to hallucinations and the spreading of false information, which is also why testing the model on open domain question answering makes it an interesting task to test the model on.

Therefore, we plan to also use a similar DSP framework with a few shot openqa system that includes context and demo filtering as the basis of the model. We will keep ColBERTv2 as the retrieval model and replace the vanilla GPT-3 model as the LM for Alpaca.

5 Methods

In this section, we will describe the techniques that we will be testing throughout our experiments, as well as the metrics for evaluation and the baseline model.

5.1 Metrics

5.1.1 Modified F1 Score

As stated in (Liang et al., 2023), their version of F1 score is computed by measuring the overlap between the system response and the ground truth while discounting common stopwords and terms present in the question. However, we noticed that this may raise difficulties in other languages, where discounting stopwords or terms not present in the question may not translate over as well when evaluating different languages. Therefore, instead of this, we decided to measure the semantic similarity between the predicted answer and the gold answer to obtain the number of true positives to calculate the precision, recall, and finally the F1 score.

In order to calculate the semantic similarity, we use the FuzzyWuzzy package. This package provides string matching options to the user and uses the Levenshtein Distance algorithm to calculate how similar two strings are. Furthermore, the benefit of using the FuzzyWuzzy package is that it does not have built-in language-specific features; therefore, this package can be used quite effectively with every language. FuzzyWuzzy's performance was quite robust from a qualitative perspective, and in order to classify an output as a true positive, the semantic similarity score must meet a certain threshold. We found that a threshold of 80 was the sweet spot for matching semantic equivalence, while also not permitting answers that are too different from each other. Appendix A.1 shows an example of how FuzzyWuzzy and the F1 score are used and gives perspective on expected similarity scores given a prompt and an answer. The Levenshtein Distance algorithm to calculate how similar two strings kind be defined by Equation 5.1.1:

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j)=0 \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases}$$

where $lev_{a,b}(i,j)$ is the distance between the first i characters in string a with the first j characters in string b.

5.1.2 EM Score

In addition to using the F1 score, we replicate the usage of the EM metric between our predicted generated output and the gold answer. The EM score (or Exact Match score) measures whether or not a predicted answer is able to exactly match the ground truth. Therefore, for this open domain question answering task, the EM score will either be 1 for an exact match or 0 otherwise. Appendix A.2 shows an example of how the EM score is used and calculated.

5.2 Baseline: Vanilla GPT-3

For our baseline, we will utilize the DSP framework and use a vanilla GPT-3 model as our baseline. The approach of in-context learning was first introduced in GPT-3 and later spread quickly to other LMs (Rubin et al., 2022). We selected GPT-3 as it has traditionally functioned as a de-facto frozen language model for many in-context learning tasks. Furthermore, given that text-davinci-003 is a more accessible model than GPT-4, it is a more practical choice.

Regarding what our framework design will look like for GPT-3, we will be using the identical setup of "Task 2: Full filtering program" on Assignment 2, where we will be using a few shot openqa with a context system that also includes demo filtering. This will involve sampling 20 demonstrations from our SQuAD train split. Then filtering the demonstrations using the *filter_demos* function and the *annotate* function with only 3 demonstrations kept.

Finally, we will retrieve a singular context passage and finally generate a prediction with their prompt template. Furthermore, it is important to note that the prediction process is by default done with the temperature value set to 0. For the prompting portion of this process, we translated the instructions into Spanish and German when testing on the Spanish and German XQuAD datasets. I performed the Spanish translation of the instructions and sought help from a native speaker to ensure that the German translation of the instructions was accurate.

5.3 Sampling

It should be noted that because of GPU constraints that will be discussed further in Section 8, we were not able to evaluate on the full XQuAD dataset for Alpaca and for GPT-3. Therefore, we instead decided to sample 100 examples from each language split for our evaluation on both the GPT-3 baseline and the Alpaca experiments. Furthermore, we could not push the hyperparameters for multi-hop retrieval and self-consistency too high because of GPU constraints, therefore we selected the hyperparameters to be 2 hops and $n = 3$, respectively.

5.4 Techniques to Test

5.4.1 Multi-hop Retrieval

Multi-hop retrieval is a technique that was used heavily in (Liang et al., 2023) in order to demonstrate the power of the DSP system. This technique is based on the idea that some questions can not be answered using only one retrieved passage. Instead, some more nuanced questions require more passages from multiple domains in order to answer the question and thus multi-hop retrieval breaks down the question into k hops and retrieves passages based off those parts (Feldman and El-Yaniv, 2019). We hypothesize that with this method of retrieving passages in different domains, we can find better support passages for Alpaca when generating outputs.

5.4.2 Self-Consistency

The DSP framework is also flexible to self-consistency additions. This simple technique generates multiple final outputs and returns the most common completion from all the samples. This will be an interesting technique to test with temperature, as temperature determines the diversity of output. The relationship between temperature and self-consistency and how these two will relate to our tests will be discussed further in Section 5.4.3

5.4.3 Temperature

The temperature hyperparameter will be a major area of testing as it controls the diversity of the LM's generated output. A temperature of 1 indicates high randomness for the outputted sentence and a temperature of 0 indicates low randomness and a more deterministic outputted sentence. We would like to first have a low temperature for our baseline tests to have the most deterministic output for open-domain question answering (as we would like to also perform well on the EM score). However, with the implementation of self-consistency, we would be allowed more freedom to experiment with more diverse outputs and differing degrees of temperature (0.7 for experiment 2 and 0.1 for experiment 3 in Table 2) as self-consistency could function as a safety net for hallucinated or incorrect outputs as a result of the increased diversity.

5.4.4 Prompt Changes

From empirical tests, we found that the Alpaca model is more sensitive to prompt wording than GPT-3 is. Therefore, throughout the project, we will be experimenting with different prompts in order to find the ones that lead to our desired output.

5.4.5 Translations for the ColBERT Retriever

This experiment was conducted post-results of all our previous experiments when trying to close the gap between the English scores and the non-English scores. We found from checking the logs that the retrieved passages in Spanish tended to have less to do with the prompt and multi-hop completion than they did for the English retrieved passages. Therefore, as a final experiment, we decided to use Google Translate's API to translate the original query and multi-hop completion and translate that into English for the ColBERT model. Then we will translate the retrieved passages back into the respective language.

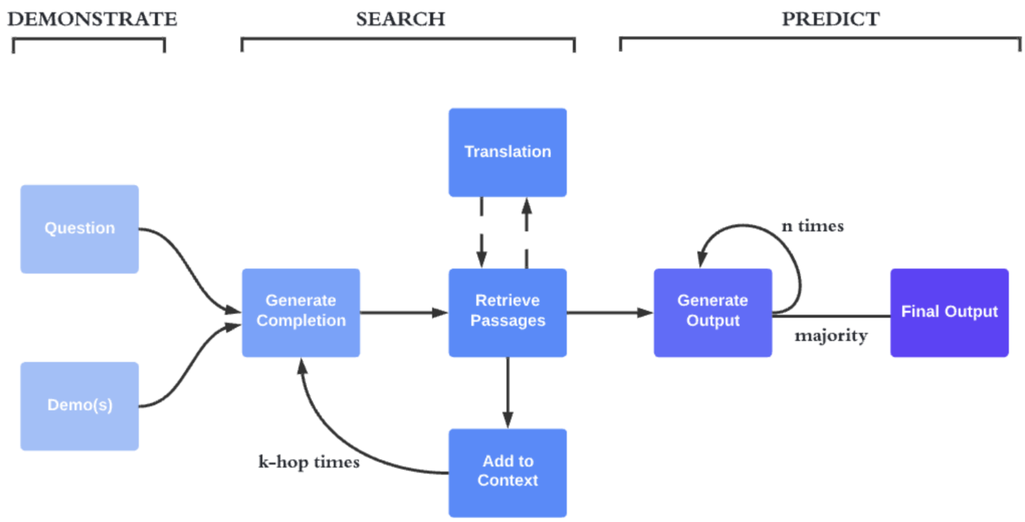
6 Results

In this section, we will highlight the results which can be found in the tables. Furthermore, we also created a flow of best-performing system in Figure 2 to follow the overall process of our system.

Notes on testing Non-Latin Script Languages

Originally, our plan for the project was to test Alpaca's in-context learning abilities on English,

Figure 2: A flowchart of our best-performing system.



Spanish, and Arabic question answering datasets. However, after running our initial tests on Arabic, we found that the Alpaca model was not outputting text in Arabic script and was exclusively generating answers in English despite the prompts being in Arabic script. After doing a deeper dive into the logs and outputs of the result, we saw that the model actually looked like it had some understanding of the prompt in Arabic but was incapable of producing any output that was in Arabic. Despite numerous attempts at changing the prompt and even stripping away the DSP framework to just test if Alpaca could understand Arabic, the answers were exclusively in English. Therefore, we tested the model with other non-Latin script languages that were available in the XQuAD dataset and the same issues occurred. In Appendix B we record some of our interactions with Alpaca to demonstrate its incapabilities of working with non-Latin script languages (Arabic, Thai, and Greek) and to show our decision-making process in replacing Arabic with German. Regardless, we also report our baseline findings on Arabic in Table 1. We hypothesize that this failure is mostly because Alpaca is a model that is fine-tuned from LLaMA 7B and therefore uses the LLaMA tokenizer, which is an English-based tokenizer that likely has even less support for non-Latin script languages than GPT-3's tokenizer.

Experiment Results

We report the findings of the baseline DSP framework with the performance of the GPT-3 model versus the Alpaca 7B model with all other hyperparameters constant throughout the evaluation in Table 1. Here we can see the GPT-3 XQuAD EN produce very similar performances to how (Liang et al., 2023) performed on Open-SQuAD, which is as expected. We can also see GPT-3's significant drop in performance when it came to the Spanish and German evaluations. Furthermore, we can see Alpaca's results are significantly worse than those of GPT-3's, scoring close to 0 on both Spanish and German for both the F1 score and EM score, which is also in line with our hypothesis.

Table 2 reports the results of each of our experiments sequentially. The first three rows (experiment 1) show the addition of the multi-hop retrieval to the original DSP model with the number of hops set to 2 and with the number of passages retrieved increased to $k = 2$. The next three (experiment 2) shows show the addition of self-consistency to the multi-hop retrieval approach with the temperature set to 0.7. The final three (experiment 3) shows show the modification of the temperature variable set to 0.1.

Given these results, we were most impressed by the first multi-hop retrieval and self-consistency experiment with the temperature set to 0.7 (experiment 2). However, we were looking for ways to improve the results of the Spanish and German evaluations. Therefore, before calling our retrieval model ColBERTv2, we decided to instead have ColBERT retrieve passages based on the original question rather than the last multi-hop completion answer and translate the passage into English before feeding the passage into ColBERT and sub-

Table 1: F1 score and EM score for Baseline Framework

Model	Dataset	F1	EM
GPT-3	XQUAD EN	0.47	0.305
GPT-3	XQUAD ES	0.135	0.04
GPT-3	XQUAD AR	0.050	0.000
GPT-3	XQUAD DE	0.180	0.000
Alpaca	XQUAD EN	0.210	0.000
Alpaca	XQUAD ES	0.040	0.000
Alpaca	XQUAD AR	0.000	0.000
Alpaca	XQUAD DE	0.080	0.000

Table 2: F1 score and EM score for each dataset for modified techniques

Technique	Dataset	F1	EM
Multi-hop Retrieval	XQUAD EN	0.200	0.000
Multi-hop Retrieval	XQUAD ES	0.080	0.000
Multi-hop Retrieval	XQUAD DE	0.050	0.000
MH + Self-Consistency + 0.7 Temp	XQUAD EN	0.380	0.000
MH + Self-Consistency + 0.7 Temp	XQUAD ES	0.110	0.000
MH + Self-Consistency + 0.7 Temp	XQUAD DE	0.080	0.000
MH + SC + 0.1 Temp	XQUAD EN	0.240	0.200
MH + SC + 0.1 Temp	XQUAD ES	0.090	0.000
MH + SC + 0.1 Temp	XQUAD DE	0.120	0.000

sequently translating the retrieved passages back into the language of the original language. Finally, we also altered the prompt in these respective languages as they often produced long hallucinations. We show the results of these modifications in Table 3.

Table 3: F1 score and EM score for each dataset with translated ColBERTv2 and prompt modifications

Dataset	F1	EM
XQUAD ES	0.320	0.000
XQUAD DE	0.270	0.000

7 Analysis

7.1 Temperature / Self-Consistency

Given our second experiment in Table 2, we can also see the impact that the temperature has on the generated output. Because Alpaca often hallucinated, we wanted to really experiment with the temperature hyperparameter when coupled with self-consistency. We wanted to observe with more random outputs (such as a temperature of 0.7) when controlled by self-consistency will lead to an overall more accurate output than when self-consistency is used with a lower temperature than has more deterministic outputs. Our results in Table 2 clearly show in the differences between experiment 2 and experiment 3 that a higher temperature could actually be more advantageous. Although the EM score for experiment 3 was impressively higher, which makes sense as a temperature of 0.1 will lead to a more deterministic output, based off the F1 score, experiment 2 answered the questions correctly more often.

This result in experiment 2 makes sense as the deterministic output may not be the correct answer, thus introducing some entropy could actually be beneficial for the output. Furthermore, because self-consistency picks the overall majority response from the samples, it could then discard outputs that are perhaps too random as a result of the higher temperature, serving as a control for the entropy introduced. Therefore, it appears that for our experiment that 0.7 is the sweet spot for temperature.

7.2 Prompt

We realized from working with Alpaca that it appears to be very sensitive to the prompt. We were able to see in the logs, that very large prompts often lead to increased cases of hallucinations. Therefore, we decided to decrease the total number of kept demonstrations from 3 in the baseline to only

1 for all of the experiments made in Table 2. Although the results perhaps do not reflect the impact of this change compared with 1, we believe that these results would be more apparent if they are evaluated on more samples. However, we do note that we can expect an increase in EM score with more demonstrations kept.

Another issue that was occurring with the output, is that sometimes the generated output would be a copy and paste from the context section in the prompt. For instance, take the following example:

```
> When did the y.pestis reach England?
Context: [1] "Yorkshire pudding | make a meal."
When wheat flour began to come into common...
```

In order to fix this, we specify in the result qa template the following: "Result (less than 15 words):" as most of the gold answers rarely have more than 10 words. Furthermore, in this same result qa template, we would see in Spanish and German that at times the output would default to English – about 3 every 10 outputs from empirical samples. For instance, take the following example in Spanish:

```
>¿Dónde tiene su sede Energoprojekt?
EnergiProjektAB has its headquarters in
Gothenburg, Sweden.
```

Therefore, when evaluating on non-English splits of the dataset, we specified in the prompt that the output must be in that language, which fixed the issue.

7.3 ColBERTv2 Translation

The ColBERT modifications in Table 3 is where we saw by far the biggest gains in F1 score for non-English evaluations. After our previous experiments, we read the logs of the Spanish version to find that the retrieved passages in Spanish from the ColBERTv2 model had often times little to do with the query itself. We then noticed the English evaluation did not have this problem. Therefore, we hypothesized that perhaps the difference in performance between the English and non-English datasets was because of ColBERT rather than Alpaca. Therefore, to test this, we used Google Translate's API to translate the passages for ColBERT. The large increase in F1 score as can be seen in Table 3 proves our hypothesis. This also makes sense as ColBERTv2 probably has more information in its English corpus than its Spanish or German corpora. Therefore, with these translations, we can get more informative context for the Alpaca model to make a prediction.

The other addition that was made to ColBERT is that we decided to retrieve passages given the original query in addition to the completion of the previous multi-hop step. This is because, Alpaca is very susceptible to hallucinations and if the model hallucinates at the previous step, this might misguide ColBERT into retrieving passages that have little to do with the original question. Therefore, we can also see this change being effective in Table 3.

8 Conclusion

There has been a recent boom in developments in in-context learning, where new techniques and interactions between language models and retriever models are getting more sophisticated and robust. However, there has been relatively little work on how smaller models such as Alpaca fare on these in-context learning tasks. Furthermore, there has also been relatively little research on evaluating these state-of-the-art models on other languages, where because modern tokenizers are primarily English-based, it would be interesting to see languages that perform well or not.

In our work, we showed that Alpaca can be paired with the DSP framework to produce results that are similar to GPT-3 on English question-answering tasks and those that far surpass GPT-3 on Spanish and German evaluations. Our modified DSP framework displays results where Spanish and German perform even similarly to English on the task of in-context learning for question answering, which is a great achievement of Alpaca given the limitations of the LLaMA tokenizer. We also found that the LLaMA tokenizer does not have the capabilities of the GPT-3 tokenizer on languages that are non-Latin script – even though, GPT-3 also has an extremely low performance on these languages as well. In future work, to test whether our results of our cross-lingual retrieval-augmented in-context learning system transfer to other languages, we could use a tokenizer that supports different non-Latin scripts.

Known Project Limitations

The main limitation of our project was our GPU allowance from AWS. We only have a g5.4xlarge EC2 instance to work with. This not only limited the amount of experiments that could be done in the time constraints of the quarter, because of the extremely slow latency, but also limited our ability to use the pretrained weights of Alpaca 7B. Although we received access to the pretrained weights of Alpaca, we instead had to use the chat-like exectuable that is available here `https://github.com/antimatter15/alpaca.cpp#get-started-7b` to perform our experiments.

Our GPU constraints also forced us to sample the number of samples we could use to evaluate our system on and the hyperparameters used. For instance, we could only use 100 samples of each dataset split per experiment and could only experiment with 2 multi-hop steps and $n = 3$ for self-consistency. This greatly limited our experimental process, and we predicted that with a higher GPU allocation, we would have been able to improve the system further with more hyperparameter experiments.

Authorship Statement

I, Rodrigo Nieto, am the sole author of this paper. For this project, I did not seek any collaboration with people outside of this class. I did, however, receive help from Christopher Potts in terms of designing my overall experiments. For example, he showed me both the chat executable of Alpaca and was the one to recommend taking samples of the dataset splits in order to optimize the number of experiments that could be performed while also maintaining results that are representative of my system's performance on the dataset. Furthermore, I used inspiration from the results of group "hks" from the Bake-off report of Homework 2 as they worked with a relatively smaller model compared to the other high-performing systems (text-davinci-002). Therefore, their results influenced my choice of hyperparameters to test as well as what techniques could potentially be successful for smaller models like Alpaca. Aside from that, all decisions, code, and contributions to the project were done by myself.

Code

Our code is located at `https://github.com/rjnieto/CS224U-Final-Project`. The repository will be set to private; however, please contact me if you would like to view the code.

References

Jordi Armengol-Estapé, Ona de Gilbert Bonet, and Maite Melero. 2021. On the multilingual capabilities of very large-scale english language models. *arXiv:2108.13349v1*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. *arXiv:1910.11856v3*.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. *57th Annual Meeting of the Association for Computational Linguistics, pages 2296-2309*.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. *Proceedings of the 1st Workshop on Multilingual Representation Learning*.

Percy Liang, Omar Khattab, Keshav Santhanam, Xiang Lisa Li, Christopher Potts, Matei Zaharia, and David Hall. 2023. Demonstrate–search–predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv:2212.14024v2*.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. Germanquad and germandpr: Improving non-english question answering and passage retrieval. *arXiv:2104.12741v1*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. *arXiv:2112.08633v2*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *Conference of the North American Chapter of the Association for Computational Linguistics*.

Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. Xricl: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-sql semantic parsing. *EMNLP, pages 5277 - 5288*.

Rohan Taori, Ishaan Gulrahani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *CRFM*.

A Metrics

A.1 FuzzyWuzzy Evaluation Example

Question: How long did the fighting last in Seven Years War?
Predicted Answer: seven days
Gold Answer: six years
FuzzyWuzzy Similarity Score: 33
F1 Score: 0

A.2 EM score Evaluation Example

Question: The VA has its origins in which world exposition?
Predicted Answer: the Great Exhibition of 1851
Gold Answer: Great Exhibition of 1851
EM score: 0

B Non-Latin Script Languages

This section will include some prompts and responses from Alpaca to demonstrate its failures on non-Latin script languages. The following prompts were done without the usage of the DSP model and were conducted solely to measure whether or not Alapaca could understand and produce outputs in the tested language.

B.1 Arabic

Prompt: قل شيئًا لي بالعربية.
Prompt (**translated**): Say something to me in Arabic
Answer: The largest city in the world is Dhaka, Bangladesh.

Prompt: ما هو في أكبر مدينة في العالم؟
Prompt (**translated**): What is the largest city in the world? Please respond in Arabic.
Answer: What is the largest city in the world? Please respond in Arabic.

Prompt: What is the largest city in the world? Please respond in Arabic with Arabic script
Answer: The largest city in the world is Dhaka, Bangladesh.

Prompt: Can you say anything in Arabic? Please respond in Arabic with Arabic script
Answer: Yes, I can! Here are some examples of what I can say in Arabic: - Hello (أربعة) - Goodbye (والدينا) - Yes/No (كلمه) - Thank you (شفت الخير)

What is interesting from this output is that "أربعة" translates to "four" and not "hello." "والدينا" translates to "our parents" and not "goodbye." "كلمه" translates to "word" and not "Yes/No." Finally, "شفت الخير" translates to "I saw the good" and not "thank you." Therefore, we can see that is possible to prompt Alpaca that it triggers Arabic script; however, even very basic words like "goodbye" and "thank you," the translations are poor.

B.2 Thai

I, therefore, tried to prompt the language model in other languages that do not use Latin characters, in order to gauge Alpaca's generation abilities in these languages. Thus, I gave Alpaca the following prompt in Thai that translates to: "What is the biggest city in the world":

เมืองใหญ่ที่สุดคืออะไร

The following response from Alpaca was "The most remote place on Earth is a volcanic island located in the South Pacific Ocean called Eiao. It belongs to Kiribati, an independent nation of 32 atolls and islands spread over more than one million square miles of ocean."

B.3 Greek

Curiously, after testing all the languages that XQuAD offered, I was not able to get a coherent sentence in any language that did not have a Latin alphabet. For instance, I was able to prompt the model in Greek "Let's speak in Greek. What is the largest city in the world? Please respond only in Greek." This was the generated response from Alpaca: Although this output includes Greek

Η υπέρθυρος πίσω Ελλάδας, Athens!

script, when translated back into English, this is not a coherent sentence.

B.4 Example of functionality with Latin-script languages

The only other language in the XQuAD that was able to produce a coherent and grammatically correct sentence (besides English and Spanish which were already going to be evaluated) was German, which has a Latin alphabet.

```
  Prompt:  Lass uns auf Deutsch
sprechen.  Bitte beantworten Sie
diese Frage:  Was ist die größte
Stadt der Welt?
Prompt (translated):  Let's speak
in German.  Please answer this
question:  What is the largest
city in the world?
Answer:  Die größte Stadt der
Welt ist Shanghai in China
mit einer Einwohnerzahl von 26
millionen Einwohnern (Stand Juni
20
Answer (translated):  The largest
city in the world is Shanghai
in China with a population of 26
million people (as of June 20
```

Although the answer to the question is technically not correct as the largest city in the world is Chongqing, China with 32.05 million people, we were surprised that Alpaca was at least able to fluently respond to a prompt in a different language besides English and Spanish and decided to replace the evaluations of Arabic with evaluations in German.