Cooking Guidance: Integrating RAG Methodology into a Quechua NLP Recipe Consultant

Bruno del Río, Ángel Lino, Fernando Candia, Bryan Ruiz, Alexander Taboada
Pontificia Universidad Católica del Perú
Lima, Perú

***Abstract*— This paper introduces a novel approach to cooking guidance in Southern Quechua, a resource-scarce language spoken primarily in Peru through the integration of a Natural Language Processing (NLP) based tool such as Retrieval-Augmented Generation. We discuss the design considerations for the corpus extraction and implementation details covering the solution training and the deployment of the web service via an API.**

Keywords*— *Retrieval-Augmented Generation, Quechua, Recipe, Large Language Model

I. Introduction

Natural Language Processing (NLP) has transformed various fields of knowledge and everyday practice, from automatic translation to content generation. However, the inclusion of indigenous languages in these technological advancements has been limited. This paper presents a guide for integrating the Retrieval-Augmented Generation (RAG) methodology into a recipe consultant in Quechua, one of the most widely spoken indigenous languages in South America.

The RAG methodology combines information retrieval with text generation, allowing the system to consult large databases and generate coherent and contextualized responses. This methodology has proven effective in question-answering and content generation applications, offering a robust solution for creating intelligent conversational systems [1].

Choosing Quechua as the target language addresses the need to preserve and revitalize indigenous languages through technology. According to the National Institute of Statistics and Informatics of Peru, more than 3.7 million people speak Quechua in the country, highlighting the importance of developing technological tools that support and promote its use [2].

In this context, implementing a recipe consultant in Quechua not only facilitates access to traditional cuisine but also promotes cultural and linguistic transmission to new generations. This work is based on the RAG methodology to develop a system capable of understanding and generating texts in Quechua, offering traditional recipes and culinary advice adapted to the users' preferences and needs.

Integrating RAG into Quechua processing poses several technical and linguistic challenges. Quechua, with its multiple dialects and variants, presents additional complexity in terms of language normalization and standardization [3]. Moreover, the availability of digital resources and corpora in Quechua is limited, making it difficult to train advanced language models [4].

II. Related work and Background

The integration of indigenous languages into NLP systems has garnered attention in recent years, particularly in the context of language preservation and revitalization. Various studies have explored different methodologies and applications for indigenous language processing, which provide a foundational background for our work on integrating RAG into a Quechua recipe consultant.

A. Indigenous Language Processing

Previous efforts in indigenous language processing have primarily focused on creating basic linguistic resources such as corpora, lexicons, and morphological analyzers. For instance, [1] discusses the development of morphological analyzers for various indigenous languages, emphasizing the importance of linguistic resources in facilitating NLP applications. Similarly, [2] outlines the creation of digital corpora for indigenous languages, highlighting the challenges of data scarcity and the need for community involvement in data collection.

In the context of Quechua, [3] presents a comprehensive overview of the linguistic characteristics and dialectal variations within the language, underscoring the complexity of developing standardized NLP tools. This work provides valuable insights into the linguistic intricacies that must be addressed when building an NLP system for Quechua.

B. Retrieval-Augmented Generation (RAG)

The RAG methodology combines the strengths of retrieval-based and generation-based models, offering a hybrid approach that enhances the accuracy and relevance of generated responses. RAG retrieves pertinent information from a large corpus and then uses a generative model to produce coherent and contextually appropriate text. This methodology has been successfully applied in various NLP tasks, including open-domain question answering and conversational agents.

In their seminal work, Lewis et al. [4] introduced RAG as an effective approach for knowledge-intensive NLP tasks. Their experiments demonstrated that RAG outperforms traditional retrieval and generation models by leveraging the strengths of both paradigms. Similarly, Karpukhin et al. [5] showcased the effectiveness of dense passage retrieval, a key component of RAG, in improving the performance of open-domain question answering systems.

C. Applications in Recipe Consultation

Recipe consultation systems have been a popular application area for NLP, providing users with cooking advice, ingredient substitutions, and personalized recipe recommendations. These systems typically rely on large databases of recipes and culinary knowledge to generate accurate and useful responses. Research in this domain has explored various approaches, from rule-based systems to advanced machine learning models.

For example, [6] describes the development of a multilingual recipe recommendation system, highlighting the challenges of language processing and personalization in the culinary domain. Additionally, [7] explores the use of NLP techniques to understand and generate recipe texts, emphasizing the importance of context and user preferences in delivering relevant recommendations.

D. Challenges and Opportunities

Integrating RAG into a Quechua recipe consultant presents unique challenges and opportunities. The primary challenge lies in the limited availability of digital resources and corpora in Quechua, which complicates the training and fine-tuning of NLP models. Furthermore, the linguistic diversity and dialectal variations within Quechua require careful consideration in developing a standardized approach to language processing.

However, the integration of RAG offers significant opportunities to enhance the capabilities of the recipe consultant. By leveraging retrieval-augmented generation, the system can provide accurate and contextually relevant culinary advice, bridging the gap between traditional knowledge and modern technology. This approach not only facilitates access to traditional Quechua cuisine but also promotes the preservation and revitalization of the Quechua language.

III. METHODOLOGY

A. Corpus description

The corpus used in this study comprises a collection of 280 recipes in Spanish and Southern Quechua, which were obtained through a combination of web scraping and translation processes. The selected recipes were chosen to represent the diverse array of culinary traditions and ingredients in order to ensure a broad and comprehensive dataset.

In addition to the batch of recipes, the corpus includes 10 documents originally written in Southern Quechua, sourced from legal documents, dictionaries and the bible [8]. Together, these documents provide a comprehensive overview of a scarce-resource language such as Southern Quechua and will enhance the overall utility of the corpus making it a valuable resource in the development of language technologies aimed at preserving and promoting this variant of Quechua.

B. Methodology for the corpus extraction

The corpus extraction for this paper involved a combination of automated and manual methods to gather the dataset of recipes and related documents to the target language. In order to achieve this, the following methodology has been employed:

1) Web scraping for recipe extraction

The implementation of a web scraping process helped to successfully collect around 280 recipes from various online sources divided into 3 different sections: name of the recipe, ingredients and instructions.

2) Data Cleaning and Preprocessing

After the web scraping, the extracted recipes underwent a data cleaning and preprocessing phase, which involved standardizing the format of the recipes to comply with the 3 sections described above, correct inconsistencies in measurements and also handle errors during this process. This step was essential to ensure the quality and reliability of the recipes dataset.

3) Automated Translation

In order to obtain the recipes in Southern Quechua, we automated the translation from the original batch of 280 recipes using Google Translate's API, which were already cleansed and preprocessed to ensure a more accurate translation.

4) Manual Search

In addition to the automated methods, we conducted a manual search to identify and extract 23 relevant documents written in Southern Quechua. This manual effort was essential to locate these types of documents such as dictionaries because they provide contextual information and are an important factor for a robust corpus in a language with low representation. [4]

C. Methodology for the solution training

For the solution, we fine-tuned a pre-trained model (Llama-2-7b), using the "TrainingArguments" function from the "transformers" Hugging Face python library. We employed the "steps" strategy for evaluation, allowing the model to be evaluated and checkpoints saved every 100 steps for a total of 500 steps.

The optimizer used was "paged_adamw_8bit", which is suitable for handling large models efficiently in a resource-constrained environment. Additionally, we used a linear Linear Regression scheduler type to adjust the learning rate dynamically during training.

Data preparation involved loading training and validation datasets from CSV files, organized into a DatasetDict for easy manipulation. We used the tokenizer from the pre-trained Llama-2-7b model, configured to add end-of-sequence tokens and handle padding appropriately.

The training parameters were fine-tuned using LoRA (Low-Rank Adaptation) to enhance training efficiency and model performance. The training was conducted using the SFTTrainer, a specialized class for fine-tuning large language models.

During the training, we monitored the training and validation loss at each checkpoint. The training started with a loss of 4.525400 at step 100 and progressively improved to 2.387000 by step 500. Similarly, the validation loss decreased from 2.974348 at step 100 to 2.371483 by step 500. These checkpoints, saved every 100 steps, ensured that the model's progress was tracked and could be resumed if needed.

The fine-tuned model and tokenizer were saved locally and compressed for further use and analysis. The final training

output indicated a global step of 500 with a training loss of 2.926580, showcasing the efficiency of the fine-tuning process and the improvements made over the course of the training.

D. Methodology for the solution deployment

For the final deployment of the solution, we also added the RAG layer, which is feeded by the pre-trained model we fine tuned.

Initially, we set up the environment by installing the necessary dependencies, including the deep-translator library, to ensure compatibility and proper functionality. Preprocessing and translation functions were defined to handle the conversion of recipe text from Spanish to Quechua. The preprocessing function parsed the recipe files to extract key sections such as the recipe name, ingredients, and instructions, while the translation function utilized Google Translator for accurate translations. To manage file operations, utility functions were created to write the translated Quechua recipes to text files and retrieve the original Spanish recipes from the specified directory. Subsequently, we implemented a systematic approach to process and store the translated recipes by creating a dedicated function that translated each recipe and saved the output in a predetermined directory, ensuring consistency and accuracy.

We developed and locally hosted an API to interact with the model. We then used that API to develop a simple web page, where we could give inputs to our model and it could give back a corresponding output.

IV. Conclusion

The training of the model gave us poor results of 3.065. We concluded that the reason behind this was the low quantity of training data we were able to secure. Likewise, we got similar results when actually testing the application. Despite further optimization of our model, we were not able to produce better results. We end this discussion with the knowledge that for future developments, a greater quantity of resources must be found and used.

References

[1] . Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv preprint arXiv:2005.11401, 2020.

[2] National Institute of Statistics and Informatics (INEI), "Peru: Sociodemographic Profile, National Report," 2020.

[3] W. F. H. Adelaar, *The Languages of the Andes*. Cambridge University Press, 2004.

[4] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. T. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," arXiv preprint arXiv:2004.04906, 2020.

[5] J. G. Pérez-Beltrachini and J. A. Rodríguez, "Morphological Analyzers for Indigenous Languages: Challenges and Solutions," *Journal of Computational Linguistics*, vol. 46, no. 3, pp. 789-807, 2020.

[6] L. Chitic, G. Mihalcea, and C. Banea, "Multilingual Recipe Recommendation and Personalization," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, Marseille, France, 2020, pp. 1235-1242.

[7] S. Majumder and A. Ekbal, "Understanding and Generating Recipe Texts Using NLP Techniques," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, Online, 2020, pp. 6755-6766.

[8] Nouman Ahmed. (2023). *MT-SharedTask/data.* GitHub. https://github.com/nouman-10/MT-SharedTask/tree/main/data