A Few-shot Learning Approach for a Multilingual Agro-Information Question Answering System

by

Fiskani Ella Banda

Submitted in partial fulfillment of the requirements for the degree
Master In Information Technology (Big Data Science)
in the Faculty of Engineering, Built Environment & Information Technology,
University of Pretoria, Pretoria

December 2023

Publication data:

Fiskani Ella Banda. A Few-shot Learning Approach for a Multilingual Agro-Information Question Answering System. Masters Dissertation, University of Pretoria, Department of Computer Science, Pretoria, December 2023.

A Few-shot Learning Approach for a Multilingual Agro-Information Question Answering System

by

Fiskani Ella Banda
E-mail: thefiskanibanda@gmail.com

Abstract

Agriculture plays a crucial role in numerous households across Sub-Saharan Africa. Developing a question answering system that utilizes agricultural expertise and agro-information can effectively bridge the support gap for farmers in the local community. Most advances in question answering research involve large language models trained on extensive data. Nevertheless, the conventional approach of fine-tuning has demonstrated a significant decline in performance when models are fine-tuned on a small amount of data. This decline is primarily attributed to the disparities between the objectives of pre-training and fine-tuning. One proposed alternative is to utilize prompt-based fine-tuning, which permits the model to be fine-tuned with only a few examples. Extensive research has been done on the application of these methods to tasks such as text classification and not question answering. This research aims to study the feasibility of recent few-shot learning approaches, such as FewshotQA and Null prompting, for domain-specific agricultural data in 4 South African languages. We evaluated the overall performance of these approaches and investigated the effects of adapting these approaches for cross-lingual extractive question answering of domain-specific data. The results obtained in this study have shown valuable insight into the applicability of these methods to domain-specific data. These results have shown that these methods are capable of adequately capturing the textual information of domain-specific data from the initial subset of data points. Thus, there is potential for using these methods as a practical solution for limited data.

Keywords: Agriculture, cross-lingual, extractive question answering, few-shot, low resource languages, natural language processing.

Supervisors : Prof. V. N Marivate
Dr. J. N. Nabende
Department : Department of Computer Science
Degree : Master of Information Technology (Big Data Science)

"We must appreciate that in a society of higher poverty levels, inequality and low growth, getting agriculture going is critical for various reasons."

Angela Thokozile Didiza, Minister of Agriculture, Land Reform and Rural Development (State of the Nation Address 2022)

"While improving the representation of African languages in cutting-edge NLP research, it is vital that African NLP activities lead to greater access and better quality of life for populations that speak African languages"

Siminyu, K. et al., (2023) 'Consultative engagement of stakeholders toward a roadmap for African language technologies'

Acknowledgments

I would like to recognise and express my gratitude to the following people who have been instrumental throughout this degree :

- My Supervisors, Prof. Vukosi Marivate and Dr Joyce Nabende, for the exposure and the opportunity that was presented by this research study.I have gained a new perspective on the potential impact of Natural Language Processing at a local level. I am also very grateful for all the guidance, patience and encouragement that was offered throughout;
- To my family, Prof. Eunice Mphako-Banda, Prof. Mapundi Banda, Marumbu Banda, Chikondi Banda, for the support, words of encouragement, and comfort that has been provided throughout this stressful time;
- My partner, Siyabonga Ngcobo, for being my sounding board when I encountered problems and for always motivating me to reach the finish line.

Contents

List of Figures

List of Algorithms

List of Tables

Chapter 1

Introduction

History plays an important role in any country's socioeconomic challenges that are still being felt in this day and age. For contemporary South Africa, one of the major challenges is poverty. One deeply rooted historical and sociopolitical factor that has contributed to this challenge is the Apartheid system. This system marginalised the majority of Africans and pushed them to the edge of ecological and social economic systems. This was systematically done through disenfranchisement and depriving them of the freedom deemed necessary to achieve valuable functioning in society[34].

Poverty in South Africa can be classified as both structural and multidimensional. This means that there are several factors that contribute to poverty, in addition to just an individual's financial status. This results in the need for more complex solutions to combat the problem, in addition to one-dimensional solutions such as employment. Such solutions have minimal impact, as they only address a single aspect of the problem[18].

Targeted interventions are, however, the way to go. One of these interventions is through small-scale or small-holder farming. In many South African households classified as in a state of poverty, small-scale farming is at the centre of their income and food security. On a larger scale, these farms make up a great deal of the food supply. In an ideal world, the aim would be to maintain and close yield gaps to ensure self-sufficiency, food security, and an income for a lot of these households, thus addressing various dimen-

sions of poverty. To achieve this, one can consider reducing the factors that contribute to agricultural challenges by creating suitable conditions for local farmers. [16, 18].

Currently, many of these farmers face these challenges without any resources or assistance. To bridge this gap, factors such as initiatives that promote and support local farmers and investments in agricultural infrastructure can be useful. The Department of Communications and Digital Technologies in South Africa has made suggestions to integrate Information and Communication Technologies (ICT) with agriculture [18]. An ICT solution has been suggested is building a low maintenance support system. Such a system has the potential to provide support and reach a large majority of small-scale farmers in need.

A system that can be crucial to helping farmers is a Question Answering (QA) system. A QA system is built to be able to automatically answer questions that are posed by a user, in this case it would be farmers. This can be achieved using a Language Model (LM) in which answering the question is the Natural Language Processing (NLP) downstream task. The language model is tasked with retrieving the most appropriate answer to a question based on a given text/document.

In this study, we focus on one such solution, which is a platform that can provide agro-information (agricultural information) to farmers during different times of the planting cycle. We focus on the aspect of QA, where farmers can ask questions and an answer can be returned based on readily available agro-information. The hope is that this solution will play a critical role in the decision-making process for farmers. We take advantage of readily available multilingual agro-information to create a QA dataset.

Using this dataset, we leverage known few-shot model fine-tuning techniques for Pretrained Language Model (PLM) to determine the feasibility and applicability of these techniques for a domain-specific cross-lingual QA dataset. This chapter provides an introduction to the research study, a discussion of the motivation behind doing this study, followed by a detailed breakdown of the research questions and the corresponding

objective, and lastly a detailed outline of the rest of this paper.

1.1 Motivation

Globally, there are several research studies based on the use of QA systems in the agricultural domain. The more recent studies [11, 14, 51] in this field have been based on agriculture and farming in India, where the main conclusion from all these is the significance of such systems for farmers. These studies deal with the development and implementation of QA systems in which a precise response is generated for a user query. Because agriculture is influenced by various environmental factors and is specific to each region, it is challenging to transfer agricultural systems to South Africa due to the unique conditions and requirements of the country.

Currently, the main source of this information is agricultural experts as well as fellow farmers, in an African context. This means that there is a need for farmer-to-farmer and farmer-to-expert interactions. The Adhoc surveillance tool by Makerere University [32], tries to cater to these interactions through an Information Technology (IT) solution. This tool was created to help monitor cassava plants and the pests and diseases that affect them. The limitation of this tool is the reach that it has as there is a limited amount of agricultural experts to address the issues posed through the tool.

With such a limitation, it means that farmers do not always get the information they need when they need it. Another consideration to be accounted for is the language which is used on such platforms as the majority local small-holder farmers in African countries do not speak and/or cannot read and write English. This presents a language barrier when trying to use the platform. Although there are solutions [11, 32, 51], most of them are monolingual and provide information in major languages such as English. For many of these farmers, even if the information is readily available, it does not cater to them due to this language barrier.

In this research, a different approach is presented, a multilingual/cross-lingual QA

system for farmers that supports 1 major language, English, and low-resourced South African languages. This will hopefully address the limitations presented by some of the current solutions. We focus on the use of multilingual representational models that have proven pivotal in the area of Natural Language Understanding (NLU). Although there is an exponential increase in the focus on developing datasets in African languages, commonly referred to as Low Resource Language (LRL) such as AfriQA [33] and KenSwQuAD [49], no domain-specific text corpora are readily available for agro-information. By using readily available South African agro-information, another contribution can be made by creating a novel domain-specific dataset in South African languages.

1.2 Objectives

The primary goal of this dissertation is to address the identified problem while taking into account the limited scope. The primary constraint is limited data, as the data must be about agriculture and be available in at least one of the low-resource South African languages. This can be done by trying to answer the following research question.

> *Are recent prompt-based fine-tuning techniques feasible for multilingual domain-specific text in the context of agro-information Question Answering for Low Resource Languages ?*

By considering this research question, we aim to investigate the feasibility of some of these techniques which have been studied for a variety of NLU tasks, mainly text classification. To fully answer this question, it can be broken down further into the following sub-questions and corresponding objectives:

1. What automated process can be used to effectively create a QA dataset based on agro-information in multiple languages?
    (a) Determine the tools that can be used to develop a parallel multilingual dataset to remove the need for machine translation.

(b) Determine what methods can be used to generate a high-quality question-answering dataset.

2. What is the feasibility of applying the model fine-tuning techniques for domain-specific monolingual(English) QA data ?

   (a) Determine what adaptations, if any, can be made to the established fine-tuning techniques that result in promising results.

3. Based on the results returned from the previous sub-question, what is the feasibility of applying the same model fine-tuning techniques for multilingual domain-specific QA data in LRLs?

   (a) Determine what adaptations, if any, can be made to the established fine-tuning techniques that result in promising results. These adaptations consider language-specific linguistic adaptations.

1.3 Contributions

From this study, the following novel contributions are made :

- The collection and curation of a parallel multilingual QA agricultural corpus that is connected to Agriculture in South Africa
  - The development of a novel automated pipeline to create an English question answering dataset using a Large Language Model (LLM).
  - The extension of this pipeline to make use of the parallel nature of the corpus to create a multilingual dataset.
- Showing the potential of the use of different few-shot fine-tuning approaches in a different setting - cross-lingual extractive QA for agro-information. By establishing this foundation, the use of prompt-based few-shot learning for limited domain-specific multilingual data is feasible. This work can be expanded to further investigate different methods designed specifically for QA and the addition of language-specific prompting methods.

1.4 Dissertation Outline

- **Chapter 2** provides a comprehensive review of the research done on agricultural information systems. It delves deeper into research done for information retrieval to provide the necessary background to understand the recent research that has been done and where the research study falls.
- **Chapter 3** focusses on the technical background that is needed to understand the experimental setup. It discusses the different models and the fine-tuning methods that are considered.
- **Chapter 4** introduces the text corpus that is used in this study. Provides an overview of the data source, collection, and pre-processing that is followed to prepare this corpus for the data annotation.
- **Chapter 5** give an overview of the analysis performed on the corpus of collected and processed text. From this exploration, the attributes of the data were provided.
- **Chapter 6** gives a detailed discussion of the different components that contribute to the automated annotation pipeline used to create the final QA dataset.
- **Chapter 7** explores the different attributes of the resultant annotated dataset through NLP data analysis techniques. This analysis is used primarily to evaluate the quality and consistency of the annotated data.
- **Chapter 8** provides a detailed account of the experimental setup used in this study. It outlines the different scenarios that are studied and provides additional technical information.
- **Chapter 9** discusses the experimental results that focus on the use of fine-tuning methods for a monolingual domain-specific dataset. The aim of the experiment is to answer the second sub-question of this study.
- **Chapter 10** gives a detailed discussion of the results obtained, where the focus is on the use of the same fine-tuning methods for a domain-specific multilingual

dataset. This aims to answer the third sub-question and subsequently answer the main research question .

- **Chapter 11** combine the insights observed throughout the study to provide a summary of the main findings of this study. This discussion also looks at the limitations and future work that can be done to extend the research achieved.

To support the main details of this study, some additional material has been included in the following appendices :

- **Appendix A** provides a list of examples obtained during the document matching stage of the data preparation done in Chapter 4. It aims to provide examples for each of the languages used in this study.
- **Appendix B** gives a more detailed account of the different components used in the data annotation pipeline described in 5.
- **Appendix C** provides an example of the output of the phrase generation algorithm.
- **Appendix D** gives the 2 evaluation flow charts that are used to determine the quality of the generated questions and answer pairs.
- **Appendix E** give the final dataset subset sizes for the different languages. The datasets are split into training, development, and testing subsets for the experimentation.
- **Appendix F** provides the detailed results obtained for the different runs performed for domain-specific experimentation.
- **Appendix G** provides the detailed results obtained for the different runs conducted for cross-lingual experimentation.

Chapter 2

Literature Review

The incorporation of ICT into agriculture, specifically in South Africa, has been highlighted as a solution to the problems faced by small-scale farmers. This has inspired more research on the use of Machine Learning (ML) and Artificial Intelligence (AI) as solutions to incorporate to support agriculture. A common solution that has recently been studied in a South African context is the use of images and remote sensing with ML [30, 31, 36]. These solutions try to use ML to assist with mapping of crop types and yield prediction.

By using such systems, the yield of a crop can be predicted and can be used to provide information back to small-holder farmers. A theme that has been emphasised in all three studies is the importance of information in the decision-making process of the farmers involved. Although these systems provide valuable information, they are still limited in the type of information that can be used to maximise the production of small-scale farming. If a farmer encounters problems with pests and diseases, these systems cannot help in regulating or managing the issue.

One such study that attempts to address this is the Adsurv system in [32]. In this study, the research is based on using a mobile crowd-sourcing platform for the surveillance of viral diseases and pests in cassava in Uganda. This system relies on farmers uploading images and questions. In response to this, experts and other farmers can re-

spond to the questions. The main source of information is the knowledge of the people on the platform. However, the limitation with this solution is still access to information, as it is limited to those on the platform.

In general, all of these studies have indicated the potential that exists in this field and the need for a more comprehensive platform that can provide accurate and specific information to farmers in a timely manner. This study aims to investigate the plausibility of such a system using large language models. This chapter discusses the conceptual background of information systems and QA systems in Section 2.1. Subsequently, the application of agro-information to information systems, describing their data, methodologies, and applicability of these systems to the objectives of this study, is discussed in Section 2.2. This study made use of NLP question answering methods, which are presented in Section 2.3. The main focus in this study was the use of fine-tuning of large language models on limited domain-specific data; therefore, in Section 2.4 we briefly discuss fine-tuning approaches. Lastly, a summary of the information discussed is included in Section 2.5.

2.1 Information Systems

Information retrieval and information extraction form the core foundational concepts of QA systems. By combining these two concepts, the system is able to provide more concise and direct responses to a user query/question, as compared to using the search functionality of search engines. As such, QA can be defined as a NLP task.

2.1.1 Information Retrieval

Information retrieval research started growing from 1961. It is defined as an encapsulation of all the activities that are involved in the organisation, processing, and accessing of information. This information can be presented in various forms such as images, text, sound via recording, etc. An information retrieval system is an extension of this concept, where the system is intended to bridge the gap between people and information systems, allowing people to communicate with the system and get relevant information that meets

their needs [8] .

To simplify the overall flow of the system, the process can be divided into two sides: the side of the query from the user and the side where the documents are stored and organised. In linking these two processes, there are several subsystems that are defined by the task they perform. Our research focuses on the user query end, where the main research objectives in this field are focused on the querying, searching, and retrieval of the information [8]. The shortfall of a pure information retrieval system is that for the task we aim to cater for, a deeper analysis and understanding of the language text is required to provide the relevant information to the user, versus just looking into querying like keyword searches.

2.1.2 Information Extraction

As there was a rapid increase in the data that is available to humans, research interests moved towards the synthesis of this information, i.e. information extraction. This research deals with information management strategies that can be used to establish order in text, i.e. to extract structured information [10, 44]. Previously done through the use of information retrieval techniques, these systems are able to find and link relevant information while removing extraneous information that is considered irrelevant [10]. This field introduced an overlap of information extraction and NLP, where the objective of this field changed into a problem of NLP.

Initial research was done using a more rule-based approach where manual rules had to be designed and used. The aim here was to look at entities in the texts, the relationship between entities, and any other attributes that can be used to describe an entity. As the scope of information extraction increases, these methods have become more vulnerable and have led to the development of statistical learning approaches through generative and conditional models [44]. Overall, some of these approaches still fall short in understanding text.

2.1.3 Question Answering - An Application of Information Retrieval and Information Extraction

There are three basic approaches that can be used for automated QA : question templates, NLP and information retrieval. Depending on the contextual application of the use case, these different approaches are appropriate [2]. The question templates approach uses pattern matching and is known as template-based QA. It bases its intelligence on a collection of question templates that have been manually collected. Based on the study use case, this approach does not meet the desired objective of the research. This approach tends to convert the QA task into more of a classification problem where the question needs to be assigned to the most appropriate template to get the result. When analysing the problem presented in this study, linguistic intuition is required to understand the question and provide a comprehensive answer. This problem can be solved with the NLP approach [6].

2.2 Agro-information Question Answering Systems

The task of QA and the systems associated with it have been studied extensively and various advances have been made in the field [5]. However, research is limited to academia and is not easily transferable to industrial applications. Although these systems have proved beneficial for human interaction with information systems, there is still limited research on how such systems perform in a domain-specific scenario[21].

The development of agricultural QA systems has been of interest in several research studies. However, the study of what NLP techniques can be adapted for this domain has been limited. [11, 51] investigated the use of NLP to process questions/queries to make them understandable to be executed by a machine. The NLP tasks of concern is mainly the use of part-of-speech tagging and dependency trees to process the questions. An extended use of NLP was presented in [14] where a custom domain-specific Named Entity Recogniser was included in the system. Although there are overlaps between the point of focus for these studies, the value derived is that the system showed improved performance when dealing with domain-specific data.

There is an evident correlation between the location application of research studies conducted in agricultural QA. In [11, 14, 51], the three studies focus on developing QA systems for farmers in rural India. Despite the focus of these papers, the data used in [11, 14] in these systems are mainly in English. In a country as India, the applicability of these systems would still need to be further studied in a multilingual context to cater for small-scale local farmers where a language barrier might exist.

Analysing the limited research done in the QA studies in agriculture, the use of adapted NLP has shown significance; however, these studies are limited in NLU tasks and a multilingual context. This study aims to look at domain-specific QA as a NLU task, and then extend this task to a cross-lingual setting, where low-resource languages are studied.

2.3 Natural Language Question Answering

The QA task requires that a system is able to understand a given question and the context on which the question is based. This has made QA a challenging task in understanding natural language. As is known, natural language is dynamic and therefore challenging. To address this problem, the objective of QA was changed to a data-driven objective, where instead of looking at the methods, the concentration is placed on the data.[17].

Throughout the formulation of this task, different approaches have been developed through research. At first, the dominant approach consisted of using a rule-based method. This involved creating a collection of patterns manually, based on heuristics, and then employing them to ascertain the solution. One tool that has been used is decision trees, which can provide a logical representation of the linguistic structure of text and mimic human understanding. By implementing rules and patterns, they can use grammatical semantics [17].

The fatal drawback of this method is the requirement to manually create patterns. This allowed for the introduction of a statistical approach using tools such as Support Vector Machine (SVM), Bayesian classifiers and maximum entropy models. The statistical approach aimed to predict the answer based on the data, emphasising the idea that the approach is data driven. However, this method requires that some hypothesis be formulated before building a model, as it sets the tone[17].

Building on the statistical approach, a self-learning element was introduced that involved the addition of machine learning. This approach allows the algorithm to understand the linguistic features. The more recent approach was the Deep learning approach, which has the added ability to process raw natural language data by learning the underlying features of the data using neural networks(RNNs) [17].

From 2018/2019, the concept of LLMs was introduced and became the go-to solution for many natural language understanding tasks. This introduced a new paradigm in NLP research: Pre-training and fine-tuning [26]. LLMs also known as transformers, were designed to predict the probability of subsequent words, taking into account the contextual information from the preceding words. This prediction was based on the calculation of the generative likelihood of a word sequence. Even though this was the primary objective of language models, some natural language processing problems can be applied to work with transformers by reformulating the problem as a text-to-text format.

2.4 A New NLP Paradigm - Pre-train and Fine-tune

The introduction of LLMs/transformers has introduced an overlap in the research between NLP and human performance. A language model that showed high performance for multiple NLP downstream tasks, including QA is Bidirectional Encoder Representations from Transformers (BERT). The paradigm on which these models are based is to pre-train the models on large amounts of unlabelled text and then fine-tune these models for the desired downstream task, with an additional output layer [35]. Initially, this model

showed State Of The Art (SOTA) results on various tasks, including both SQuAD[40, 41] datasets (benchmark QA datasets) where SQuADv1.1 was 93.2 and SQuADv2.0 F1 is NLP 83.1 [12].

Adaptations to the basic transformer architecture led to different types of transformer being designed. One of these adaptations was based on the decoder part of the transformer [48] . These models are infamous for being used for generative tasks and have led to great advancements in NLP when it comes to more fluent and coherent text generation[35]. To achieve the SOTA results, the models were pre-trained on a larger corpus of text data using an unsupervised learning objective. The models were then tasked to predict the next token in a sequence given previous tokens, auto-regressive[35].

One such model is Generative Pre-trained Transformer (GPT) which was introduced in [38]. In [5], the aim of the investigation was to try and assess the quality of the outputs produced by these models through human evaluation. These models were systematically fine-tuned in several settings, including zero-shot, one-shot, and few-shot settings. These results were then used to measure the ability of humans to distinguish between synthetic data and real data [5]. Although there are notable limitations, it was generally hard to distinguish between the models' outputs and the real ones, defining the progress of the NLU by the models.

There are other variations in the transformer architecture [23, 39], and the commonality between these models is the highly influential ones when it comes to achieving SOTA results in NLP downstream tasks. These models are based on a new framework - ***Pre-train and fine-tune***. In order for these models to gain such results, the pre-training required the models to learn rich contextual representations of words, and thus a very large amount of data was required. Once this process is done, these transformers were then fine-tuned for different tasks. Thus, the idea of having a single task-agnostic model achieves strong natural language understanding [37, 39].

Due to the size and the nature of how the pre-training is performed on the models,

there is a limitation of the performance of the model through standard fine-tuning. With a study such as this where the focus of the data is domain-specific, the models need to be fine-tuned. Through fine-tuning , the language models are able to pick up patterns and terminology that are specific to the domain, therefore, generating high quality inputs. For domain-specific data, the amount of data is usually limited and relatively smaller than open-domain data.

The original fine-tuning objective for language models for QA is based on span extraction - the model predicts the span of text from a given context, which answers a question [12]. To improve the task-agnostic approach performed during the original pre-training, models such as SpanBERT [19] have been developed to align the pre-training objective with the task objective by performing the pre-training specifically for span extraction. Based on their results using the BERT model, there was a significant increase in the performance of the model (94.6% and 88% F1 on SQuAD 1.1 and 2.0). Despite this increase in performance, large training data sets were still required (SQuAD is approximately 100 000+ data points [41]).

Due to the limitation of the performance of the models through standard fine-tuning, a realistic setting needs to be considered where even if there is limited data, the model still performs well. One reason for the need to consider a setting where there is limited data is that when there is a drastic difference between the amount of data used during pre-training and fine-tuning, the performance of the model is degraded. This inspired research to make PLMs become few-shot learners. In few-shot learning, the model is only given a limited amount of data points to learn from.

In [42], the objectives of pre-training a model for QA were revisited, where the aim was to align the pre-training objectives with the fine-tuning objectives. This study showed that the discrepancies between the performance of the PLM were due to the fact that their pre-training objectives and the fine-tuning objectives do not align. Thus, by revisiting the pre-training objectives, the model's performance can be improved even if it is fine-tuning in a few-shot learning setting. With this new improvement, the data

used is open domain and it is not practical to revisit the pre-training objectives for a domain-specific dataset.

The opposite approach to this is prompt-based fine-tuning, also referred to as few-shot prompt learning, where the model is provided with a limited amount of data and fine-tuned with these examples, which are related to the desired domain or task. From 2021, there was another shift in the paradigm of NLP language models. This shift was based on the fact that instead of fine-tuning language models to specific NLP tasks, the objectives of the NLP task are based on the original pre-training objective. There is a reformulation of the downstream tasks to solve the problem in a similar way to the original pre-training that was performed on the language model [26]. This is done through a textual prompt which has been designed to predict the desired output, where the objective is to fill in the blank. This paradigm was developed to circumvent the need for large datasets, which are required for standard fine-tuning.

Several prompting strategies have been developed. The prompting strategy used depends on how the prompt template is designed in terms of the input and output [26]. Most of the prompting methods have been extensively tested in text classification and regression [15, 28, 46, 47]. The listed strategies have shown a significant improvement on the chosen NLP task compared to the standard fine-tuning procedures in a few-shot setting, a limited number of annotated examples that are fed into the model and used for the training. These methods showed that they can compete comparably well or even better than standard fine-tuning. The main objective of these prompts was to show that smaller models can be used to achieve comparable results compared to larger models. All of them have been shown to be approaches that are applicable for low-resource settings.

Although most of the techniques have been studied on text classification, variations have been made to them for other tasks such as text generation [45]. However, none were tested for QA. From these approaches, the only one which can be adapted to QA is Null prompting in [28], where this prompting method is not task-specific and does not include any natural language prompts. The other methods, however, are not easily applicable

to QA as tasks such as classification are based on the design of prompt templates based on patterns that are not available for QA.

In [7], the first prompting method that has been applied to the downstream task of QA is investigated. They introduced a framework where they converted the QA objective into a text-to-text framework so that the objective of the task matches that of the pre-training of the model. The prompt template used is to create the input as a concatenation of the question, a masked token which is the answer, and the context of the question. The two models considered were BART and T5, as these models' pre-training objectives align with the objective of multi-mask prediction, unlike BERT, which predicts a single mask token and requires that the answer length needs to be known prior, in order to predict multiple masks [7] .

In general, this method showed significant gains when it comes to training a language model in scenarios with limited data in a monolingual and multilingual setting for open domain data [41, 20, 9]. Although the multilingual setting was investigated, the investigation was limited to exploring whether the framework, as it is, will work for multilingual data, which only included 1 African language (Swahili).

2.5 Summary

The purpose of this chapter was to investigate the current research landscape for the various aspects of the field of Question Answering. The necessary background is provided to understand the scope of the field. It further positions the relevance of this study with respect to the current research and justifies the chosen approaches that are investigated in this study. In contrast to the studies discussed in this chapter, this study aims to investigate prompt-based methods on a novel dataset that is not restricted to being monolingual.

Chapter 3

Technical Background

This chapter provides the background information necessary to understand the technical aspects of the subsequent chapters. Following the objectives of this study, different decisions are made based on the information provided in this chapter. For the first main objective of this research, the considerations and tools that are needed to create the final QA dataset is explored. For the second main objective, the different models and model tuning techniques are discussed. A more in-depth discussion is conducted on the fundamentals of the fixed-prompt LM techniques, as they form the basis of the main experiment carried out. Slight modifications are made to suit the NLP task of QA, to the original techniques. This means that the fundamentals of each of the approaches are maintained and thus need to be understood. A detailed overview of the following technical aspects is provided :

- Section 3.1 discuss several widely used benchmark monolingual and multilingual question answering datasets. From these datasets, we explore the qualities of these datasets.
- Section 3.2 provides a brief overview of the alignment of sentences and the details of the approach used.
- Section 3.3 details the different types of pre-trained language models that are considered in this study.

- Section 3.4 discusses the 2 main model fine-tuning techniques that have been implemented in this study. It goes into further detail of the different prompt-based fine-tuning.
- Section 3.5 summarises the main concepts discussed in this chapter.

3.1 Defining the Question Answering task

With increased research on the NLP approach to question answering, several benchmark datasets have been created for the purpose of the development and evaluation of question answering systems. When exploring the main attributes of these datasets, a dataset of adequate quality can be developed. The datasets can be classified in several ways according to the characteristics they have.

3.1.1 Open vs. Closed Domain

A QA dataset can be defined by domain constraints that are placed on the information the data is based on. Open domain question answering refers to a dataset where there are no restrictions on the domain that is covered in the dataset. For closed domain, the data is based on a predefined domain, such as agriculture. Most of the datasets that are available are open domain datasets. SQuAD [41], TriviaQA [20], WikiQA [50], and Natural Questions [22]. A common source that is used to create these datasets is Wikipedia. Despite this, each of these are different from each other in terms of contextual information and how they are processed.

3.1.2 Generative vs. Extractive Question Answering

With these datasets[20, 22, 41, 50] advances in research have been made in both extractive and generative question answering. The fundamental difference between these two question answering tasks is that:

- Generative question answering tasks the model to generate an answer to a question based on the context that it is provide. This task can be performed with models such as GPT.

- Extractive question answering tasks the model to extract an answer to a question based on the context that it is provided. The final segment of text that is returned as an answer is supposed to be verbatim according to the context of the question. A model that has been fine-tuned to perform this task is BERT.

Here, for both the defined QA, there are three data features which up a data point: the context, the question, and the answer. An example of both these tasks is provided in 3.1 with the defined data attributes.

Context: Dry beans are so named because they are normally left on the plant until the pods have dried. The entire plant is then pulled up, placed in the shade where possible and allowed to dry for an additional one to two weeks. The dried pods are then split up and the beans removed.

Question: Why are dry beans named as such ?

Extractive Question Answering
Answer: they are normally left on the plant until the pods have dried

Generative Question Answering
Answer : Dry beans are named as such because they are harvested when the pods and the beans inside have dried out completely. Unlike green beans, which are picked while still fresh and tender, dry beans are left on the plant until the pods become dry and brittle

Figure 3.1: An example of the attributes of a data instance for the different QA tasks based on an article published by Pula Imvula on May 2023[2]. ChatGPT model was used to generate the generative answer.

3.1.3 Monolingual, Cross-lingual and Multilingual

With language-specific datasets, there are several key distinctions that define the evaluation setting of the question answering task.

Monolingual

The data that is used to train and evaluate the model is in the same language (e.g. the TyDiQA [9]). The context, questions, and answers that make up the data point are all in the same language, thus being monolingual.

Cross-lingual

The data used to train and evaluate the model is in multiple languages. However, the context, questions, and answers that make up the data point are in different languages(e.g. MLQA [24]). An example of this is that the question is in English and the context is in a different language. In such a case, the answer that is returned can be either in English or in the other language.

Multilingual

The data used to train and evaluate the model is in multiple languages. However, the question is in one language, which is known as the target language, and the context of the data point is available in multiple languages (e.g., CORA [4]). In this case, the answer returned is restricted to the target language.

3.2 Parallel Sentence Alignment

Sentence alignment can be defined as the process of matching sentences from what is defined as a source language to the equivalent translations in a target language. These matched sentence pairs are meant to be translations of each other. This process is widely used on text data to create parallel corpora in different languages for tasks such as machine translation. Various methods can be used to perform this, including the use of multilingual embeddings. The embedding is able to map the sentences into a vector space which is then used to identify sentences that are semantically similar.

A common embedding that has been used is the LASER toolkit [3] released by Facebook. However, a newer method has been proposed in [1], which has shown superior performance for sentence alignment of LRLs, compared to LASER. A multilingual embedding provided by CoHere [3] is used. Once the sentences are mapped into the vector space, a simple nearest-neighbour approach is used to align the most likely sentences.

[3] https://docs.cohere.com/docs/multilingual-language-models

3.3 Pre-trained Language Models

Pre-trained language models are grouped and defined by their architecture, their pre-training objectives, and any designs that are specific to an NLP task.

3.3.1 Masked Language Model

This language model architecture is based on only the encoder part of the transformer architecture in [48]. These models are pre-trained based on a contrastive task known as masked language modelling [35]. This is when the model is required to predict the missing tokens, and this enables these models to learn rich contextual representation of words. One such model is *BERT* [12] and *RoBERTa* [27].

3.3.2 Left-to-Right

These models are adapted based on only the decoder part of the transformer [48]. These models are famous for being used for generative tasks and have led to great advancements of NLP when in comes to more fluent and coherent text generation [35]. In order to achieve their SOTA results, the models are pre-trained on a larger corpus of text data using an unsupervised learning objective. The models are then tasked to predict the next token in a sequence given previous tokens, that is, autoregressive [35]. GPT [38] is an example of such a transformer.

3.3.3 Encoder-decoder

These types of transformers are based on the Sequence to Sequence (Seq2seq) objective where there is a transformation from one sequence to another sequence. This is particularly useful when it comes to tasks where the input and output are not necessarily of the same length and are a sequence. One such model is *BART* [23], which is a denoised auto-encoder for this task. The pre-training approach is for the model to learn how to map corrupted documents(documents with added noise) to the original document. This approach has proven to perform just as well as RoBERTa[27]. It generalises BERT and

GPT where there is a combination of bidirectional and auto-regressive transformers [23]. Another adaptation of this type of Transformer is the *T5* model which is a text-to-text model [39].

3.4 Domain Adaptation

PLMs can be adapted and trained in various downstream tasks such as text classification and question answering for domain-specific data. To do this, the model goes through model fine-tuning. To determine which techniques to use, two types of parameters are considered important in the design decision. These are (i) the PLMs and (ii) the prompting.

3.4.1 Traditional Fine-tuning

Traditional fine-tuning can also be referred to as Promptless fine-tuning. The aim of this technique is to retrain the model on a specific dataset while maintaining the original objectives defined during the initial training of the model for the downstream task. In the case of extractive QA, models are trained with the objective of predicting the span of an answer. A span can be defined as the character positions of where the answer segment starts and ends in a given context. For PLMs, during initial training, the models are generally trained on large amounts of data, presenting a challenge when adapting the models to work on a limited text corpora [35].

3.4.2 Prompt-based Fine-tuning

This can also be referred to as fixed-prompt LM fine-tuning. The idea behind this method is to reformulate the NLP downstream task to solve the problem in a similar way as the original pre-training that was performed on the language model [26]. This is achieved through the usage of textual prompts, where the objective is changed to a *fill-in-the-blank* type of task, i.e., masked language modelling.

The textual prompt can be thought of as a set of instructions that are provided to guide the model on the task at hand. These instructions are used to 'prompt' the model to fill in the blank by predicting what the masked tokens of the input are. The reason for using such a method is that this method was originally developed to help circumvent the need for large datasets that are required for traditional fine-tuning. As such, the methods based on this fine-tuning are referred to as **few-shot** where a limited amount of annotated examples can be used to fine-tuning the model and still obtain comparable results.

The model is tasked with using the probability of the text input x itself $P(x;\theta)$, compared to standard supervised learning where the model predicts the output based on the probability of the input $P(y|x;\theta)$. In order to incorporate prompting, a prompting function can be defined and added to the text input as:

$$prompt\ x' = f_{prompt}(x) \tag{3.1}$$

Where :

- x is the original text input
- prompt x' is the new textual input

After the application of the prompting function to the input, an additional element called a template is included. A template refers to the prompt that is appended to the inputs. The model then searches for the highest scoring output that maximises the score.

3.4.3 Prompting Strategy

Based on the above-mentioned definition, when applying this new objective, the following design considerations need to be made :

- the choice of the pre-trained language model
- the training strategy that is considered
- the template design of the prompt.

The prompting strategy used depends on how the prompt template is designed in terms of input and output [26]. Most of the prompting methods have been extensively tested on text classification and regression. The listed strategies have shown a significant improvement in their chosen NLP tasks compared to standard fine-tuning procedures. These methods showed that they can compete comparably well or even between as compared to standard fine-tuning, thus making them suitable for a limited low-resource language dataset.

FewshotQA

This is one of the first prompting methods that have been applied to the downstream task of Question Answering. They introduced a framework where they converted the QA objective into a text-to-text framework so that the objective of the task matches that of the pre-training of the model. The prompting template used is creating the input as a concatenation of the question, a masked token which is the answer, and the context of the question. The two models which were considered were BART and T5, as these two models pre-training objective line up with the objective of multi-mask prediction as compared to BERT which predicts a single mask token and the answer length need to be known prior in order to predict multiple masks. In general, this method showed significant gains when it comes to training a language model in scenarios with limited data in a monolingual and multilingual setting [7].

Null Prompts

In [28], they explored the idea of considerably reducing the need for prompt engineering by testing a concept known as null prompting. In null prompting, the input is simplified as a simple concatenation of the input and the mask token without any natural language/discrete prompts. This method is task-agnostic, meaning that this method can be used for a multitude of NLP tasks without any task-specific templates.

Better Few-shot Fine-tuning of Language Models (LM BFF)

Here, they propose the idea of not just prompting, but including a demonstration of the task into each context. By dynamically and selectively choosing the example that

is incorporated, significant results were obtained. This method is also task-agnostic and only requires minimal assumptions to be made for any NLP task. These demonstrations are included as part of the input context for the annotated data. This method was strictly tested on classification and regression [15]. This method can be classified as Prompt Augmentation.

3.5 Summary

A detailed discussion was conducted on the various aspects that contribute to the definition of the QA task. This included looking at what open domain versus closed domain is and what the difference is between extractive and generative QA. Another aspect that was defined was the language-specific characteristics of a dataset that need to be considered in the evaluation setting of the task. The alignment method was then discussed to create parallel multilingual datasets. The main concept of the chapter was to discuss in detail the different PLMs and fine-tuning methods that are considered in this study. The contents of this chapter provide the reader with a comprehensive overview of the technical background necessary to understand the rest of the chapters that follow.

Chapter 4

Data Pre-processing

There are several great sources of agricultural information, such as books and the Internet. However, the information from these sources is generally unstructured and broad, thus making it difficult to get specific information in a timely manner. Due to this fact, as highlighted in previously studied QA systems, there is still a need for more robust domain-specific systems [51]. One of the objectives of this study is to create a South African low-resource multilingual agricultural text dataset. The first step to do this is the data collection of the necessary publicly available textual information. This chapter aims to provide the reader with a detailed explanation of the steps followed to create and handle the dataset. In addition to this, it aims to provide some understanding of the final data and their properties. The multiple steps that follow are discussed in the following sections:

- Section 4.1 outlines the properties that are required for the final dataset, the details on the source that us used to gather the raw data and provides a summary of the final languages that are selected.
- Section 4.2 explains the process followed to gather the collection of the articles that are used as raw data.
- Section 4.3 describes the pipeline to create the final parallel article dataset. This included some of the considerations and processes that are implemented to standardise and manage the quality of the data prior to the data annotation process.

- Section 4.4 provides the details of the quality control that is performed after the data preparation step. This includes an outline of the manual review that has been completed.
- Section 4.5 gives a summary of how the final data is structured and distributed for the different languages.
- Section 4.6 provides a brief summary of all the steps that are discussed in this chapter to create the final data that is used for the next chapter.

4.1 Data Overview

Prior to collecting any data, it is important to outline some specifications that are required for the final resultant dataset. This is discussed in detail and defined as the data properties. From the data properties, an ideal source of information can be identified, and the required information can be refined. This data source is then used to determine the plausible South African languages that can be selected.

4.1.1 Data Properties

To successfully create a dataset for this thesis, several desired properties need to be outlined and discussed. Although there are several open-source benchmark datasets, none of them satisfy all the properties simultaneously.

Multilingual and Parallel Corpora

The dataset must consist of instances that are available in multiple languages. These instances also need to be parallel across the different languages. This will allow for a fairer comparison between languages. One source for such data is to get naturally parallel documents that are available. This means documents that are readily available in multiple languages. This is advantageous as high-quality datasets can be created without the need for manual translations.

Domain and Region Specific

The primary goal of this research is to cater for local farmers in Southern Africa. This means that the dataset needs to be domain-specific and region-specific. The main topic for which this dataset needs to addressed is agriculture, that is, being able to answer questions that revolve around agriculture. Farming around the world is diverse, and the decisions made about what can be grown depend on various factors. Physical factors, including climate, disease and pests, and terrain type, are among the factors that influence local farming practices. Therefore, it is important for the dataset to be tailored to the specific region, such as Southern Africa..

Extractive Question Answering

QA systems are designed to focus on providing accurate and concise answers based on a user's query or question. This involves the retrieval of relevant information based on the content of the question and how it is interpreted. An aspect of information retrieval is Extractive QA, where based on a document, a model aims to extract a minimal span of text which is returned as an answer to the question. An example of what the final data instance should look like can be seen in Figure 4.1

Context: Dry beans are so named because they are normally left on the plant until the pods have dried. The entire plant is then pulled up, placed in the shade where possible and allowed to dry for an additional one to two weeks. The dried pods are then split up and the beans removed.

Question: Why are dry beans named as such ?
Answer: they are normally left on the plant until the pods have dried

---

Figure 4.1: An example of the attributes of a data instance for extractive question answering based on an article published by Pula Imvula on May 2023[2]

4.1.2 Data Source

One source of data that satisfies the properties in 4.1.1 for Agro-information in South Africa is Pula Imvula. Pula Imvula is a monthly South African magazine designed to provide support to developing local farmers. The main objective is to provide information that helps the development process of farmers to help them become sustainable commercial farmers. It also provides useful information on agriculture production such as grain production [43]. The magazine is distributed in multiple South African languages including English, Zulu, Xhosa, Afrikaans, Sesotho, and Tswana. Currently, the articles that are published on the website range from October 2011 until the most recent month.

4.1.3 Language Selection

The South African languages were chosen by considering both the diversity of linguistic properties and practical factors. The first practical consideration is based on the multilingual embedding that is discussed in Section 3.2for the use of parallel sentence alignment, as this embedding is only trained on a handful of languages. The second practical consideration is the complexity of the linguistic structure of the language.

We explored an automated approach to create the annotated dataset. This means that for languages which have more complicated translations from English, present a problem as a more manual approach is necessary to clean up and align the data. One South African language which is an example of this is Sesotho, where in the final chosen data source, one sentence in English was sometimes translated into multiple sentences. On a more fine-grained level, it was observed that a word in English resulted in multiple words in the language, which resulted in the aligner struggling. The final languages selected, their ISO-639-2 code and their linguistic categories are listed in Table 4.1.

Table 4.1: Summary of the information for each of the selected South African languages

Language	ISO-639-2 code	Language Family
English	eng	British English
Afrikaans	afr	Hollandic
isiXhosa	xho	Nguni
isiZulu	zul	Nguni

4.2 Data Acquisition Process

Pula Imvula provided articles in Zulu, Xhosa, Afrikaans, Sesotho, and Tswana in text file format (.txt). The time range of the articles spans from 2011 to 2019. These original articles were not evenly distributed, which means that not all of the articles were available in all languages. Since the original English articles are not provided, the articles were scraped from their website directly.

4.2.1 Web-based Extraction

From the website, the articles are published in two different formats - pdf (portable document format) and as online articles (HTML web pages). For the English articles, anything prior to May 2015 is published as pdfs only, and then everything after this is online articles and pdfs. The *request library*[3] in Python is used to get a list of the different links to the articles available online. This list is compiled from all linked web pages on the homepage of the website [4]. To simplify the extraction process, the following criteria were followed:

1. The idea behind a parallel corpus is to have the same documents available in different languages. This means that not all the English articles available on the website are necessary. Due to this reason, the time ranges for the articles are then selected according to the dates (month and year) that already exist in the articles

[3] https://pypi.org/project/requests/

[4] https://www.grainsa.co.za/farmer-development

in the other languages. This filters out English articles that are not available in another language.

2. Only articles available in HTML format are selected, as these are easier to process. This means that the only articles downloaded are from May 2015, since this is when the magazine started producing online articles.

From the selected articles, the content of the Web page is obtained using the *BeautifulSoup library*[5] in Python. For each of the web pages, the elements of interest are:

- the title of the article
- the date the article was published (month and year)
- the textual content of the page

These elements are stored in a text file for further processing. In general, there are 600 articles, and the number of articles in each language can be seen in Table 4.2.

Table 4.2: The final amount of articles collected for each language that fall between May 2015 until 2019

Language	Number of Articles
eng	202
afr	143
xho	114
zul	141

4.3 Data Preparation

To create the raw parallel data of the articles, a simple processing pipeline was designed. For each of the steps, a manual evaluation was completed to ensure the quality of the

[5] https://pypi.org/project/beautifulsoup4/

data. For this particular dataset, to create a cross-lingual question answering dataset, the selected languages need to be classified. In this study, English is known as the ***source language*** and all other languages are the ***target language/s***. This pipeline can be divided into two major steps. The first step was to align the articles available in the target languages with the articles in the source language. The second step was to then perform parallel sentence alignment to match up the sentences in the body of the articles with each other.

4.3.1 Document Matching

The title of the articles in the different target languages is translated into English using Google Translator through the *Translator library*[6]. For each of the translated titles of the articles, the cosine similarity was computed between the translated titles and the original English titles. The translated titles were then paired with the most likely English title according to the highest similarity score calculated. The only English titles that are considered for this matching are those that fall under the same publication date.

Using the score of each of these pairings for each language, they were ranked from the highest similarity score to the lowest. Subsequently, only the top 50 article pairs were selected for each language. From the resulting pairings, they were manually evaluated to verify the matches. Since all matches had some kind of similarity, according to the similarity score, complete mismatches can only be picked up manually. The matches are rated and grouped according to the categorises defined as follows:

- Full Match: The translated title matches or is extremely similar to the English title. This group indicates a very accurate match. The similarity score for this match is 0.8 or greater.
- Partial Match with High Similarity: The translated title closely resembles the English title, where there is a minor discrepancy in the wording returned by the translator. This discrepancy is usually a different word or 2 and has a relatively high similarity score. The similarity score for this match is between 0.6 and 0.8.

[6] https://pypi.org/project/translators/

- Partial Match with Low Similarity: The translated titles not only have some similarity to the English titles but also contain significant discrepancies. These discrepancies can be defined as a difference in phrases and not just a word in the translated title. At the core of it, the translated title still maintains the core meaning of the original English title. The similarity score for this match is between 0.4 and 0.6.

- Mismatch: The translated title does not have a similarity or minimal similarity to the English title. The matches show that there is a lack of correspondence between the 2 titles and indicate that there was a failure in the matching process. These mismatches have a similarity score below 0.4. Some exceptions are also manually determined.

Extensive examples for each of the languages can be found in the Appendix A. The final distribution of the matches according to the different categories can be seen in figure 4.2. From the manual assessment of the matches, no mismatches were detected.

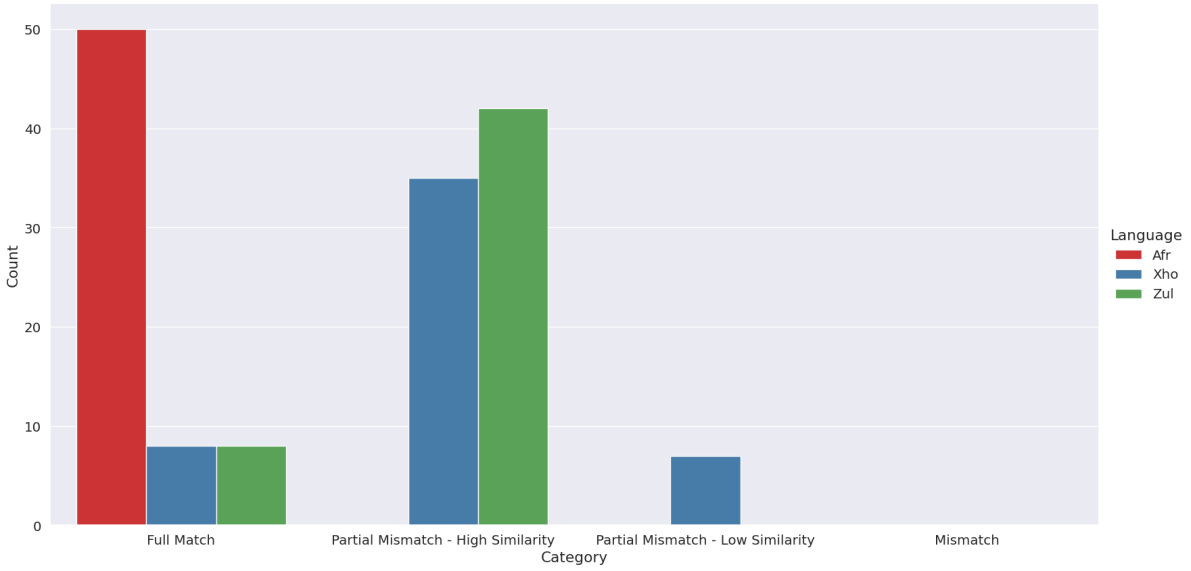

Figure 4.2: The distribution of the quality of the matches between the original English titles and the translations for the different languages

4.3.2 Pre-processing

We performed pre-processing on the articles to standardise the format across all the languages and ensure the quality of the data prior to the annotation process. First, to clean up the data, we follow the steps below for the individual paragraphs from each article:

- Removal of leading and trailing spaces.
- Standardisation tabs and multiple spaces, by ensuring that there is only one space between words.
- Replacement of non-ASCII punctuation with equivalent ASCII punctuation, such as apostrophes. These characters are not removed, as in certain paragraphs they are used to indicate things such as quotations, which is important for reading comprehension tasks.
- Replacement of Unicode characters with the equivalent ASCII characters. This included standardising accented characters present in Afrikaans. Some of these characters are *é, ê, ë, ï, ö, ó*, which are replaced with the ASCII letter equivalent. Unlike languages such as Latin where some Unicode characters need to be replaced with English word equivalents, the diacritic is used to place emphasis in a given word, thus the meanings of the words are still maintained after the replacement.
- Removal of Unicode characters such as bullet points and other symbols that are not listed in the above-mentioned pre-processing step.
- Concatenation of strings to form full sentences; this is mainly required for the scraped data, as the format of the sentences is not standardised.
- Full paragraphs are required for the data annotation process. To ensure that full articles are formed, the subheadings of the articles are concatenated into the trailing paragraph.
- For bulleted lists in the articles, each of the items of the list is grouped together to form a paragraph.

From all the articles, the important information that is needed is the title of the article, the date of the article, and the textual body of the article. In order to standardise the format of the articles, the extra information was removed from the articles. Figure 4.3 shows an example of this additional information that is removed. This information included :

- The word count of the articles.
- The captions of photos, graphs, and tables as these are not part of text derived from the scraped articles.
- Tables which are manually removed from all articles not in English.
- Instructions are listed in the articles.
- Contact details and details of the author of the articles.
- The publication edition and the section the article falls under, according to the website, for the scraped articles.

```
Datura (Maize) (710 words)                    Afrikaans Pula Feb 2016
Instructions:
• Pull (exploding) quotes: (Highlight)
• Total Photos:

Folio strap:
Onkruidbeheer

Byline:
Artikel verskaf deur Gavin Mathews, Baccalaureus in Omgewingsbestuur. Vir meer inligting, stuur 'n e-pos na gavmat@gmail.com.
Captions:
Foto 1: Die sade van die Daturaplant is uiters giftig.
Foto 2: Die Daturaplant is algemeen bekend as Olieboom.
Foto 3: 'n Goeie onkruidbeheerprogram is noodsaaklik om jou opbrengs te beskerm.

Publication:
 April 2019
Section:
 Pula/Imvula
```

Figure 4.3: An example of the additional article information that is removed during pre-processing.

The final pre-processed articles are then structured as follows.

- Line 1 : Title (in the respective language)

- Line 2 : Date (month and year)
- Line 3 : empty line
- From Line 4 : body of the article

An example of the final pre-processed article can be seen in figure 4.4. The above-mentioned two lists are not exhaustive for all the articles, but based on the final sample of the articles that are selected. For future work, there may be a need for additional pre-processing steps. All pre-processed articles are then renamed according to the title of article in the respective language, followed by the publication date.

```
Crop rotation and the use of inoculants for soybean seed
July 2016

Farmers are becoming more aware of the importance to implement crop rotation into their management practices.
Crop rotation is the practice where farmers rotate the crop planted in a certain land from year to year; this is the opposite of monocropping
where a farmer will plant the same crop every year.


So, for example, a farmer may plant two crops for two years in a row such as maize and then in the third year they will rotate by planting a
second crop such as soybeans.
There are many different advantages for the farmer in doing this which I will outline in this article.


Firstly, different crops require different nutrients from the soil.
If we constantly plant the same crop every year we will continually be depleting the same nutrients.
If on the other hand, we rotate the crop that we plant, the different plants will take advantage of varying nutrients which will give others a
chance to rejuvenate.
This is also why it is advantageous to plant a legume crop in the rotation which will not only take advantage of different nutrients than maize,
but it will also fix nitrogen back into the soil for the following year's crop to make use of.
Nitrogen is required especially by maize to produce a high yielding crop.
```

Figure 4.4: An example of how the articles are structured after pre-processing.

4.3.3 Sentence Alignment

In this step, we leveraged the naturally written articles available from Pula Imvula and avoid translation of the textual contents of the articles. In our approach, we aimed to have 2-way parallel sentences for each of the documents. For each of the target languages, we independently aligned them with English, forming two-way parallel sentences. The process described in **??** using the CoHere multilingual embedding [7] is followed. The results from this alignment can be used in future work to create N-way parallel sentences for more cross-lingual research.

[7] https://docs.cohere.com/docs/multilingual-language-models

4.3.4 Context Curation

Before creating parallel paragraphs, some manual pre-processing was completed on the selected articles in English. This pre-processing was based on observations made, included removing lines which contained a single sentence, and grouping it with a paragraph above or below it. This was done to ensure that the paragraphs in the articles are at least 2 sentences or more for the annotation process. Revisiting one of the properties of the desired final dataset, we are required to create an extractive question answering dataset.

As described in Section 3.1 of extractive question answering models, the objective is to retrieve an answer to a question based on a given text. This text can be a full document (an entire article) or individual paragraphs of the article. This text is referred to as the context, i.e. it provides the context to an answer of a question. We decided to use individual paragraphs from the articles as the contexts for question-answer pairs. To create two-way parallel contexts, the structure of each context is based on the English article structure. For each of the paragraphs in the English article, the parallel sentences from the target language are grouped accordingly to recreate a full paragraph.

4.4 Quality Control and Management

Since an automated approach is used to create the data prior to the annotation process, manual review is required to ensure the quality of the data. The primary characteristic of the data that requires evaluation is the accuracy of the sentence alignment. Additionally, it is important to manually identify the cause of any discrepancies that may be observed. In order to evaluate the data, a random subset of data instances are selected for each of the target languages (afr, xho, and zul). Since this review is time-consuming, the size of the subsets was limited to 10% of the total instances for the specific language.

From this subset of instances, the contexts in the respective languages are translated to English and then manually rated according to three categories:

- Full match: The translated contexts match completely the original English context.

This means that all the sentences that make up the paragraphs match or are extremely similar.

- Partial Match: This means that there is a minor discrepancy between the translated contexts and the original English context. This minor discrepancy is identified as one or two sentences that are not similar in the overall context. However, even with these discrepancies, the content of the contexts is aligned.
- Mismatch: The translated contexts do not have any similarity to the original context. This means that overall the contexts do not align.

From this initial manual review, it can be seen that overall for Afrikaans (91.6% full matches), Zulu (79.49% full matches) and Xhosa (73.68% full matches), the embedding performed relatively well. In general, all discrepancies were then further reviewed and analysed to isolate the cause and improve the quality of the data.

Two main causes are identified in all languages. The first cause is that for some of the articles, the articles in the target language contained extra information which was not initially picked up in the pre-processing. For these articles, there was a significant difference in the length of the articles in terms of the number of sentences contained in the overall articles. This large difference in the article lengths introduces biases and noise, where the noise makes it trickier for the embedding to map out with the extract same sentences as the content is still about the same content and is very similar.

Secondly, for some of the articles, there is a discrepancy between the format of the sentences and paragraphs. These discrepancies are brought up by things such as missing punctuation, where two sentences are written as one sentence. This means that in one article there will be two sentences that will technically need to be mapped out to the same sentence. The problem introduced ambiguity, where there was a one-to-many relationship for some of the sentences.

In order to improve the quality of the data, an iterative process was completed in which for each of the languages the following steps were followed:

1. Articles that differ significantly in length are filtered and isolated.
2. From these articles, they are manually reviewed and if any changes are needed to be made to the format of the article, they are made.
3. Once the manual corrections are done, the sentences are then manually split (without using the automated sentence tokenisation route) to create the relative files needed for the sentence alignment process.
4. The results from the updated alignment are used to create the new contexts for the data annotation process.

From the manual evaluation, if major discrepancies occur between the articles, these articles are excluded. Overall, for each of the 3 languages, there is a total of 49 articles. There was a common article that was excluded from all 3 languages as the structure, and the content of the articles with the scraped article equivalent is a total mismatch.

4.5 Data Summary

Initially, across all languages, the corpus contained more than 100 articles for each respective language. After all the interim steps, there are now 49 articles for each of the languages excluding English. Each of these articles was then broken down into paragraphs to form the context that is needed for the data annotation step. The purpose of this data summary is to consolidate all the steps that were performed and give an overview of the final dataset that will be used for the data annotation steps.

4.5.1 Data Structure

For each of the languages, the final data instance compromises of (i) the date and English title of the article and (ii) the final context in both English and the target language. A detailed example of what the final data point looks like is provided in Figure 4.5 for each set of languages.

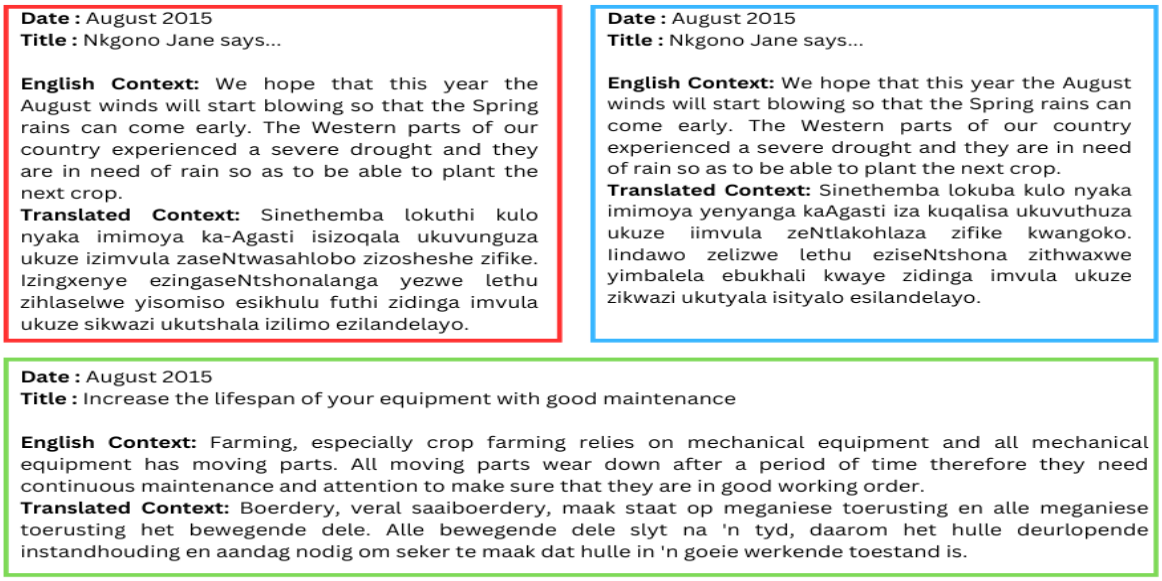

Figure 4.5: An example of how the final contexts are structured. The text outlined in blue is an example from Xhosa, the red is an example from Zulu and green is an example from Afrikaans

4.5.2 Language Distribution

In total, there are 712 unique contexts in English. These contexts were concatenated from the different context pairs in the different target languages. To understand the distribution of these contexts for each of the target languages, the number of unique contexts is provided in Table 4.3

Table 4.3: The final amount of contexts that have been prepared for each language

Language	Number of context
afr	462
xho	360
zul	359

4.5.3 Final Quality Assessment

The final contexts were manually reviewed again in the same manner as mentioned in Section 4.4. From this final review, no major mismatches or discrepancies were detected between the English and target language data.

4.6 Summary

In this chapter, the text corpora used in this study is introduced. To create the final dataset, the articles were obtained directly from Pula Imvula and scraped from their website. An overview of the desired data characteristics was provided to support the choices made during the creation of the text corpora. The collected raw data was then prepared through a simple pipeline for the next chapters. This pipeline included matching the articles in different languages together, aligning the content of each of the articles, and creating the final contexts for the final QA dataset. The method used to ensure the data quality of the final dataset is also provided. The next chapter provides a more in-depth analysis of the contents of the resultant dataset.

Chapter 5

Exploratory Data Analysis

In the previous chapter, Chapter 4, the corpora of interest was introduced. A detailed discussion of how the data was collected and how the data was processed is provided. The final structure of this data is then summarized. Once the collection process is complete, the data is intended to be utilized for the purpose of data annotation. Before doing this, it is important to understand the different attributes of the data and the content of the data. By performing exploratory data analysis, the aim is to get more insight into the data, and the quality of the data before taking any further steps. This can be done through exploratory data analysis. Exploratory data analysis uses standard text analysis techniques to provide insight on the structural and statistical properties of the resultant dataset. The information and observations made from this data can then be used and leveraged in other steps. The data analysis is structured as follows :

- Section 5.1, which provides details of the structure of the data through the use of statistical analysis.
- Section 5.2 gives an overview of the contents of the data in terms of topic modelling and the language complexity of the data in the source language.
- Section 5.3 analyses the qualities and relationship between the data in all languages using correlation analysis.
- Section 5.4 gives a detailed summary of the main observations that are made for all the analyses performed.

5.1 Descriptive Structural Analysis

For each of the languages considered in this study, we analysed the general distribution of the data at different levels of granularity: at the article level and the context level. Table 5.1 shows a summary of the general statistical structure of the data collected. The length is defined as the word count of the text after being tokenized. Since this study focusses on creating a dataset to be used in the final experimentation, it is important to understand the statistics of the corpus. From Table 5.1, it can be seen that the average length of the context of the article, the minimum and maximum values for the context are higher for English and significantly lower for Zulu and Xhosa.

Table 5.1: Summary of context data that was created in Section 4.3

	eng	**afr**	**xho**	**zul**
Total Number of Contexts	712	462	360	359
Average Article Length	804.88	854.59	568.57	557.53
Average Context Length	96.09	90.64	77.39	76.10
Minimum Context Length	14	18	16	10
Maximum Context Length	405	394	233	291

Using Figure 5.1, further analysis is performed by comparing the matched context in the different languages with the equivalent context in the source language English. When some of the contexts at these values are analysed, some of the linguistic characteristics of the different languages can be derived. For languages such as Zulu and Xhosa, some of the cases where in English they make use of stopwords or prepositions, in these languages, there may not be an exact equivalent direct translation, but rather a semantic translation. When compared to Afrikaans, however, there are direct translations for some of the prepositions and stopwords. This means that for the subsequent steps in the data annotations, the answers in the target languages should be anticipated to be shorter for some, if not all, entries in Zulu and Xhosa.

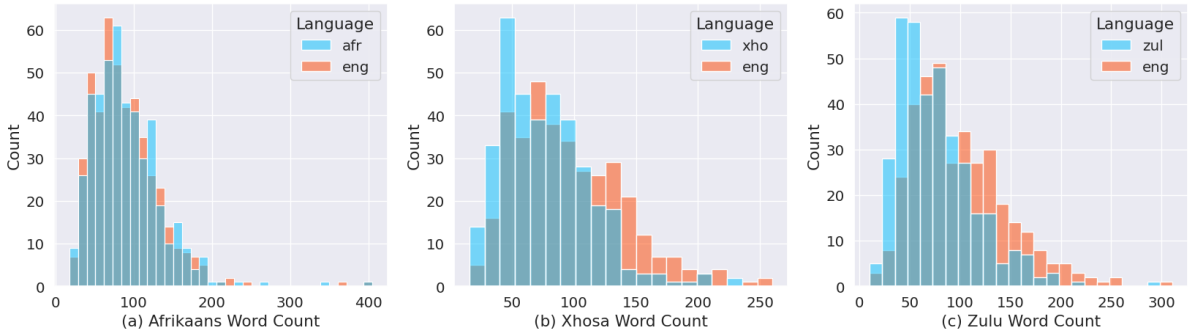

Figure 5.1: The word distribution of each of the languages, compared to the equivalent English data. This is done at the level of the contexts and not the full articles

5.2 Exploring the Contents of the Data

Highlighting one of the main objectives of this study, the dataset that is created needs to be based on agro-information. It also needs to be able to address pertinent topics that will be useful to small-scale farmers. To confirm whether or not the collected data will be able to achieve this objective, naive NLP techniques can be used to grasp the main content of the articles collected in Section 4.2 and the complexity of these articles. By understanding these 2 properties, the expectations out of the annotation process can be pre-emptied.

5.2.1 Topic Modelling

As mentioned in Section 4.1, we require that the data is about agriculture. We assumed that most of the articles, if not all, address topics related to agriculture. From the initial manual inspection, there are some articles which are based on things like interviews, where some of the contexts in those articles are not necessarily about farming, but personal experiences of the interviewee. To support one of the research objectives, simple NLP techniques can be used.

One approach to get a high level understanding of the data is by considering term frequency through word clouds. Word clouds are a visualisation method which shows the most frequently occurring words in a text dataset. The size of the word is determined

by how often the word occurs; the larger the word, the more frequent it occurs.

A word cloud is generated in figure 5.2, which provides a high-level view of the topics addressed in the articles. From this figure, the emerging topics that are addressed in the articles can be seen from the most occurring terms. Some of these topics include :

- Crops such as canola, maize and dry beans.
- Farming techniques such as weed control, soil analysis, and crop rotation.
- Financial advice, such as funding.
- Essential tools such as technology.
- Development of farmers' progress through initiatives like the creation of business plans and marketing strategies, as well as participation in training programmes.

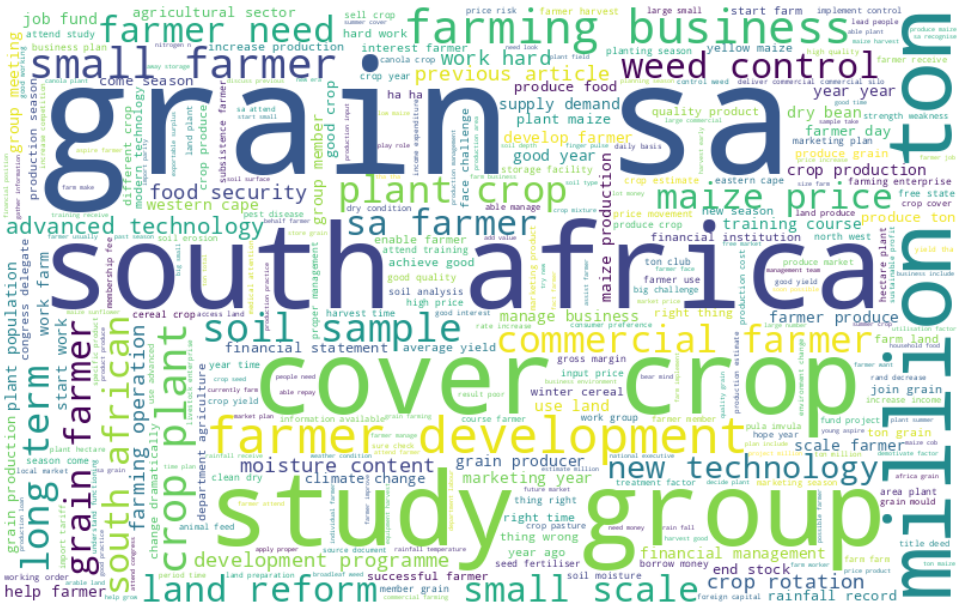

Figure 5.2: Word cloud of all the articles in the annotated dataset.

5.2.2 Corpus Complexity

There are two ways in which corpus complexity can be assessed : readability and lexical richness. By assessing the complexity of the data, it can guide the prompting that is used for the next step as well as the expectation to be had for the resulting annotated dataset in Chapter 6. Besides this, we can also get information about the quality and appropriateness of the text for its intended use, i.e. for small-scale farming.

Lexical richness of a text can be calculated using the Type-Token ratio(TTR). This value can be obtained by taking the sum of the different words that occur in the text and dividing it by the total number of words. To get this value for the dataset, the TTR is calculated for each of the contexts, and then the average is obtained. An average score of **0.72** is obtained for this data using the *Lexical Richness* library [1].

From the value obtained, it can be interpreted that there is a high degree of lexical variation. We determined that the vocabulary used in the text is sufficient. This indicates that the articles are well-written and can serve as a solid foundation for developing more effective and engaging content. This also suggests that the data is more detailed about the agriculture topics that are discussed.

Another way to grasp how complex a text can be is to assess using the readability of the text. This score is employed to evaluate the comprehensibility of the text by predicting the grade level necessary to comprehend it. As was calculated for the Lexical Diversity, the average score is obtained across all the articles. The *Readability metrics* library [2] is used and a score of **10.875** is obtained which can be rounded to a grade level of 11 against the US education system. From this value, we concluded that the information in the articles is comprehensive and further emphasised the conclusion made based on the lexical diversity.

[1] https://pypi.org/project/lexicalrichness/

[2] https://pypi.org/project/py-readability-metrics/

5.3 Cross-lingual Analysis

Traditionally, when a multilingual dataset is created, there are two common paths which are followed. The first path is to create the dataset in a source language such as English and then translate the data into different languages using human translators. This method usually results in high-quality data as the annotations are completed by people who understand the language. The second path is to use a parallel data corpus. In this case, the data which is used to create the final dataset is sourced for the different languages and then the equivalent texts are then matched up or aligned. Once the data is aligned or matched up, an additional step which can reinforce the quality of the data is the use of people who speak the language to confirm the alignments.

We explore the idea of using correlation analysis to confirm the quality of the main alignments in addition to the results of the manual review in Section 4.4. In simple terms, we wanted to assess how the resultant texts are related to each other according to the structural features of the text. The expectation for all three target languages is that there is a high positive correlation between the lengths of the parallel aligned contexts. The length can be defined as the word count. These expectations are confirmed in Figure 5.3, where for each of the languages the correlation coefficient to English is greater than **0.9**. This coefficient is based on the Pearson correlation coefficient, which is a widely used method to calculate linear correlation.

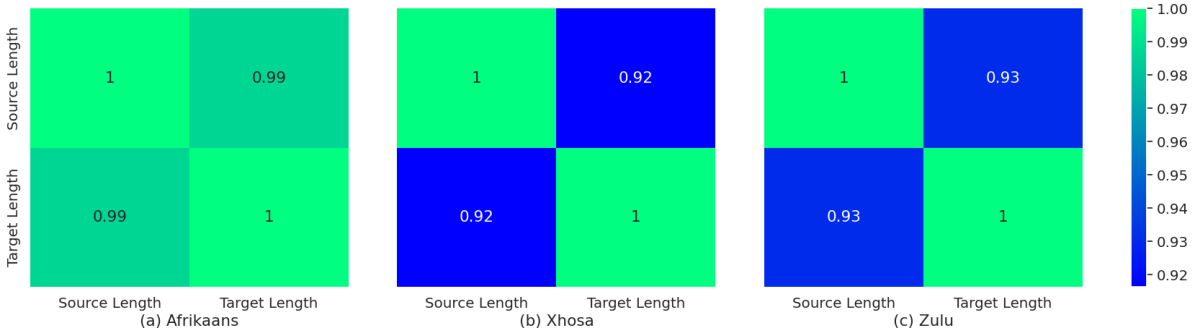

Figure 5.3: The correlation between the context lengths for the different target languages to the source language(English).

As mentioned in Section 5.1, some deductions were made about differences in the linguistic characteristics of the different target languages (afr, xho, zul). To reiterate and support these observations, correlation analysis was performed on the basis of the TTR of the different contexts in the respective languages. The results are provided in Figure 5.4. From these results it can be seen that Afrikaans has the highest correlation, meaning that the translations of Afrikaans are more direct and literal translations. Comparing this result with those of Xhosa and Zulu, the correlation is low, emphasising that the translations are more semantic, i.e. meaning-based.

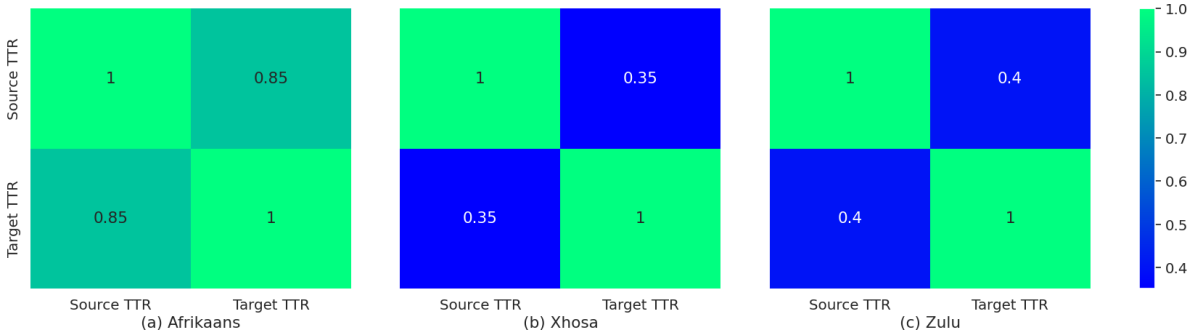

Figure 5.4: The correlation between the context Type-Token Ratio for the different target languages to the source language(English).

5.4 Summary

The exploratory analysis performed in this chapter focused on providing information about the dataset created for data annotation. From the initial descriptive statistical analysis, the structural relationship between the data in the different languages is observed. In general, the data in Xhosa and Zulu are significantly shorter in terms of word length, compared to Afrikaans and English. This indicated the differences in the linguistic features of the data. This point was re-emphasised using correlation analysis where it was concluded that Zulu and Xhosa are more semantic based translations compared to Afrikaans, which are more literal translations.

Overall, the fulfilment of the research objective to create a dataset based on Agro-information in a South African context was confirmed through the topic analysis done in this chapter. All the main derived topics were based on farming in South Africa. In addition to this analysis, the quality of the data was confirmed to be comprehensive and more detailed based on the corpus complexity results.

Chapter 6

Data Annotation

Following the data collection done in Chapter 5, the next step is to use this data to create the final QA dataset. This is done in three steps that create the annotation pipeline: an automated data annotation technique, data filtering, and the cross-lingual dataset creation. The data annotation involves creating question and answer pairs from the raw dataset. Since a generative model is being explored as a tool to annotate the data, simple heuristic filtering is implemented to ensure the quality of the data from the model. Once the annotated data is filtered, a cross-lingual dataset is created where the equivalent answers are extracted for the different languages. To encapsulate the entire pipeline process, different aspects of the pipeline are discussed in the following sections :

- Section 6.1 provides a detailed outline of the design choices made for the annotation pipeline.
- Section 6.2 gives the details of the first step of the question-answer generations where key points are derived from the text to form the basis for the question answering generation.
- Section 6.3 describes the prompting that is used for the generation of questions and answers using the GPT model, in the source language.
- Section 6.4 explores methods to ensure the consistency of the annotated data. This includes looking at different attributes that have been explored in previous

literature to create a simple heuristic filtering method for the generated data and the retrieval of answers in the different target languages.

- Section 6.5 provides the results from the manual reviewing that is done to ensure the quality of the final dataset
- Section 6.6 is a summary of the all the steps that discussed in this chapter to get the final dataset.

6.1 Annotation Pipeline Design

To do the experimentation required to answer the research question, one of the objectives is to create an adequate dataset. To create this dataset, traditionally it is done through manual annotation. For manual annotations, annotators are sourced through professional companies or crowd-sourcing. This process involves asking the annotators to read the given context and from that context create questions. After creating these questions, annotators are tasked with answering the questions. These answers need to be directly derived from the context and written as is in the context. Once this is done, a data validation stage is added where the annotated data is evaluated and the final data points are selected based on the outcomes of this.

This process is usually expensive and time-consuming. With the recent emergence of large language models, these models can be explored to replace manual annotation with automated annotation. In this study, we explored the use of large language models and NLP techniques to create an automated data annotation pipeline. This pipeline is provided in Figure 6.1.

The first stage involved using the *GPT 3.5* (text-davinci-003) model by OpenAI [1] to generate the question answer pairs in the source language (English). In this stage, there are three stages that are followed, which are: keyword extraction, question generation, and answer generation. For each of these stages, the model is prompted, and then the

[1] https://platform.openai.com/docs/models/gpt-3-5

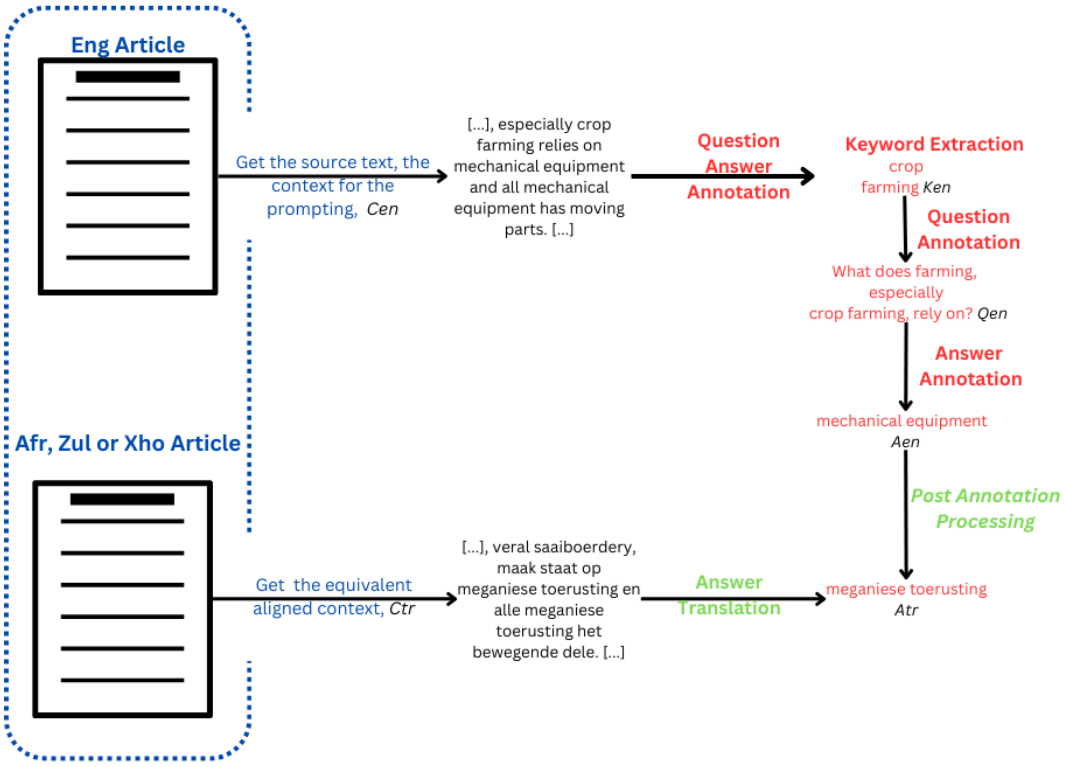

Figure 6.1: An overview of the different steps that are followed for the annotation process

final response is captured. The next stage is the post-annotation processing where the question and answer pairs go through several steps of validation and filtering to ensure the quality of the data. A simple heuristic approach is followed using NLP techniques to eliminate undesirable output from the GPT model. This approach is built on the limitations of model and undesirable QA dataset attributes based on previous literature [20, 22, 41, 50].

The last stage is to obtain the annotated data in the target languages (Afrikaans, Xhosa, and Zulu). For this, parallel alignment of the answers is done to get the equivalent English answers in the respective languages. The final output of the pipeline is a data point that contains the different variables presented in figure 6.1 (C_{en}, C_{tr}, Q_{en}, A_{en}, A_{tr}).

After this, a manual evaluation approach is used to verify the final data and determine the quality of the dataset.

6.1.1 Selection of Language Model

GPT [5] has recently become a point of significant interest due to the strong results achieved for NLU. Since these models have gained traction, extensive research has been conducted where the main aim is to explore what the capabilities are of these models. This research has resulted in knowing where these models perform well through experimentation and the limitations that are present when using such models. This model was selected based on the main results of [5] where in addition to the strong performance for the various tasks, the model was able to generate data that, when evaluated by human evaluators, the articles were difficult to distinguish from those written by humans.

6.1.2 Prompt Design

As in manual annotation of data, a comprehensive guideline needs to be provided to the annotators. This guideline is useful for providing the annotators with the theoretical background and the detailed instructions that will help them provide an accurate output. This guideline prevents ambiguity of the output and problematic output. Following this idea, a similar approach needs to be developed for the model to obtain high quality data. This guideline is provided to the model through prompting.

Since a GPT model is used for the annotation of the data, prompt design is a very important aspect of the annotation process. The output of the model is very dependent on how the prompt is structured. The factors considered for the prompt design are based on known GPT limitations that have been highlighted in the research. Some of these factors are as follows:

- Directness and Specificity: From different experiments, the models have been shown to be sensitive to the prompts. This means that minor variations can lead to a different output. Therefore, the prompts need to be straightforward but still as detailed as possible.

- Inclusion of examples: It has been shown that the few-shot scenario where several examples are provided in addition to the prompt does not translate to all tasks i.e., it does not improve the performance for all the tasks.
- Context limitation: Since the model is a generative model, it means that it synthesises text. This means that the scope of what the output should be based on should be limited to the data that is provided directly to model and not generated from anywhere.

For each of the steps which require prompting, an iterative approach is followed, where different variations of the factors mentioned above are tested to settle on the best prompt. This iterative process is done on a small subset of examples of the dataset, and then once the final prompt is decided upon, the process is scaled up to the entire dataset.

6.2 Keyword Extraction

The objective of this step is to prepare the data for the generation of questions and answers. The desired outputs are for an agro-information dataset that will assist farmers. To avoid generating general questions and answers, the main information must be extracted from the contexts that were curated in Section 4.3. The prompt is designed to identify the most informative keyword or key phrase in the text. The final prompt used is :

> *Given a block of text and the title of the text, first, read the text and establish the main topics of the text. Based on these topics, extract a list of key keywords or short phrases from the text that capture the topics. Return these key keywords and short phrases in a list that is in a concise format.*

A two-step approach is decided where the model first tries to identify the main topics of the text. This is done on the basis of the context and article title that are provided to the model. The title is included as it provides high-level information about what the article is about in its entirety. From these main topics, the appropriate key phrases are then selected. By basing these key phrases on the main topics, it prevents the model

from selecting information that might seem important in the context but not related to the underlying topics.

Through the iterative process, a few-shot scenario is also used to guide the model. Without the inclusion of examples, the model tended to return less comprehensive key phrases i.e., a word or 2 is left out, which for reading comprehension tasks would be considered significant. A complete prompt is provided in the appendix B.

Keywords are then split and individually paired with the corresponding context. These keywords are then processed before the next step. Some of the processing steps include the following :

1. Removal of empty strings that are considered as null values.
2. Removal of duplicates.
3. Removal of keywords which referred to a table, photo, or graph. Since no other article content is captured besides the text, the references to either table, photo, or graph can be considered as secondary information, which is not very informative of the main content of the context provided.

A simple string matching algorithm was implemented using the *Regex* library [2] search function as a validation step. This algorithm was used to verify that the keywords came directly from the context. For keywords that were not an exact match, three unique cases were noticed. Firstly, some of the keywords ended up coming from the title of the article and not from the context. In this case, these keywords were removed. Second, by manual observation the rest of the keywords were related to the main content of the context, but just a paraphrased version of a phrase in the context. Since at this stage, the exact wording of the keywords were not of concern but rather the meaning of the phrase, i.e. it just needed to communicate the same point. Third, some of the outputs produced by the model did not match the expected output format that was communicated to the model. For this case, these cases were manually corrected. This included cases where the

[2] https://docs.python.org/3/library/re.html

model returned a list of keywords in an improper format and outputs where the model states that it failed to generate a list of keywords.

6.3 Question and Answer Generation

For this stage of the pipeline, a sequential prompt is created. This means that the model is prompted to generate a question first, and then based on this question it is prompted to generate an answer. The outputs from the keyword extraction are used as the basis for the question generation. Keywords are meant to provide the fundamental information to ensure that the questions that are generated are topic specific. The main aspect of the prompt that is required for this stage of the annotation is to reinforce the context, i.e. remind the model to only produce questions and answers based on the context it is provided and the keyword. By reinforcing this, it ensures that the questions and answers are as relevant as possible. The final prompt for this is :

> *Given a context and a key phrase that highlights a main theme from this context, generate a relevant factual question where the answer to the question is based on the key phrase. From the generated question, provide a concise and direct answer to the question only using the information from the text. Ensure that the answer is an exact extract from the paragraph and is written as it is in the paragraph without any changes. Return the question and the answer for the corresponding text based on the provided examples.*

A few-shot scenario is also used for this step of the annotation, which is provided in Appendix B. It was noticed through the iterative process that for the generation of questions, the models tended to produce some complicated questions. In this case, complicated questions are those that require not only a particular phrase from the context but a more in-depth answer, where different key points from the context are combined together to answer the question adequately. For answer generation, the answer needs to be extracted and written as is in the context. Without examples, the model tended to produce more paraphrased answers than with examples. It also returns the answer as phrased in the context with minor inconsistencies, which required more manual changes such as :

- For some of the words, the GPT model returned the American English version of the word, as compared to British English used in the original context. An example of this is *meter* and *organization* was return, instead of *metre* and *organisation* respectively.
- For decimal points, the original articles indicated these as a comma between 2 numbers, whereas GPT returned them as a period between 2 numbers.
- For ranges presented in the original articles - such as *5mm to 7mm* , GPT returned these ranges as *5mm - 7mm* instead.
- GPT was not always case sensitive. This meant that for some of the answers, in the original context, these answers were capitalised, and in the GPT response they were all lowercase.
- GPT returned a shortened answer where the equivalent phrase of the context contained extra words, mainly stopwords.
- GPT rephrased sentences slightly by replacing a word or 2 with a word that has the same meaning. An example of this is in the original context the phrase was *produces risks* where the returned response is *producing risk*.

With the addition of examples, some of the inconsistencies were addressed, except the last two were then addressed during the post-annotation processing. At the end of the whole annotation process, there are 7038 data points where each of the data points is structured as in Table 6.1.

6.4 Post Annotation Processing

As with any annotation process, a post-annotation processing step needs to be added. Since an automated approach is used for the verification, there are three methods that are followed to refine and verify the outputs of the model. This step is important as it allowed us to verify that we met the objective of creating an extractive question answering dataset. Prior to any application of these methods, general cleaning up of the data is done, where duplicated entries and any null values are removed.

Table 6.1: An example of the different features of the final data point

Feature	Text
Title	Increase the lifespan of your equipment with good maintenance
English Context	Farming, especially crop farming relies on mechanical equipment and all mechanical equipment has moving parts. All moving parts wear down after a period of time therefore they need continuous maintenance and attention to make sure that they are in good working order.
Keyword	crop farming
Question	What type of farming relies on mechanical equipment ?
Answer	Farming, especially crop farming relies on mechanical equipment

6.4.1 Annotation Consistency Management

GPT is a generative model, and as such the outputs of the model needed to be validated to ensure the quality of the annotated outputs generated. One of the things that needed to be checked is the consistency of the annotation. This meant ensuring that the annotated questions and answers are consistent with the desired output, as illustrated by the prompt examples provided to the model as in the Appendix B. This makes the dataset more reliable.

To verify the annotation consistency for generated answers, we first checked if the answers generated are phrased and identical to a phrase (written word for word) in the given context, i.e., there is an alignment between answer and the source which is the context in this case. The *Regex* library is used to find identical matches in two strings. The generated answer is matched with the context; if a match is found, this means that the answer is extracted from the context, thus consistent with the prompt. After this process was completed, a manual extraction algorithm 6.1 was developed to deal with some of the inconsistencies mentioned in Section 6.3.

Initialise all variables
Tokenize the context into *sentences*
Tokenize the answer into *words*
for each *sentence s in the sentences*
    Check if *s* contains all the non-stopwords of *words*
    if *all words w are in the sentence* **do**
        Find the index of the starting and ending character for each *w*
        Find the smallest index *i* and the largest index *j*
        Create a sub-string *a* from *s* from *i* to *j*
    return *a*
return *None* if no match is found

Algorithm 6.1: The algorithm used to manual extract answers from a given context.

Once manual extraction was performed, if a match is still not found, these answers were removed. For majority of the entries which were removed, it meant that the question required a more complicated answer where multi-sentence reasoning is required. Multi-sentence reasoning in reading comprehension tasks means that to answer a question sufficiently, multiple sentences are needed to generate an answer. Such questions are more appropriate for generative question answering. Approximately **9.7%** of the data was filtered out and found to be inconsistent.

6.4.2 Question and Answer Filtering

To ensure the quality of the generated question-answer pairs, NLP techniques can be used to filter and validate output that is not standard. To guide what the filtering methods could be, three specific qualities were considered based on benchmark datasets [20, 41, 50]. These qualities are relevancy, redundancy, and ambiguity. For each of these qualities, different combinations are tested and a score is assigned. From these scores, a custom scoring system is implemented where, depending on the score, a data point is filtered out or kept.

Table 6.2: Summary of the criteria that is followed with the scoring system to filter out the QA pairs

Scenario	Criteria
Ambiguity	This is defined when there is a question based on the same context that is very similar or the same as another question. A cosine similarity score of a pair of questions of 0.85 or higher in conjunction with answers that are not similar to each other. The cosine score for that is 0.3
Redundancy	This is defined when there is 2 very similar or the same question based on the same context with very similar answer. The cosine similarity score for this is 0.85 or higher for the questions and 0.6 or higher for the answers. For the redundant questions, the more relevant pairs are kept, and the rest is discarded.
Relevancy	The first case is if the question or answer score with the context was extremely low (0.2). The consideration here is that the question and answers are compared with the entire context as a whole and not just a specific sentence.

Ambiguity

This can be defined as a question which can have several answers which are not similar, i.e. a question can be answered in multiple ways. It is not unique. Such questions can cause problems when training the extractive question answering model, as it could return a correct answer but it will not register as correct depending on which of the answers the original annotated answer is associated with that specific entry. A simple approach is used where the questions and answers are converted to Term Frequency Inverse Document Frequency(TF-IDF) vectors and the cosine similarity is computed for different combinations. If a high similarity score is obtained between a question and another question, but the answers are not similar at all, this is considered ambiguous, like the example in Figure 6.2.

Context: Certain equipment needs special attention such as planters and combines which are made up of lots of intricate parts. It is also essential to make sure these kinds of machinery are always in top shape as time is crucial when they are operating. Sometimes you may only have a limited time to work and breakdowns will only hold you up and may cost you getting in a crop or reduce your crop quality when harvesting. On these kinds of equipment there are finer parts that need to be checked over such as sieves, augers, blowers, planter plates and vacuum's.

Question 1 : Why is it important to ensure that equipment is always in top shape during harvesting ?
Answer 1 : breakdowns will only hold you up and may cost you getting in a crop or reduce your crop quality when harvesting

Question 2: Why is it important to ensure that certain equipment is always in top shape ?
Answer 2: time is crucial when they are operating

Figure 6.2: A detailed example of a set of question answer pairs which are considered as ambiguous

Redundancy

This can be defined as duplicates, or very similar questions with very similar answers. By removing the redundant entries, this can allow for a more diverse dataset where the entire dataset is made of completely different data points. The same approach is used as for the ambiguity, where the questions and answers are converted to TF-IDF vectors. The different combinations of these vectors are then used to calculate the cosine similarity score. If a high similarity score is obtained for a question with another question, as well as its answer with the same source context, then the question answer pair is considered redundant, as in Figure 6.3.

Relevancy

This can be defined as the semantic similarity between the generated outputs and the source text. We ideally want the generated output to be directly related to the source text, i.e. it should be relevant. This ensures that the generated outputs are contextually appropriate. Three different combinations are tested :

- The semantic similarity between the generated question and the source text confirms that the question is contextually appropriate.
- the semantic similarity between the generated answer and the source text to con-

Context: We have no sooner completed the harvesting of the last seasons crops when we have to start the physical and financial planning for the basket of summer crops to be planted in the 2015 to 2016 summer production season. One of the highest cost centres in crop production gross margin analysis is the budget and financing for fertilisation. I hope as farmers you were fortunate enough to have had a reasonable crop despite the very difficult season just past. Whether you had good or poor yields the question will arise as to the planning of an optimum fertilisation taking into account the physical and financial restraints experienced on your farm.

Question 1 : When did the planning for the basket of summer crops for the 2015 to 2016 production season **begin**?
Answer 1 : we have to start the physical and financial planning for the basket of summer crops to be planted in the 2015 to 2016 summer production season

Question 2: When did the planning for the summer crops in the 2015 to 2016 season **start** ?
Answer 2: We have no sooner completed the harvesting of the last seasons crops when we have to start the physical and financial planning for the basket of summer crops to be planted in the 2015 to 2016 summer production season

Figure 6.3: A detailed example of a set of question answer pairs which are considered redundant

firm that the answer is contextually appropriate , as well as an important aspect of the source text.

- The semantic similarity between the generated questions and the answer is ensured that the answer to some extent addresses the question.

For each of these combinations, a word embedding is generated using the *Sentence Transformer* [3] Library which is based on the BERT model. These embeddings are returned as vectors, and then the cosine similarity is computed to get the final similarity score.

6.4.3 Target Language Answer Annotation

To get equivalent answers in the target language for the generated English question-answer pairs, the same method as 4.3.3 is used with slight adjustments. For each answer in English, the sentence that contains the answer is extracted. The aligned sentence is then extracted in the target language. Since the answers are a phrase in the sentence, the sentence in the target language is then used to generate all the possible phrases at varying length using algorithm 6.2 (see Appendix C for an example of the output). The

[3] https://www.sbert.net/

CoHere multilingual embedding is then used again to align the English answer to the most appropriate phrase in the target language. The aligned phrase is then set to be the answer in the target language.

Initialise all variables
Split the sentence into words w
for i *in the range from* 0 *to the total number of words* l
    for j *in range from* $i + 1$ *to* l
        Create a phrase p by concatenating the words from i to j
return all unique phrases p

Algorithm 6.2: The algorithm that generates phrases based on a given sentence.

Manual evaluation is performed to assess the quality of the aligned phrases. Due to the 1 to many relationship that exists for trying to align the answers according to the specific phrase in a sentence, the answers in the target languages are maintained at sentence level and also at phrase level. At sentence level, the answer in the target language corresponds to the aligned translation of the English sentence which contains the final answer.

6.4.4 Data Consolidation

The context, questions, and answers are brought together to create the final two-way parallel dataset. There are 8727 extractive question answering instances , where each instance contains a context, question, and answer in two languages, one of which is English. Overall there are 5276 unique question answering instances. Table E.1 shows a summary of multi-way instances for the target languages.

Table 6.3: Number of parallel instances between the source language (english) to the target languages

	eng	**afr**	**xho**	**zul**
eng	5276	3279	2711	2737

Full examples of the final data instance for each of the languages are provided in Figure 6.4.

English Context: Datura, commonly known as 'Thorn apple' or 'Olieboom', is a very common weed in most crop producing regions of South Africa. It drives us all mad and they never seem to stop emerging. This weed can be an absolute curse to control and can have severe negative impacts on your crop if left unmanaged.
Translated Context: I-Datura, yaziwa kakhulu nge- 'Thorn apple' noma 'i-Olieboom', wukhula olujwayelekile ezifundeni ezikhiqiza izilimo eNingizimu Afrika. Yingakho sibona lolu khula lumila ngamandla ikakhulukazi emihlabeni ejwayele ukuthilwa. Lelisu kufanele liqaliswe ukusetshenziswa kusukela kutshalwa futhi kuqhutshekwe nalo esigabeni sonke sokukhula kwesitshalo sommbila.
Question : What is the common name for Datura ?
English Answer : Datura, commonly known as 'Thorn apple' or 'Olieboom',
Translated Answer : I-Datura, yaziwa kakhulu nge- 'Thorn apple' noma 'i-Olieboom',

English Context: Farming, especially crop farming relies on mechanical equipment and all mechanical equipment has moving parts. All moving parts wear down after a period of time therefore they need continuous maintenance and attention to make sure that they are in good working order.
Translated Context: Umsebenzi wokufama, ngakumbi owokulima izityalo uxhomekeke kwizixhobo ezingoomatshini kanti zonke izixhobo ezingoomatshini zineendawo ezinentshukumo. Zonke iindawo ezinentshukumo ziyadleka ethubeni ngoko ke zidinga ukusoloko zilungiswa kwaye ziqwalaselwa ukuze zibe kwimeko elungileyo yokusebenza.
Question : What is needed to ensure that mechanical equipment in farming is in good working order ?
English Answer : continuous maintenance and attention
Translated Answer : ukusoloko zilungiswa kwaye ziqwalaselwa

English Context: Farming, especially crop farming relies on mechanical equipment and all mechanical equipment has moving parts. All moving parts wear down after a period of time therefore they need continuous maintenance and attention to make sure that they are in good working order.
Translated Context: Boerdery, veral saaiboerdery, maak staat op meganiese toerusting en alle meganiese toerusting het bewegende dele. Alle bewegende dele slyt na 'n tyd, daarom het hulle deurlopende instandhouding en aandag nodig om seker te maak dat hulle in 'n goeie werkende toestand is.
Question : What type of farming relies on mechanical equipment ?
English Answer : Farming, especially crop farming relies on mechanical equipment
Translated Answer : Boerdery, veral saaiboerdery, maak staat op meganiese toerusting

Figure 6.4: An example of how the final data is structured. The text outlined in blue is an example from Xhosa, the red is an example from Zulu and green is an example from Afrikaans

The final dataset is then split into a training, development, and testing dataset for the final experimentation. This split is a 40%, 40%, 20% split for each of the languages.

The statistics of the full dataset can be viewed in Appendix E.

6.5 Quality Control and Management

Despite using an automated approach for the annotation of the data, some manual review of this data needs to be performed. Traditionally, for the data annotation process, the annotations are assessed according to the annotation guide that is provided, and only the entries that meet the criteria in the annotation guide are kept. This is a time-consuming process which is done simultaneously during the annotation process.

To assess the quality of the data that is generated, an annotation guide is created and used to assess a random sample of the annotated data. This review aims to pick up any minor inconsistencies that would not be picked up with the automated methods and to get any insights of some limitations that could have been encountered with this method of annotation.

The evaluation guidelines which were included in the annotation guide can be found in the appendix D. To evaluate the quality of the question-answer pairs, a set of questions need to be answered with respect to both the question and the answer. Once a rating is assigned to the question and the answer, an overall rating can be assigned to the question-answer pairs. From this annotation guide, the evaluation flowcharts in Appendices D.1 and D.2 are used for each of the data instances of the random sample and given a final rating of:

- Poor: This is when at least one of the ratings is bad, i.e. the question or answer rated as bad. This means that the output did not meet expectations.
- Reasonable: This is when at least one of the ratings is good, i.e., the question or answer rated as good. This means that the output is not an exact match of the desired output, but is still useful.
- Excellent: This is when both the question and answer rating is excellent. This means that the output did meet the expectations.

Using this evaluation guide, the aim was to get a comprehensive overview of the logical and grammatical quality of the data, i.e., to make sense and is proper grammar used. The final distribution of the manual assessment is provided in Figure 6.5. A random subset of 120 data points was assessed.

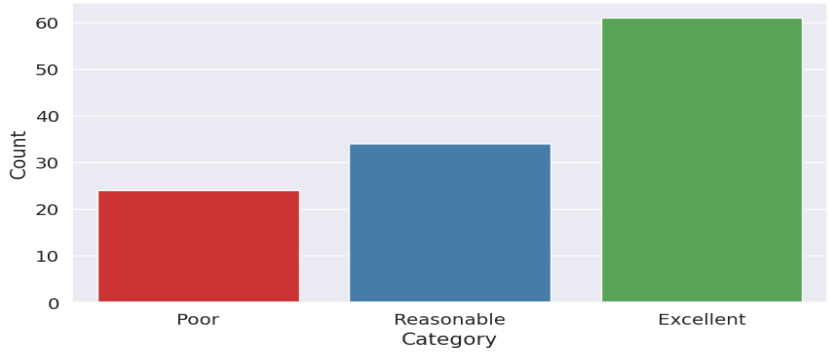

Figure 6.5: The final distribution of the rating of a random subset of question and answer generation outputs

Overall, from the evaluation, the dataset performed relatively well, where the majority of the data were categorised as reasonable and excellent. From the examples that are rated as excellent, most of them were paraphrased versions of each other, i.e. the question was a paraphrased version of the answer. This meant that there is a strong logical relationship between the questions and the answers. Another major type of question which were rated as excellent was questions which dealt with quantitative answers. The reasonable questions were still logical, but mostly the answers contained extra information which made the answers less straightforward, which is a key quality for extractive question answering. For most of the questions that were rated poor, the questions showed more complexity and the respective answers could not sufficiently answer the question.

6.6 Summary

The purpose of this chapter was to provide a detailed account of the data annotation pipeline that is used. Additionally, it provided a discussion on the quality of the data through manual review. From this manual review, it was concluded that the data an-

notation pipeline was able to create a dataset that met the desired characteristics. By following the steps outlined in this chapter, the research objective of developing a cross-lingual agro-information dataset was successfully achieved.

Chapter 7

Analysis of the Annotated Data

From the resultant data generated through the annotation pipeline in Chapter 6, this chapter aims to further analyse the data. Through this data analysis, we hope to derive the attributes of the final data and assess the robustness of the automated annotation method used. To fully understand the properties of the annotated dataset, we analyse the questions and answers that were generated. We specifically analyse three aspects of the question-answer pairs :

- Section 7.1 gives the descriptive statistical qualities of the question and answer pairs that were obtained.
- Section 7.2 provides the topics that are addressed in the question and answer pair.
- Section 7.3 gives an overview of the diversity of the questions and answers of the generated data.
- Section 7.4 provides a summary of the main observations that are made during the analysis.

7.1 Descriptive Statistics

Understanding the general structural properties of the annotated data can provide insight into the filtered data and the results of the annotation. The way in which the

annotated data is structured can provide insights about the quality of question answer pairs, as well as the difficulty of the question answer pairs.

Table 7.1: Summary of the unique values of the annotated source language data

	Total Number
Instances	5276
Articles	85
Context	707
Unique question-answer pairs	5248
Unique questions	5151
Unique answers	3863

Table 7.1 shows the overall distribution of all articles and contexts in the source language. Overall there are 5276 data instances, but not all these instances have unique question and answers. This indicated that some of the articles tackled very similar topics as other articles. This also means that there is limited diversity present in the initial articles that were used for the annotation process.

The next analysis aims to understand how the generated data is distributed. This was done by analysing the lengths of the context, question, and answers individually. To represent the length, the word count for each tokenised text was used. The distribution of the length for the context, answer and questions can be seen in Figure 7.1.

Figure 7.1 a shows that the context length ranges from 18 to 405. Generally in pre-trained models, the maximum sequence length is 512. Since all contexts fall below 512, truncation of the context is not necessary for the experiment that is performed. In figure 7.1b, the length of the question ranges from 3 to 33. Since all the questions fall under 128 tokens, truncation is not necessary during training. An example of a question of minimum length is *"What is AgriCloud ?"* and *"What is NAMPO?"* and at maximum length is *"What are some minor nutrients that are usually added to the*

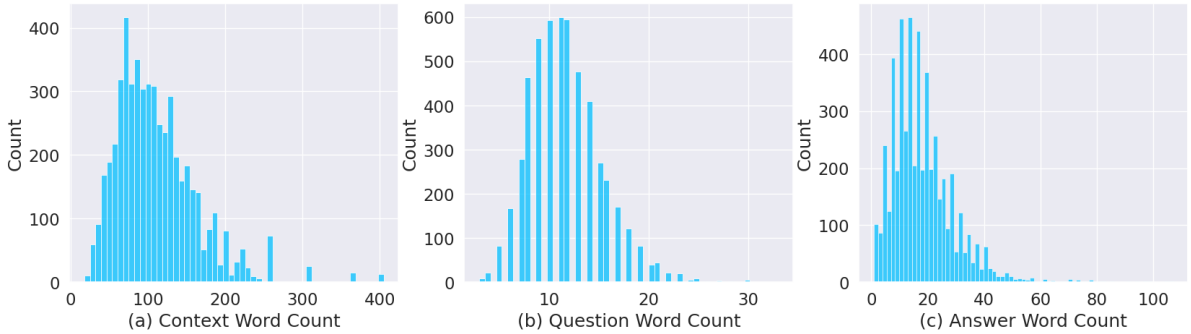

Figure 7.1: Distribution of the context, answer and question length for the English annotated data

fertiliser mix or seed dressing in South Africa as a precaution against any deficiency in the soil leading to poor plant growth? ". From this example, at both the minimum and the maximum lengths, the questions make logical sense and show that the length of the question could be correlated with the complexity. The longer question is more detailed and specific, which would require a bit more understanding to derive the correct question.

From the last figure, Figure 7.1c, the answer length ranges from 1 to 107. Most of the answers are of length 1 to 40, meaning that most of the answers are short form answers. At length 1, an example of a question answer pair is *"What crop is recommended to be planted after beans in a crop rotation plan?"* ***maize*** . From this example alone, it can be seen that even though the answer is 1 word, the question is still adequately answered. At the maximum length of the answers, an example is *"Why is it important for profits to be made?"* ***Should no profits be made: You will for instance not be able to pay yourself a salary - you work for nothing; Should you wish to expand, or improve your farming business you will have no funds available to do it; Should you wish to replace movable assets such as a tractor or tools, which becomes necessary with time, you will not have funds available to do it; and Should you then wish to do the above-mentioned and you need to obtain a loan it will not be possible, because how will you be able to repay the loan*** [1]. From this it can be seen that the answers at the maximum length have a

[1]Directly from the articles from Pula Imvula

lot more detail but still adequately and fully answer the question.

Another type of analysis that was performed is the correlation between the length of the questions and the answers. The main question here is to see if there is any correlation between the length of the question and the answers. One way to obtain the correlation is by calculating the Pearson coefficient, which measures the relationship between two variables in terms of strength and direction. The coefficient of length between questions and answers is 0.077267. This number can be rounded to 0, which means that there is no correlation between the length of the questions and the length of the answers.

7.2 Content Analysis

Since the data annotation did not specify topics to be addressed when creating the question answer pairs, the topics that are being addressed in these pairs need to be uncovered. One way to gauge the recurring topics is to look at the word frequency. Instead of looking at the top individual words that occur, the bigrams of the text is used. The bigrams offered more comprehensive insight into what is being addressed in the questions and answers, compared to individual words.

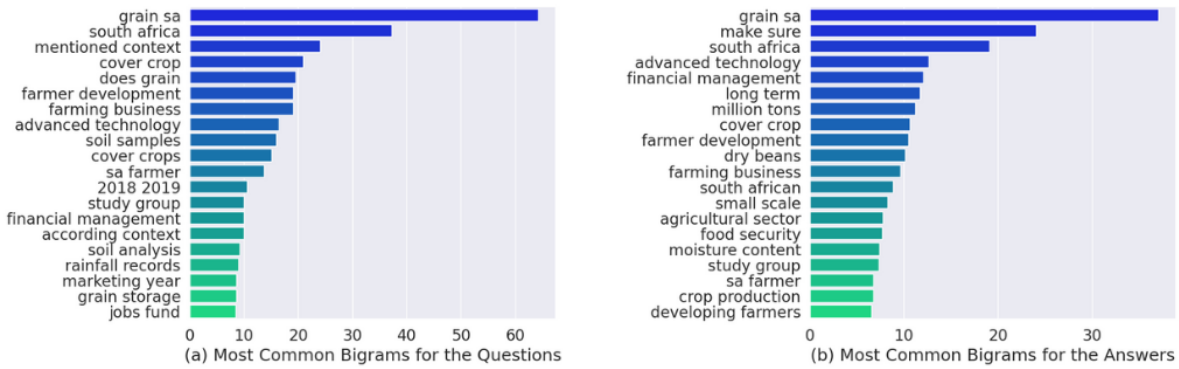

Figure 7.2: Distribution of the most common bigrams that occur in the generated questions and answers

In Figure 7.2, there is some overlap between the question content and the answer content. From these overlaps, the main topics that are addressed in the question answer pairs can be derived. Some of these topics are as follows:

- Financial Management
- Farmer Development
- Marketing
- Technology
- agricultural produce such as cover crops
- environment related such as soil and rainfall

Referring back to one of the objectives of this thesis, the common bigrams indicate that the generated question and answer pairs address the property of the dataset being domain specific (Agriculture) and region specific(South Africa). On top of this, it can also be derived that the question and answer pairs address important issues that would be relevant and useful to developing farmers.

The bigrams can also be compared with the word cloud generated in Section 5.2.1 from the original articles, to evaluate the effectiveness of the data annotation pipeline. The main aim of the data annotation was to capture the main topics that are addressed in the original articles used to create the contexts. Figure 7.2 shows the words that occur most frequently in the questions and the answers. Through comparison of Figure 5.2, it can be seen that the annotations made for the data address and capture the main topics addressed in the articles.

7.3 Diversity

To understand the properties of the generated answers, we specifically explore the types of answers defined in the initial analysis performed in SQuAD [41]. Table 7.2 shows all

the types of answers in the dataset. To start off the categorisation of the answers, the answers can be split into either numerical or non-numerical answers. Numerical refers to any answer that contains a number. The general numerical answers make up 7.45% . The numerical answers include dates, a variety of measures such as weight and distance, and money.

The non-numerical answers can be further categorised using Named Entity Recognition (NER) and Part of Speech (POS) tags. The *Stanford CoreNLP package*[2] is used for named entity recognition and constituency parsing. Through part of speech tagging, answers that contain proper nouns can be determined. In general, proper nouns make up 15. 95% of the data and can be further broken down into Person, Location, Organisation, and other entities. For all the phrases that do not contain proper nouns, they fall into one of the 3 types of phrases: verb phrase, common noun phrase and adjective phrase. From these 3 types of phrases, the majority of these phrases are common noun phrases which make 76.29% of the dataset.

Another way to analyse the diversity that exists in the dataset is to analyse the interrogative words present in the question. The interrogative words provide some insight into what is being addressed in the question. From Figure 7.3, most of the data is comprised of ”what” type of questions and least type of question is ”whom”.

[2] https://pypi.org/project/stanford-corenlp/

Table 7.2: Summary of the different types of answers in the annotated data

Category	Percentage (%)	Example
Numerical	7.45	contractor charged farmers R3 500
Person	1.61	Wilbur Wright, inventor and builder of the world's [...]
Location	0.95	[...] to be the best small scale grain producer in Taung.
Organisation	4.51	[...]the agricultural futures exchange on the Johannesburg Stock Exchange (JSE)[...]
Other Entities	8.89	'Thorn apple' or 'Olieboom'
Verb Phrases	0.30	germinate simultaneously
Common Noun Phrases	76.29	[...] person will then be known in common language as your bookkeeper

7.4 Summary

Through the analysis performed in this chapter, the attributes of the final QA dataset is provided. From the descriptive statistical analysis, it was seen that there is a wide variety in the structure of the dataset where short-form answers and long-form answers are present. To answer part of the research sub-question 1 to create a QA dataset for agro-information, the contextual characteristic was that the data was analysed. Through the analysis that was completed, it was confirmed that the topics that were addressed in the question and answer pairs are aligned with the ideal data properties that were defined. Finally, the diversity that exists in the data set is analysed through the types of answers and the questions that are presented. The analysis confirmed that there is a degree of diversity between these two attributes.

Table 7.3: Summary of the different types of question in the annotated data

Category	**Percentage (%)**	**Example**
Who	4.62	Who has been outspoken about the use of agricultural contractors in the developing sector?
What	72.54	What type of farming relies on mechanical equipment?
Where	1.95	Where do the grants come from?
When	2.65	When should soil samples be taken for a new farm or land ?
Why	3.53	Why are the people against the use of contractors?
Which	1.55	Which direction should the air filters be blown out daily with a compressor?
How	11.66	How should the financial statements be compiled for management purposes for a farming business?
Whose	0.13	Whose responsibility is it to ensure that all employees are aware of their responsibilities
Whom	0.04	On behalf of whom does Grain SA negotiate when a serious problem is identified ?
Other	1.33	Can financial record keeping be done manually for small businesses?

Chapter 8

Experimental Setup

The previous chapters 4 to 7 discussed the collection, analysis, and annotation of the final Pula Imvula Question Answering dataset. This study aims to use this dataset to explore the effectiveness of different model fine-tuning techniques: few-shot learning through prompt-based fine-tuning for the specific data properties discussed in Section 4.1.1. In this chapter, we provide a detailed discussion of the experimentation that is performed.

This chapter investigates the 2^{nd} & 3^{rd} research sub-questions and their respective objectives. Different aspects to answer these 2 sub-questions motivate the way the main experiment is divided and investigated to gauge the overall effectiveness of the fine-tuning methods used. A more comprehensive technical background for this experiment is provided in Chapter 3, where the different fine-tuning methods and models investigated are discussed. To prevent redundancy as there is an overlap between the different parts of the experimentation, the common aspects of the experimentation is grouped together. This chapter is organised as follows:

- Section 8.1 provides the details of the investigation scenarios of the experimentation based on the 2 research sub-questions.
- Section 8.2 provides the details of the standard fine-tuning that is performed.
- Section 8.3 discusses the details of prompt-based fine-tuning. This includes all the

design decisions made for the final prompting.

- Section 8.4 provides the technical details of the general few-shot setup used in this experimental setup for both the standard fine-tuning and the fixed-prompt language model fine-tuning.
- Section 8.5 gives the details of the evaluation metrics used to access the performance of the models.
- Section 8.6 summarises the general details of the experimental setup.

8.1 Investigation Scenarios

The experimentation is divided into two parts. These two parts are as follows:

1. Investigate the effectiveness of the different prompt-based fine-tuning techniques in a few-shot setting for a domain-specific question answering dataset.
2. Investigating the effectiveness of the different fine-tuning techniques in a few-shot setting for a cross-lingual domain-specific question answering dataset.

Since there are two different objectives of the experimentation, there are different aspects of the experimentation that will be analysed. However, the common theme that will be maintained is to study the effectiveness of prompt-based fine-tuning for this experiment. For both investigations, the same few-shot settings are used, and these are discussed in detail in Section 8.4.

8.1.1 Fine-Tuning for Domain-Specific Text

This first experiment looks at the different aspects of the fine-tuning techniques that can affect the general performance for the prompt-based fine-tuning. The main aspects which are investigated are:

- The performance of the tuned models is based on already established prompt-based fine-tuning methodologies.

- The effect of different prompt templates on the performance of the model in a few-shot setting.
- The difference in performance of prompt-based fine-tuning methods compared to the standard fine-tuning method.

8.1.2 Prompt-based Fine-Tuning for Cross-lingual data

This part of the experimentation is an extension of the experiment done in Section 8.1.1. Here, the main objectives is to investigate :

- Training the models on the cross-lingual data, where the questions are asked in English, and the answer and context is based on the context answer.
- Adaptation of prompts to suit the languages (translation) and looking into language-agnostic prompts such as null prompting.

From the results obtained in the first part of the experiment, we used these results as the basis for this experiment. We extend the original fine-tuning framework to work in a multilingual setting by using the multilingual model equivalent mentioned in 8.4.4.

8.2 Standard Fine-tuning

For promptless fine-tuning/standard fine-tuning, the original objective of the QA task is obtained from the original fine-tuning done on the pre-trained model, as discussed in Section 3.4.1. To re-emphasise the points discussed in this section, a simplified overview of the training objective is provided in Figure 8.1.

8.2.1 Input-Output Design

Maintaining the main objectives of the standard training for QA, we define the input and output to main this. We can define the input to the model as x_{input}, which is made from the two texts that make up the Extractive QA, which is the context and the question. These texts can be defined as :

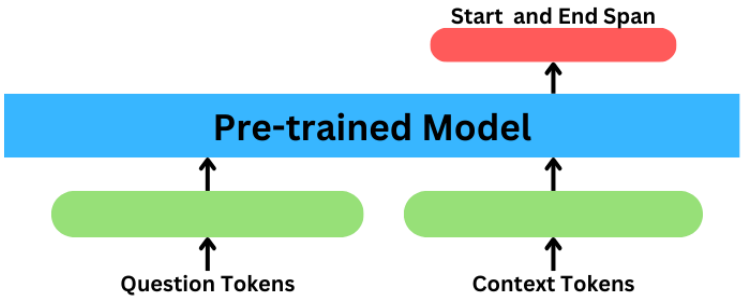

Figure 8.1: A simplified overview of the main objective of standard question answering

- x_q which is the tokens that make up the question
- x_c which is the context sequence

For the state-of-the-art models, the question and context is provided to the model as separate texts, thus the input is :

$$x_{input} = [x_q], [x_c]$$

The answer that is predicted is the span of the text that makes up the answer from the context. This span is returned as two numbers, which is the start index of where the answer starts and the end index of where the answer ends, therefore the output can be defined as:

$$y_{output} = [y_{start}, y_{end}, y_a]$$

An example of the input-output design of this fine-tuning method is provided in Figure 8.2.

8.3 Prompt-based Fine-tuning

This fine-tuning is discussed in detail in Section 3.4.2. To re-emphasise the points discussed in this section, a simplified overview of the training objective is provided in Figure 8.3

Input	Output
Question : Why are dry beans named as such ? **Context** :Dry beans are so named because they are normally left on the plant until the pods have dried. The entire plant is then pulled up, placed in the shade where possible and allowed to dry for an additional one to two weeks. The dried pods are then split up and the beans removed	**Start :** 30 **End :** 80 **Answer :** they are normally left on the plant until the pods have dried

Figure 8.2: A detailed example of how the input and output looks like for standard fine tuning

8.3.1 Input-Output Design

For all the prompting that is done during the experimentation, a standardised input-output design is used. This design is chosen and is based on the results obtained in the research for FewshotQA [7]. Since the aim of this study is not to investigate or create a new prompt-based tuning method but rather to investigate the effectiveness of the different prompting strategies for the dataset, the design decisions are made based on the results obtained from previous studies. In the research done in [7], extensive research on the input-output design was completed and we selected the design based on the performances of these different designs.

We can define the input to the model as x_{input}, which is a concatenation of the three texts that make up the extractive question answering data instance, which is the context, the question, and the answer. These three texts can be represented and defined as follows:

- x_a which is the tokens which make up the answer. In this fine-tuning, this will be the *masked* portion of the input sequence which needs to be predicted.
- x_q which is the tokens that make up the question
- x_c which is the context sequence.

From the experimentation done in [7], the order in which the sequences are designed affect the overall performance of the prompts, thus the standard order which is selected is:

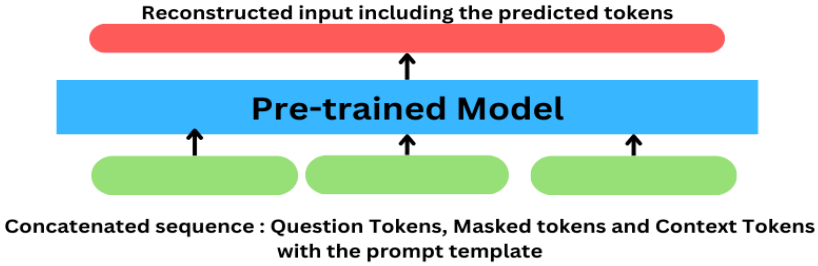

Figure 8.3: A simplified overview of the main objective of prompt-based question answering

$$x_{input} = [x_q, x_a, x_c]$$

Another aspect of the prompting which affects the performance is what the model is asked to generate as an answer. From the input sequences, the model can be tasked with predicting only one, a combination of two or to generate all three input sequences. The most effective generation setting is to only generate the question and predicted answer, therefore, the output can be defined as:

$$y_{output} = [y_q, y_a]$$

An example of the input-output design of this fine-tuning method is provided in Figure 8.4.

Input	Output
Question : Why are dry beans named as such ? **Answer** : ***[Masked Tokens]*** **Context** : Dry beans are so named because they are normally left on the plant until the pods have dried. The entire plant is then pulled up, placed in the shade where possible and allowed to dry for an additional one to two weeks. The dried pods are then split up and the beans removed	**Question :** Why are dry beans named as such ? **Answer** : they are normally left on the plant until the pods have dried

Figure 8.4: A detailed example of how the input and output looks like for the Prompt-based fine tuning

8.3.2 Prompt Strategy Design

In Section 3.4.3, an in-depth discussion is provided on the different recent prompting techniques that have shown promising results in the field. These prompting strategies can be divided according to how the prompt is designed, which is single prompts and multi-prompt. One thing that has been highlighted across all the techniques is how crucial the prompt template is. These prompt templates can be designed either manually or automatically generated through different techniques. From each of the prompting strategies considered in this study, the prompts are defined based on the prompting style defined in the different strategies.

For single prompt templates, the designed templates are based on the research done in [7, 28]. Most of the research that has been done using some of the prompting methods for domain-specific data has been extensively for tasks such as text classification, where the prompt templates can be easily adapted to suit the type of text. However, for a task such as question answering, the templates are based on the actual task and not the contents of the tasks. The final templates that are used are summarised in table 8.1.

Table 8.1: Summary of the different prompt templates that are used in the experiment based on the different prompting strategy

Template	Prompt structure
1	"{x_q} {mask} {x_c}"
2	"***Question:*** {x_q} ***Answer:*** {mask} ***Context:*** {x_q}"
3	"***Question Answering:*** {x_q} {mask} {x_q}"
4	"{x_q} ***The answer is:*** {mask} ***which is based on the following context:*** {x_q}"
5	"***Answer the following Question:*** {x_q} ***Answer:*** {mask} ***Context:*** {x_q}"
6	"{x_q} ***The answer is:*** {mask} {x_q}"

The first template is based on null prompting, which states that prompt-based fine-

tuning is still effective without the need for discrete prompts. The second template is based on FewShotQA where the prompt is based on adding the discrete prompts "Question, Answer and Context", which are added in front of the different sequences of the inputs. The third template is based on the idea of a prefix prompt where the intended task is included as the prefix. Template 4 is based on mimicking the cloze-style task where the model fills in the blank. The last two templates are designed to mimic more of an instructional prompt with direct instructions.

8.4 Few-shot Setting

The main setting of this study is to cater for domain-specific data and multilingual data in low-resource languages. This means that the data is limited, thus a few shot setting makes the most sense and caters for the objective of this study. In any few-shot study, there are several considerations that need to be noted as these affect the performance of the model in a few-shot setting. In this experiment, the different aspects of this are taken into account and lead to the different decisions that are made for the final design of the experiment.

8.4.1 Data Sampling Strategy

In a few-shot setting, the representation of the data instances is very important. Since only a limited amount of data is used to tune the model, the data included can affect the overall performance of the tuning strategy. However, since the data is domain-specific and closely related, the sampling strategy might not be effective. Here, we use random sampling.

8.4.2 Dataset Split

In a practical sense, in a few-shot setting, there is no access to the full dataset that is used for the training of the model. Generally, a dataset is split into the development set, which is used during the training of the model, and the test set, to get the performance

of the train model. Usually for this, the developmental set is significantly larger than the test set. However, for this few-shot setting, we make the developmental set and test set equal, as we are working under the presumption that there is only limited data available.

8.4.3 Hyper-parameter Setting

In traditional fine-tuning, a large dataset is required to achieve high performance. In such a setting, optimisation of the hyper parameters can be performed automatically where different parameters are changed and the most optimal combination is used. Realistically for a few-shot setting, there isn't enough data to investigate and obtain the most optimal hyper parameters for any task. Based on this, we select and use the hyper parameters used in FewShotQA [7] that are provided in Table 8.2.

Table 8.2: Summary of the different hyper parameter setting used for the experimental

Parameter	Value
Training epochs	20
Learning rate	2e-5
Training batch size	4
Optimizer	AdamW
Validation epochs	20

For each aspect of the experimentation, the model is fine-tuned on 5 different samples of the dataset. The final results collected from each of the runs are based on the highest performing model based on the test set.

8.4.4 Model

The aim of prompt-based fine-tuning is to adapt the objective of the task that is being performed to match that of the initial objective of the pre-training setup. A variety of different pre-trained language models exist, and their pre-training objectives differ. The main objective of the prompting investigated in this study is to change the objective QA to a text-to-text framework where the model is tasked with predicting the mask that

makes up the answer, rather than the length of the answer. The final model which is used in the BART model and its multilingual equivalent as these models are pre-trained to predict multi-mask outputs rather than single-mask outputs such as BERT. The details of the models are provided in Table 8.3.

Table 8.3: Summary of the details of the different models used in the experimentation

Model	Architecture	Size
Bart-large [1]	Sequence2sequence	406M parameters
mBART [2]	Sequence2sequence	611M parameters

For the standard fine-tuning, the model that is used is valhalla/bart-large-finetuned-squadv1, which is the Bart-large model that is trained on the SQuAD v1.1 dataset [41].

8.5 Evaluation

For the evaluation of the performance of the different fine-tuned models, we only consider the final text answer. To obtain the final answer for the standard fine-tuning, the predicted span indices are used to extract the text from the context according to these indices. For prompt-based fine-tuning, the generated output is a combination of the question and the predicted answer. Simple heuristics are used to separate these two strings, and then the final answer only contains the predicted answer.

8.5.1 Metrics

There are two main metrics that can be used to empirically measure the performance of question answering models, the exact match (EM) and mean f1 score. EM takes into account the question-answering pairs and exactly matches the prediction of the characters of the models versus the true answer. This is, however, a very strict metric to measure the performance of the models [13].

F1 is the harmonic mean between the recall metric and the precision metric. This metric is computed for the individual words, the predicted word against the true answer.

Let the number of words shared between the predicted and true answer be known as n_{shared}. The recall metric is calculated as the ratio between n_{shared} over the total number of words in the true answer. The precision metric is n_{shared} over the total number of words in the predicted answer [13].

8.6 Summary

The different investigations that are performed for the experimentation are described in this chapter. With each of these strategies, the aspects of the investigations are provided in detail. From these investigations, the general details followed for the different fine-tuning techniques that are done are provided. Overall, the experimental setup of this study is discussed in this chapter includes the final few-shot setting and evaluation metrics that used to assess the performance of the models.

Chapter 9

Fine-tuning for Domain Specific Text

One of the research sub-questions we asked is: *"What is the feasibility of applying the model fine-tuning techniques for monolingual(English) domain specific question answering data?"*. To answer this question, several experiments were completed to determine the feasibility of the fine-tuning methods that were investigated. This chapter discusses the results obtained from the overall experimentation done to answer this research question.

9.1 Experimental Objectives

As mentioned previously, this experimentation aims to answer the 2^{nd} sub-question. In order to adequately answer this sub-question, the following questions are posed :

- For prompt-based fine-tuning, which established methodology works better for the data, i.e. task agnostic versus task specific?
- Based on the established prompt-based methods, can any improvements be made to the prompts?
- Compared to standard fine-tuning in a few-shot setting, does prompt-based fine-tuning perform better?

- Do the results obtained from the prompt-based fine-tuning reflect the adequacy of the methods applied?

9.2 General Comparison between Prompts

Prior to performing any adaptations and changes to well-known prompt-based fine-tuning, we tested which of the 2 - null prompt versus FewshotQA works best for the domain-specific data. The prompt templates used to test these approaches are represented as template 1 and template 2 respectively in Table 8.1. For each of the selected number of example investigated for the few-shot fine-tuning, 5 different subsets of the dataset was sampled and used to fine-tune the model. The resulting F1 score was based on the average obtained in the 5 runs. The full results can be seen in Appendix F. For each run, the best model was saved and this model was used to obtain the final F1 score for the test data. The final results are presented in Figure 9.1.

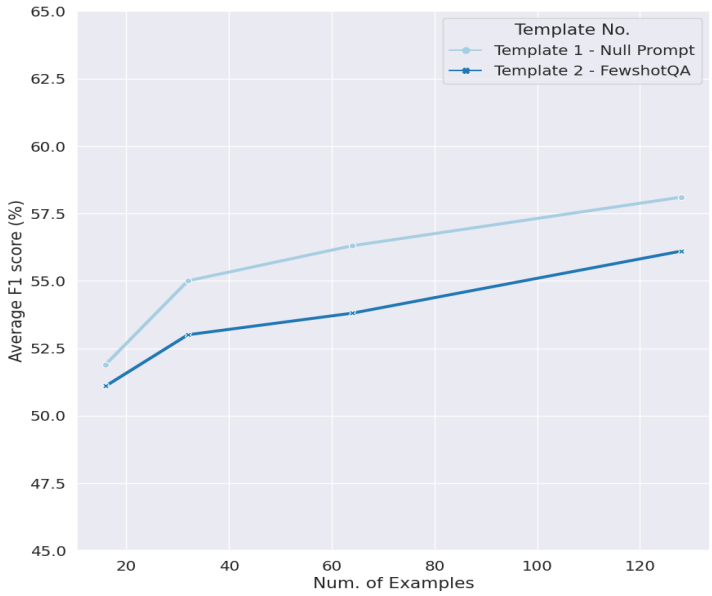

Figure 9.1: The F1 score obtained for Null and FewshotQA prompting

From the result obtained, it can be seen that there is a minor difference between the results obtained for the null prompting and FewshotQA. In general, null prompting

(**58.1%**) performs slightly better than FewshotQA (**56.1%**). Due to the closeness of the average results, the 2 templates have proven to both be applicable to the domain-specific data.

Another aspect which needs to be discussed is the general trend in the results for both these templates. As the number of examples increases, there are slight increases in the F1 score, where the highest degree of improvement is **4.8%** for the null prompting and **2.3%**. From these numbers, the Null prompting performance improves more rapidly as the number of examples increases, compared to the FewshotQA. This suggests that the null prompting method is more responsive to the quantity of examples utilized for model training.

Based on previous research, it is anticipated that there will be a significant increase in the results as the number of examples used for fine-tuning a model increases. However, in this case, the degree of improvement is minimal despite the increase in the number of examples. This can be indicative of the quality of the data used. In a case where open domain data is used, the expectation is that as the number of examples increases, the performance will increase as more information is provided for the different domains included in the dataset.

Since we are using a domain-specific dataset, there is limited improvement that is made to the results obtained as the number of examples increases. These results could indicate that the limited number of examples was an adequate representation of the overall dataset and the topics that the dataset addresses. Based on the initial samples of 16 data points, it appears that the model has captured the limited variability found in domain-specific data. This suggests that the performance of the results may be constrained by the characteristics of the actual data.

These results are also further emphasised in the topic modelling analysis that was performed in Sections 5.2.1 and 7.2, where there were a limited number of topics present in the dataset.

9.3 Template Adaptation

Prompt designing has been highlighted to have a huge effect on the performance of a model which is fine-tuned through prompt-based methods. Simple adaptations to the FewshotQA prompting method were made by making slight changes to how the model is instructed. The prompt adaptations were based on trying to mimic natural language instructions that can be used in standard reading comprehension tasks. The adaptations are presented in Table 8.1 as templates 2 to 6. The overall results for each of the prompts were obtained similar to what was done in Section 9.2, and are presented in Table 9.1. The full detailed results are provided in Appendix F, Table F.1.

Table 9.1: The average F1 scores obtained from the few-shot setting of 128 examples for the different prompt templates, as a percentage. The highest obtained score is highlighted in bold with an asterisk.

	Prompt 1	**Prompt 2**	**Prompt 3**	**Prompt 4**	**Prompt 5**	**Prompt 6**
F1 (%)	58.1	56.1	55.1	**57.3***	55.0	55.0

The results presented in Table 9.1 demonstrate that minor modifications to the prompt templates have an impact on the final results. This emphasises the significance of prompt design in fine-tuning prompt-based QA, even when the prompts themselves are not specific to a particular domain. The differences in the templates affected the performance of the prompt-based fine-tuning. From Figure 9.2, it can be seen that the performance improvements are more unstable, compared to the original prompting in Figure 9.1. This indicates that there is a level of lack of robustness with this method, i.e. the performance of the model is dependant on the prompt design. Therefore, it is necessary to carefully craft the templates and extensively test them to make them more suitable for the task.

From Table 9.1, the final result of prompt 4 performs better than the original FewshotQA prompt (prompt 2). To further analyse this result, Figure 9.3 shows the results obtained in different numbers of examples. From this figure, it shows that template 4

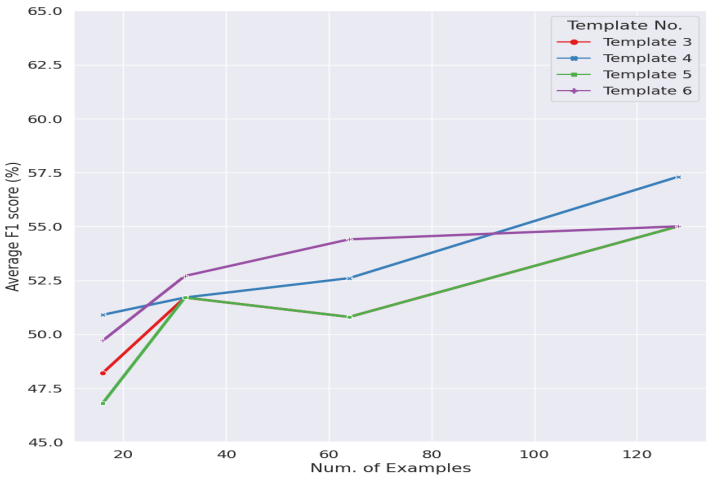

Figure 9.2: The F1 score obtained for the different designed prompts

only bypasses FewshotQA at 128 examples and performs slightly worse for 32 and 64 examples, but starts from a similar F1 score at 16 examples. This highlights the weakness of this approach, that is, the lack of robustness, as the varying degrees of improvement in Figures 9.2 and 9.3 indicate that there is no consistent pattern of improvement.

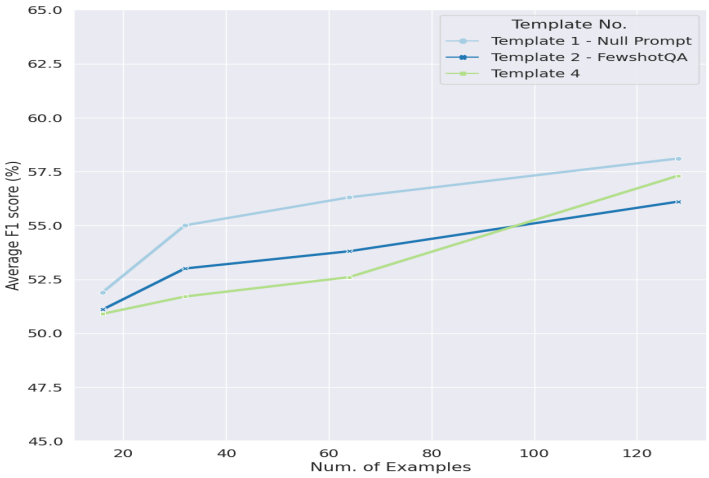

Figure 9.3: The F1 score for Null, FewshotQA and Template 4 prompting at different numbers of examples

Overall, the general results of all the templates show that as the number of examples

increases, there are only slight gains made of the average F1 score. This observation suggests that prompt-based design may be effective in capturing the limited diversity of domain-specific data, as evident from the initial samples. To build on this point, we compared prompt-based fine-tuning to standard fine-tuning.

9.4 Comparison of Fine-tuning methods

In previous research, few-shot prompt-based methods have proven to be better than standard fine-tuning for several NLU tasks. Since this is a variate application of prompt-based fine-tuning, it is important to ask the following question:

> *"How does the prompt-based method compare to the standard fine-tuning of the data?"*

By answering this question, the findings of the prompt-based method can be confirmed as either advantageous or not for the domain-specific dataset. We sampled five different subsets of the dataset for different percentages of the training data and trained the models using standard fine-tuning, null prompting, and FewshotQA. The final results are presented in Table 9.2. Full details results are available in Appendix F, Table F.2.

From the results in Table 9.2, we can see that, overall, the standard fine-tuning performs better than the prompt-based methods. We can also see that at different sizes of the dataset, different fine-tuning methods perform better than the others. To better visualise this, we calculate the difference between the F1 score of each of the prompting methods with the standard fine-tuning methods. The results of this calculation are shown in Figure 9.4.

Overall, from Figure 9.4, prompt-based methods perform significantly better (where the biggest difference is **16.5%**) at small data sizes compared to larger data sizes. This means that as the size of dataset increases, the standard fine-tuning method is more advantageous, whereas the prompt-based methods are advantageous at smaller dataset sizes, i.e. in a few-shot setting. This implies that the prompt-based method is more

Table 9.2: The average F1 scores obtained from the different fine-tuning methods using standard fine-tuning, Null and FewshotQA prompts, as a percentage. The highest score obtained at each sample size is highlighted in bold with an asterisk.

Fraction of Data Sampled	Fine-tuning method	F1(%)
0.5%	Standard	30.4
	Null Prompt	39.9
	FewshotQA	**46.4***
1%	Standard	41.6
	Null Prompt	**58.1***
	FewshotQA	51.3
2.5%	Standard	53.8
	Null Prompt	**56.9***
	FewshotQA	55.3
5%	Standard	**62.6***
	Null Prompt	57.3
	FewshotQA	55.4
10%	Standard	**66.9***
	Null Prompt	60.1
	FewshotQA	59.1

useful for limited data. This highlights the effectiveness of using prompt-based design on the domain-specific data where the data is usually limited.

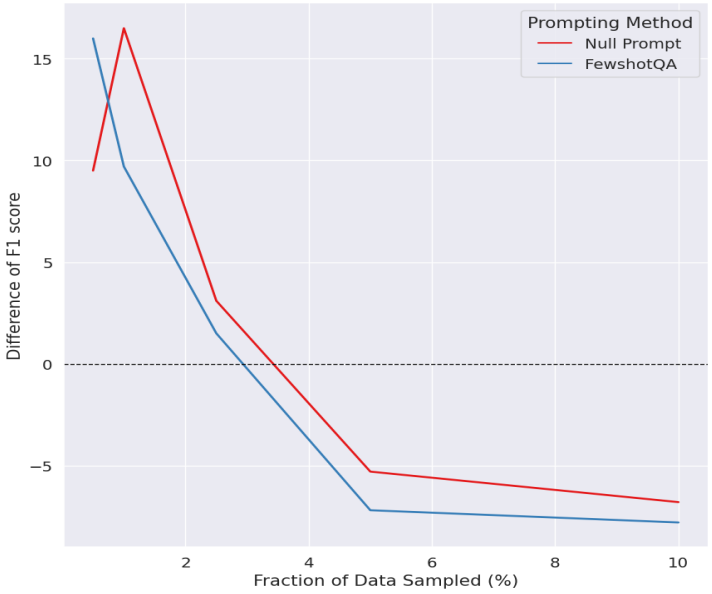

Figure 9.4: The difference in performance of the prompt-based methods to the Standard fine-tuning at the different sample sizes.

9.5 Effectiveness of the Prompt-based methods

In the original FewshotQA research [7], the SQuAD [41] dataset was used to assess the prompt-based fine-tuning method for question answering. This is an open-domain benchmark dataset that is widely used to train PLMs for question answering. In this study, we were required to create a domain-specific dataset from scratch. The automated approach we used to annotate the data could pose as a limitation of the results obtained, despite the fine-tuning method used. To establish whether or not the results obtained for the prompt-based methods reflect the fine-tuning approach, we compared the performance of the fine-tuning methods on the SQuAD dataset with our dataset. By comparing the results, we try to answer the following question :

> *"Do the results obtained from the prompt-based fine-tuning reflect the adequacy of the methods applied? "*

The same experimental setup was used for the 2 prompt templates : Null prompt(template 1) and FewshotQA(template 2). The results are provided in Table, and the detailed

results are in Appendix F, Table F.3. To gauge the discrepancy in the 2 datasets, we look at the difference between the average F1 scores at different numbers of examples.

The largest difference between our dataset and SQuAD for the 2 prompting methods are both close to 10%, where the difference is larger for the Null prompt. Given this observation, the quality of the dataset can be derived. Considering that SQuAD is a benchmark dataset, the Pula Imvula dataset achieves competitive results. These results indicate the quality of the annotation of the data, where this process is promising with some room for improvement.

Using Figure 9.5, we compared the general performance of the prompt-based methods for the 2 datasets. From the figure, the SQuAD performance as the data size increases, shows more of an exponential pattern. There are more significant improvements for the SQuAD in the performance between each data size. This could indicate that there is a limitation of the results our dataset can achieve in a few-shot setting. Despite this possible limitation, both datasets show the same general trend for the few-shot setting, where as the number of examples increases, the results increase. This reflects that the prompt-based method is as effective on a benchmark dataset as it is on our dataset.

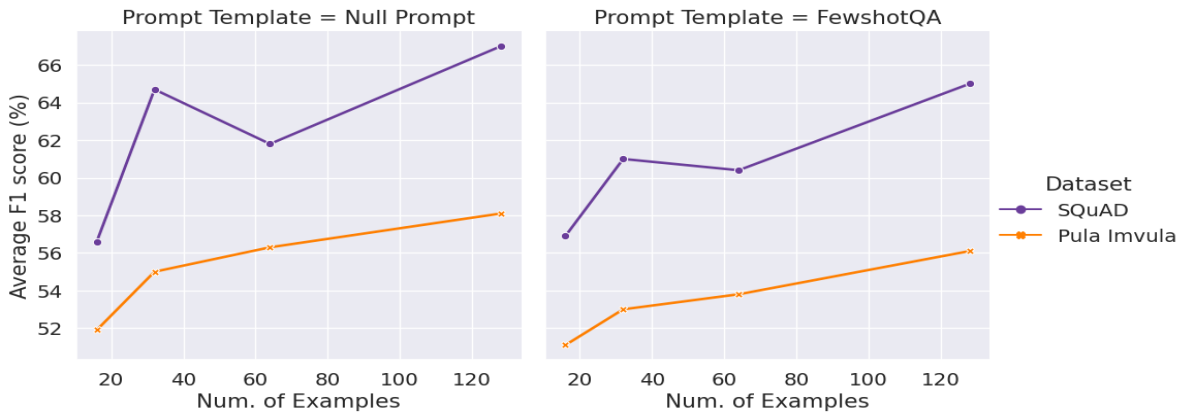

Figure 9.5: The difference in performance of the prompt-based methods on SQuAD and Pula Imvula, at different sample sizes.

Table 9.3: The average F1 scores obtained from the different fine-tuning methods using standard fine-tuning, Null and FewshotQA prompts for SQuAD and Pula Imvula, as a percentage. The highest calculated difference between the F1 scores for each prompt template is highlighted in bold with an asterisk.

Num. of Samples	**Fine-tuning method**	**Average F1 score(%)**		**Difference**
		SQuAD	**Pula Imvula**	
16	Null Prompt	56.6	51.9	4.7
	FewshotQA	56.9	51.1	5.8
32	Null Prompt	64.7	55.0	**9.7***
	FewshotQA	61.0	53.0	8.0
64	Null Prompt	61.8	56.3	5.5
	FewshotQA	60.4	53.8	6.6
128	Null Prompt	67.0	58.1	8.9
	FewshotQA	65.0	56.1	**8.9***

9.6 Summary

The results that are presented in this chapter address the 2^{nd} sub-question of this thesis, which aimed to investigate if few-shot learning methods can be applied to domain-specific question answering. The findings show that the few-shot setting is feasible and that this setting can be further investigated to achieve better results. This method was able to sufficiently capture the domain specific information from the limited amount of data where the first template - null prompting performed best. It is important to note that while the results of the few-shot prompting show promise, it will be necessary to investigate the cause of the plateau in the results. With this information, improvements can be made to the prompting strategy to ensure that significant improvements can be made as the number of examples increases. In Chapter 10, we extend the few-shot setting done in this chapter to investigate its effectiveness for cross-lingual question answering.

Chapter 10

Fine-tuning for Cross-lingual data

The last research sub-question we asked is: *"Based on the results returned from the previous sub-question, what is the feasibility of applying the same model fine-tuning techniques for a multilingual domain specific question answering data in low resource languages ?"*. To answer this question, several experiments were completed to determine the feasibility of the fine-tuning methods that were investigated.

Unlike in the previous chapter 9, the experimental objectives are based on the results obtained from Chapter 9. This chapter discusses the results obtained from the overall experimentation done to answer this research question, and thus answer the main research question.

10.1 Experimental Objectives

As mentioned previously, this experimentation aims to answer the last sub-question. In order to adequately answer this sub-question, the following questions are posed :

- Based on the prompt-based fine-tuning, how transferable are the templates that were initially designed for a monolingual setting ?
- Does adaptation of the original prompts to include language-specific prompting improve the performance?

- How do these language-specific prompting compare to language-agnostic prompting ?

10.2 General Results for Cross-lingual Data

With any fine-tuning of models in low resource languages, if the original model was not trained specifically on the data in that language, the performance of the model is degraded because of this discrepancy. Here, we aim to look at the general trend of how the F1 scores for the cross-lingual setting are for the different dataset sizes. Initially, we used two of the templates as they were designed, template 1 and 2 in Table 8.1 which are based on Null prompting and FewshotQA, respectively.The task in this experimentation was for the question to be asked in English and the context and answer to be in the respective target languages. The highest F1 scores for each of the templates in the three target languages are provided in Table 10.1.

Table 10.1: The highest F1 scores obtained for the 2 different templates (Null and FewshotQA prompts) for the 3 target languages, as a percentage. The highest obtained score for each language has been highlighted in bold with an asterisk.

Language	Prompt	F1(%)
Afrikaans	Null Prompt	**55.9***
	FewshotQA	51.6
Xhosa	Null Prompt	**23.7***
	FewshotQA	20.8
Zulu	Null Prompt	**22.2***
	FewshotQA	18.3

From the results obtained, Afrikaans perform significantly better overall than Zulu and Xhosa, where Zulu performs the worst. It is important to note that for the original model training, Xhosa and Afrikaans were one of the languages included. There is, however, a minor difference between the results for Xhosa as compared to Zulu. This closeness in results for Zulu and Xhosa could be due to these 2 languages being fairly

similar with regards to their linguistic characteristics.

The performance for Afrikaans is competitive to the results obtained for the monolingual dataset in Table 9.1. This significant performance could be due to the linguistic characteristics of the language. As discussed in the correlation analysis done in Section 5.3, there was a high positive correlation between Afrikaans and English for the type token ratio, compared to Zulu and Xhosa, which showed a low correlation. To investigate whether there is a correlation between the correlation coefficients of the type-token ratio and the resultant F1 score, we calculate the Pearson coefficient. The resulting Pearson coefficient is **0.985**. However, the initial correlation analysis only looked at a simple comparison between the languages.

In [29], one of their research questions was to investigate the underlying factors that affect the cross-lingual performance of a fine-tuned model. One of the factors that showed a positive correlation to the cross-lingual results obtained is the language similarity between the target and the source language based on typological and phylogenetic features. To complete this analysis, the similarity of the languages was calculated using the lang2vec utility [25]. This utility consists of representing languages as vectors based on the language typology, geography, and phylogeny databases.

Using the same method [29], we compute the similarity between each of the target languages to the source language based on 4 of the feature vectors: syntax, family group, phonology, and geography. The resultant similarity is the average of the similarity computed for each of the different vectors. Detailed similarities for the vectors can be seen in Appendix G, Table G.4. The Pearson correlation coefficient is calculated between the average similarity score and the highest F1 scores obtained for each of the prompting methods. The resulting coefficient is **0,99** for both prompting methods.

The 2 Pearson coefficients calculated (0.985 and 0.99) indicate that the F1 scores for the cross-lingual setting are highly dependent on the similarity between the target language and the source language. Therefore, there is a limitation to the final F1 score

that can be obtained for this dataset. Despite this, the overall general results are similar to the English results, where as the number of examples increases, only slight gains are made of the average F1 score.

10.3 Language Sensitivity

The general trend of the cross-lingual data as the number of examples used to fine-tune the model increases is provided in Figure 10.1. By observing the overall pattern of the F1 score, rather than focusing on the specific values, it becomes apparent that there are slight variations between template 1 (null prompt) and template 2 (FewshotQA). Overall, for all 3 languages, the null prompting performs slightly better than FewshotQA, following the same trend as the English results. The null prompt can be considered task agnostic and language agnostic, meaning that this template can be applied to any task and language. Its better performance can be due to the fact that it is language agnostic.

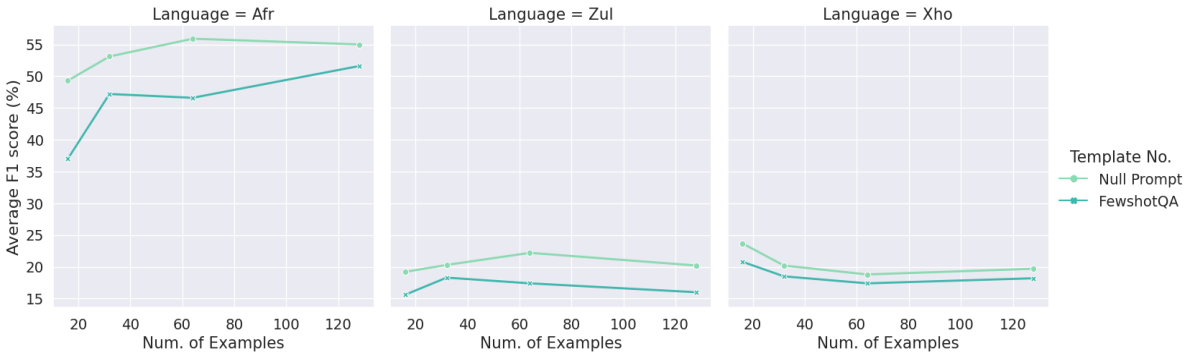

Figure 10.1: The F1 score obtained for the different designed prompts - Null and FewshotQA prompting

From the results obtained in Figure 10.1, the difference between the prompt results shows that the prompt-based method could be sensitive to the language used for the prompting. To further analyse this, we take the FewshotQA prompt and add language-specific prompting. In this prompt, we use Google Translate to translate the prefixes of the prompt : “Context” and “Answer” to the respective target languages. The final

results obtained from this prompt are compared to the original FewshotQA prompt in figure 10.2.

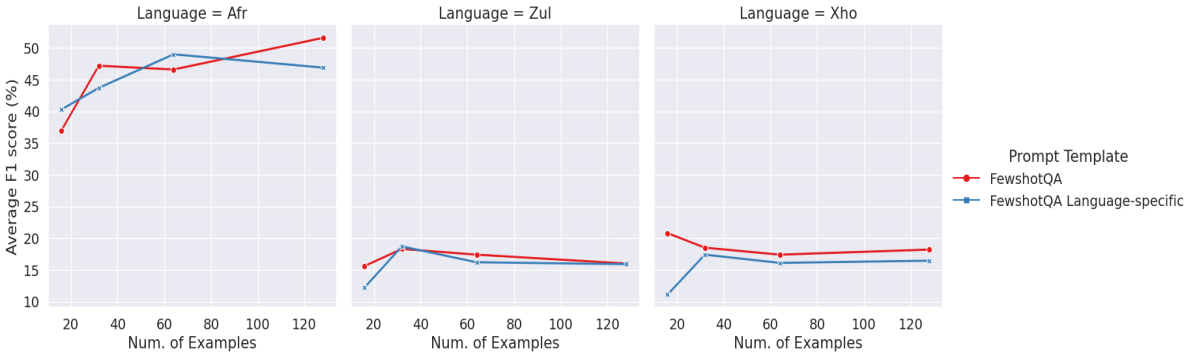

Figure 10.2: The F1 score comparison between the language specific FewshotQA prompting

From the results shown in Figure 10.2, we can see that there is a slight difference between the Original FewshotQA and the language specific one. The language-specific prompt generally performs slightly worse but is still competitive. Thus, highlighting that even though the prompts are language-specific in terms of the target language there is no major difference with the prompts in the source language.

Overall, this showed that there is a direct relationship between the language of the prompts. Using non-English prompting, an effective prompt-based method can be developed and used effectively to solve the problem at hand, that is, a prompt-based method for LRLs. This inconsistency in the differences in the template results for the 3 languages also showed that there is a level of lack of robustness with the prompt-based fine-tuning method. Therefore, it is necessary to carefully craft and investigate the effects and necessary language-specific attributes to make the templates more suitable for the languages.

10.4 Summary

The results that are presented in this chapter address the 3^{rd} sub-question of this thesis which aimed to investigate if the few-shot learning methods can be applied in a cross-lingual setting. The findings show that the few-shot setting is feasible, and this setting can be further investigated to achieve better results. It is important to note that while the results of the few-shot prompting show promise, it will be necessary to investigate the cause of the plateau in the results. Another aspect which needs to be investigated is the language sensitivity of the prompt-based methods for a cross-lingual setting. With this information, improvements can be made to the prompting strategy to ensure that significant improvements can be made as the number of examples increases.

Chapter 11

Conclusions

This study aimed to understand and test the feasibility of known few-shot learning strategies for a domain-specific and cross-lingual question answering dataset. In order to fulfil this, a text corpora which satisfied the desired data properties was created from publicly available text data. Articles about agro-information in South Africa was collected in multiple languages from Pula Imvula magazine. The contributions of this study were broken down into three main objectives. An automated annotation pipeline(Chapter 6) was created and tested on the collected data. The results of these data were then analysed (Chapter 7).

Furthermore, the resultant annotated dataset was then used for the main experiment which was broken down into 2 parts that made up the two other objectives of this study. The first part attempted to understand the applicability of few-shot learning for domain-specific data. The results of this experimentation were discussed in Chapter 9. The second part of the experimentation is to extend the results obtained in the previous chapter and apply the results of the experimentation for a cross-lingual setting. The final results are discussed in further detail in Chapter 10.

The novel contributions of this study include a question answering dataset that is based on agro-information and is available in 4 languages(English, Afrikaans, Zulu and Xhosa). Furthermore, it confirmed the applicability of few-shot prompt-based fine-tuning

for cross-lingual domain-specific data. In this chapter, we provide a comprehensive summary of the conclusions made for this study in Section 11.1 and the future direction that can be followed based on the results of this work 11.2.

11.1 Summary of Conclusions

Our findings have demonstrated that the use of prompt-based fine-tuning is a promising approach for cross-lingual domain-specific question answering. This approach is particularly useful for a few-shot learning setting. To get to this, we also had to create a dataset that is of adequate quality. Based on the quality of the dataset we created, it is concluded that using large language models to annotate data is a promising direction. For the main research question to be answered, we answered the 3 sub-questions but concluding on the main findings that contribute to these sub-questions.

11.1.1 Sub-question 1

> *"What automated process can be used to effectively create a question answering dataset based on agro-information in multiple languages?"*

The GPT model has shown that it can be used to create a dataset that is sufficient for the downstream task of question answering. From the analysis that was performed, it showed that it was able to capture the essential textual content of the articles and generate good question-answer pairs. This approach shows promise, despite the limitations that were present. With further refinement of the dataset through heuristic filtering, some of the limitations could be dealt with. It was observed that the model tended to be repetitive and redundant with the type of question answer pairs generated. Overall, a novel dataset based on agro-information was successfully created through an automated approach.

11.1.2 Sub-question 2

> *"What is the feasibility of applying the model fine-tuning techniques for monolingual (English) domain-specific question answering data?"*

In the context of using domain-specific, limited data in a few-shot setting, the few-shot methods that were investigated yielded competitive results for the task of question answering. From the initial set of examples, the model was able to score over 0.5 F1 score for the dataset, and minimal improvements were made to the F1 score as the number of examples increased. This indicated that due to the limited scope of the dataset , the initial data points encapsulated the domain-specific data well. Furthermore, when it came to the different prompting strategies, there was no clear significant difference between the different prompting methods. However, the results from the prompting method showed early plateauing as the number of examples increased. This could imply that further refinement of the data or prompting strategies may be needed to understand the cause of the plateau.

11.1.3 Sub-question 3

> *"Based on the results returned from the previous sub-question, what is the feasibility of applying the same model fine-tuning techniques for a multilingual domain-specific question answering data in low resource languages ?"*

In the context of using limited cross-lingual data in a few-shot setting, the few-shot methods that were investigated yielded promising results for the task of question answering. When comparing the general trend of the results with the English results, it was seen that the cross-lingual setting followed the same trend. Overall, Afrikaans performed relatively well compared to Zulu and Xhosa. From the initial set of examples, the model was able to score F1 scores between 0.4 for Afrikaans and 0.2 for Zulu and Xhosa with minimal improvements made to the F1 score as the number of examples increased. This indicated that due to the limited scope of the dataset, the initial data points encapsulated the domain-specific data well. Furthermore, when it came to the different prompting strategies, there was no clear significant difference between the different prompting methods. However, the results from the prompting method showed early plateauing as the number of examples increased. This could imply that further refinement of the data or prompting strategies may be needed to understand the cause of the plateau. The closeness in the prompting results between the different templates

showed that in general the few-shot setting is applicable to this type of dataset, but there it is important to further investigate the different prompting methods.

11.2 Future Work

While there were worth while contributions made in this study, there are still several avenues that can be explored in future work to expand on the results obtained in this thesis. The following is a summary of the main recommendations that can be made:

1. Investigating more efficient ways to use models like GPT3 to create annotated datasets. With the current approach that was used, the model showed signs of redundancy and repetitiveness when it came to the output. This limited the diversity in the questions and answers that were generated. Thus, exploring ways to get better results through prompt engineering can maximise the potential of these models.

2. Investigating frameworks that can be used to assess the quality of the outputs produced without human evaluation. Through the design of this pipeline, only simple heuristics were used to filter out the output that was considered redundant or ambiguous. Further investigation can be done on what specific characteristics contribute to a high-quality question-answer dataset and how these characteristics can be used to design the framework.

3. Extending the current dataset to have N-way parallel alignments between the different target languages. With such a dataset, the tansferability between the languages can be investigated.

4. Investigating further more efficient language-specific prompting for the low resource languages. The effect of linguistic-specific prompting for the few-shot setting can be used to create a better prompting method for low resource languages.

5. Investigating the cross-lingual transfer capabilities of the prompt-based methods discussed in this study, i.e., how do these methods fair when a model is fine-tuned

on one source language and evaluated on limited data available in low resource languages.

6. In an ideal setting, if such models were to be applied to industry, the size of the model will be of interest. These few-shot settings can be extended to investigating how they perform for smaller models.

Revisiting the main objective of this thesis in Chapter 1, this study aimed to examine whether prompt-based fine-tuning techniques are feasible for a multilingual domain-specific text in the context of agro-information question answering for LRL. Furthermore, to achieve this, in this study an annotated dataset needed to be created through an automated approach. Taking into account the summary mentioned above of the conclusions that have been made in this study with respect to research questions and objectives, prompt-based fine-tuning is a compelling alternative to fine-tuning models on limited data in terms of domain-specific data and data available in low resource languages. As the field grows, this approach can be used to play a crucial role in fine-tuning models in a limited data setting, and this allows for the academic results obtained in this field to be transferred to industry use.

Bibliography

[1] Idris Abdulmumin, Auwal Abubakar Khalid, Shamsuddeen Hassan Muhammad, Ibrahim Said Ahmad, Lukman Jibril Aliyu, Babangida Sani, Bala Mairiga Abduljalil, and Sani Ahmad Hassan. Leveraging closed-access multilingual embedding for automatic sentence alignment in low resource languages. 2023.

[2] A. Andrenucci and E. Sneiders. Automated question answering: review of the main approaches. In *Third International Conference on Information Technology and Applications (ICITA'05)*, volume 1, pages 514–519 vol.1, 2005.

[3] Mikel Artetxe and Holger Schwenk. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, November 2019. arXiv:1812.10464 [cs].

[4] Akari Asai, Xinyan Yu, Jungo Kasai, and Hannaneh Hajishirzi. One question answering model for many languages with cross-lingual dense passage retrieval, 2021.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. arXiv:2005.14165 [cs].

[6] Marco Antonio Calijorne Soares and Fernando Silva Parreiras. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University - Computer and Information Sciences*, 32(6):635–646, July 2020.

[7] Rakesh Chada and Pradeep Natarajan. FewshotQA: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models, October 2021. arXiv:2109.01951 [cs].

[8] Gobinda G. Chowdhury. *Introduction to Modern Information Retrieval*. Facet Publishing, 2010. Google-Books-ID: cN4qDgAAQBAJ.

[9] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *CoRR*, abs/2003.05002, 2020.

[10] Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, January 1996.

[11] Manmita Devi and Mohit Dua. ADANS: An agriculture domain question answering system using ontologies. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 122–127, May 2017.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. arXiv:1810.04805 [cs].

[13] NLP for Question Answering. Evaluating qa: Metrics, predictions, and the null response. Online, 2020. Last accessed 02 May 2023.

[14] Sharvari Gaikwad, Rohan Asodekar, Sunny Gadia, and Vahida Z. Attar. AGRI-QAS question-answering system for agriculture domain. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1474–1478, August 2015.

[15] Tianyu Gao, Adam Fisch, and Danqi Chen. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online, August 2021. Association for Computational Linguistics.

[16] Ken Giller, Thomas Delaune, João Silva, Mark Van Wijk, Jim Hammond, Katrien Descheemaeker, Gerrie W.J. Ven, A.G.T. Schut, G. Taulya, Regis Chikowo, and Jens Andersson. Small farms and development in sub-saharan africa: Farming for food, for income or for lack of better options? *Food Security*, 13, 10 2021.

[17] K. S. D. Ishwari, A. K. R. R. Aneeze, S. Sudheesan, H. J. D. A. Karunaratne, A. Nugaliyadde, and Y. Mallawarrachchi. Advances in Natural Language Question Answering: A Review, April 2019. arXiv:1904.05276 [cs].

[18] Shinice Jackson and Derek Yu. Re-examining the Multidimensional Poverty Index of South Africa. *Social Indicators Research*, 166(1):1–25, February 2023.

[19] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, January 2020.

[20] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[21] Bernhard Kratzwald and Stefan Feuerriegel. Putting question-answering systems into practice: Transfer learning for efficient domain customization. *ACM Trans. Manage. Inf. Syst.*, 9(4), feb 2019.

[22] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee,

Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 08 2019.

[23] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, October 2019. arXiv:1910.13461 [cs, stat].

[24] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating cross-lingual extractive question answering. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online, July 2020. Association for Computational Linguistics.

[25] Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain, April 2017. Association for Computational Linguistics.

[26] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 55(9):195:1–195:35, January 2023.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019. arXiv:1907.11692 [cs].

[28] Robert L. Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models, July 2021. arXiv:2106.13353 [cs].

[29] Bolei Ma, Ercong Nie, Helmut Schmid, and Hinrich Schütze. Is Prompt-Based Finetuning Always Better than Vanilla Finetuning? Insights from Cross-Lingual Language Understanding, July 2023.

[30] Theresa Mazarire, Ratshiedana Eugene, Adolph Nyamugama, and Elhadi Adam. Exploring machine learning algorithms for mapping crop types in a heterogeneous agriculture landscape using Sentinel-2 data. A case study of Free State Province, South Africa. *South African Journal of Geomatics*, 9:333–347, September 2020.

[31] Walter Mupangwa, Lovemore Chipindu, Isaiah Nyagumbo, Siyabusa Mkuhlani, and Givious Sisito. Evaluating machine learning algorithms for predicting maize yield under conservation agriculture in Eastern and Southern Africa. March 2020.

[32] Daniel Mutembesa, Christopher Omongo, and Ernest Mwebaze. Crowdsourcing real-time viral disease and pest information. a case of nation-wide cassava disease surveillance in a developing country, 2019.

[33] Odunayo Ogundepo, Tajuddeen R. Gwadabe, Clara E. Rivera, Jonathan H. Clark, Sebastian Ruder, David Ifeoluwa Adelani, Bonaventure F. P. Dossou, Abdou Aziz DIOP, Claytone Sikasote, Gilles Hacheme, Happy Buzaaba, Ignatius Ezeani, Rooweither Mabuya, Salomey Osei, Chris Emezue, Albert Njoroge Kahira, Shamsuddeen H. Muhammad, Akintunde Oladipo, Abraham Toluwase Owodunni, Atnafu Lambebo Tonja, Iyanuoluwa Shode, Akari Asai, Tunde Oluwaseyi Ajayi, Clemencia Siro, Steven Arthur, Mofetoluwa Adeyemi, Orevaoghene Ahia, Anuoluwapo Aremu, Oyinkansola Awosan, Chiamaka Chukwuneke, Bernard Opoku, Awokoya Ayodele, Verrah Otiende, Christine Mwase, Boyd Sinkala, Andre Niyongabo Rubungo, Daniel A. Ajisafe, Emeka Felix Onwuegbuzia, Habib Mbow, Emile Niyomutabazi, Eunice Mukonde, Falalu Ibrahim Lawan, Ibrahim Said Ahmad, Jesujoba O. Alabi, Martin Namukombo, Mbonu Chinedu, Mofya Phiri, Neo Putini, Ndumiso Mngoma, Priscilla A. Amuok, Ruqayya Nasir Iro, and Sonia Adhiambo. AfriQA: Cross-lingual Open-Retrieval Question Answering for African Languages, May 2023. arXiv:2305.06897 [cs].

[34] Sunday Paul Chinazo Onwuegbuchulam. A Capability Approach assessment of poverty in the sociopolitical history of South Africa/KwaZulu-Natal. *Journal of Poverty*, 22(4):287–309, July 2018.

[35] Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. Transformers in the Real World: A Survey on NLP Applications. *Information*, 14(4):242, April 2023. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[36] Brilliant M. Petja, Richard R. Ramugondo, and A. Edward Nesamvuni. Using remote sensing and geographic information system for prioritization of areas for site specific agricultural development in Limpopo Province, South Africa. In *2009 IEEE International Geoscience and Remote Sensing Symposium*, volume 5, pages V–397–V–400, July 2009. ISSN: 2153-7003.

[37] Alec Radford and Karthik Narasimhan. Improving Language Understanding by Generative Pre-Training. 2018.

[38] Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, September 2023. arXiv:1910.10683 [cs, stat].

[40] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad, 2018.

[41] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[42] Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. Few-Shot Question Answering by Pretraining Span Selection, June 2021. arXiv:2101.00438 [cs].

[43] Grain SA. Pula/imvula. Online. Last accessed 22 March 2023.

[44] Sunita Sarawagi. Information Extraction. *Foundations and Trends® in Databases*, 1(3):261–377, November 2008. Publisher: Now Publishers, Inc.

[45] Timo Schick and Hinrich Schütze. Few-Shot Text Generation with Pattern-Exploiting Training, October 2021. arXiv:2012.11926 [cs].

[46] Timo Schick and Hinrich Schütze. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online, June 2021. Association for Computational Linguistics.

[47] Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training, 2021.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[49] Barack W. Wanjawa, Lilian D. A. Wanzare, Florence Indede, Owen McOnyango, Lawrence Muchemi, and Edward Ombui. KenSwQuAD – A Question Answering Dataset for Swahili Low Resource Language, July 2023. arXiv:2205.02364 [cs].

[50] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. pages 2013–2018, 01 2015.

[51] Dr Amol V. Zade. QASAD: QUESTION ANSWERING SYSTEM FOR AGRICULTURAL DOMAIN A SEMANTIC INTERFACE FOR INDIAN FARMERS. *International Engineering Journal For Research & Development*, 6(NCTSRD):6–6, December 2021. Number: NCTSRD.

Appendix A

Article Title Matches

A.1 Summary

In this appendix, we show some of the examples of the translations that were retrieved for the article titles. For each of the matches, we categorise the matches according to the similarity. The translations obtained were matched with the English title and the cosine similarity was used to determine the quality of the matches.

Table A.1: Examples of the article titles that have been matched in the different languages according to their category

Language	**Category**	**English Title**	**Untranslated Title**	**Translated Title**
Afrikaans	Full Match	Do I use FOREIGN CAPITAL or not?	Gebruik ek vreemde kapitaal of nie	Do I use strange capital or not
Xhosa	Full Match	Grow your FARMING BUSINESS	Khulisa ishishini lakho lokufam	Grow your farming business
	Partial Mismatch - High Similarity	Are you a STANDOUT LEADER?	Ingaba uyinkokeli yenene	Are you a real leader
	Partial Mismatch - Low Similarity	Manage the good years well	Yilawule kakuhle iminyaka emihle	Take control of the good years
Zulu	Full Match	What has changed?	Yini eguqukile	What has changed
	Partial Mismatch - High Similarity	Keep rainfall records to reduce risk	Gcina amarekhodi emvula ukunciphisa ubungozi	Maintain rain records to reduce risk

Appendix B

Annotation Few-shot Prompt Design

```
input_prompt = f""" Given a block of text and the title of the text, first, read the text and establish the main topics of the text.
                    Based on these topics, extract a list of key keywords or short phrases from the text that capture the topics.
                    Return these key keywords and short phrases in a list that is in a concise format where each of the keywords is
                    separated by a semicolon (;) for the corresponding text based on the examples.
                    Example 1 :
                    Title: "1983 Cricket World Cup"
                    Text: The final of the 1983 Cricket World Cup was played between India and the West Indies at Lord's on 25 June 1983.
                    This was the third consecutive World Cup final appearance for the West Indies, having won the last two Cricket World Cups.
                    India won the Cricket World Cup which was the 3rd edition of the Cricket World Cup tournament.
                    Keywords: 25 June 1983;  at Lord's ; 1983 Cricket World Cup; between India and West Indies; 3rd edition; having lost two
                    Cricket World Cups
                    Example 2 :
                    Title: "Google"
                    Text:  Google was founded on September 4, 1998, by computer scientists Larry Page and Sergey Brin while they were PhD students at
                    Stanford University in California.  Together they own about 14% of its publicly listed shares and control 56% of its
                    stockholder voting power through super-voting stock. The company went public via an initial public offering (IPO) in 2004.
                    Keywords: Google was founded on September 4, 1998; computer scientists Larry Page and Sergey Brin; were PhD students, Stanford
                    University in California  """
input = f"""
            Title: {title}
            Text: {context}
            Keywords:
        """
```

Figure B.1: Full keyword extraction prompt provided to ChatGPT

```
input_prompt = f""" Given a context and a key phrase that highlights a main theme from this context,
                    generate a relevant factual question where the answer to the question is based on the
                    key phrase.  From the generated question, provide a concise and direct answer to the
                    question only using the information from the text. Ensure that the answer is an exact
                    extract from the paragraph and is written as it is in the paragraph without any changes.
                    Return the question  and the answer for the corresponding text based on the examples provided.


                    Example 1 :
                    Context: The final of the 1983 Cricket World Cup was played between India and the West
                    Indies at Lord's on 25 June 1983. This was the third consecutive World Cup final appearance
                    for the West Indies, having won the last two Cricket World Cups. India won the Cricket World
                    Cup which was the 3rd edition of the Cricket World Cup tournament.
                    Key phrase:  India and the West Indies
                    Question: Who played in the final 1983 Cricket World Cup?
                    Answer : was played between India and the West Indies

                    Example 2 :
                    Context: Google was founded on September 4, 1998, by computer scientists Larry Page and Sergey
                    Brin while they were PhD students at Stanford University in California. Together they own about
                    14% of its publicly listed shares and control 56% of its stockholder voting power through
                    super-voting stock. The company went public via an initial public offering (IPO) in 2004.
                    Key phrase: On September 4, 1998.
                    Question: When was Google founded?
                    Answer : was founded on September 4, 1998,
                """
input = f"""
            Context: {context}
            Key phrase: {keyphrase}
            Question:
            Answer:
        """
```

Figure B.2: Full question and answer generation prompt provided to ChatGPT

B.1 Summary

In this appendix, we provide the final full prompts used for OpenAI to generate the final dataset. The prompts include few-shot examples that are included in the prompting. Figure B.1 provides the complete prompt for generating keywords from a context(article paragraph) provided to the model, and Figure B.2 shows the prompt to generate the questions-answer pairs.

Appendix C

Phrase Generation

```
Die
Die beste
Die beste tipe
Die beste tipe instandhouding
Die beste tipe instandhouding is
Die beste tipe instandhouding is voorkomende
Die beste tipe instandhouding is voorkomende instandhouding.
beste
beste tipe
beste tipe instandhouding
beste tipe instandhouding is
beste tipe instandhouding is voorkomende
beste tipe instandhouding is voorkomende instandhouding.
tipe
tipe instandhouding
tipe instandhouding is
tipe instandhouding is voorkomende
tipe instandhouding is voorkomende instandhouding.
instandhouding
instandhouding is
instandhouding is voorkomende
instandhouding is voorkomende instandhouding.
is
is voorkomende
is voorkomende instandhouding.
voorkomende
voorkomende instandhouding.
instandhouding.
```

Figure C.1: Example of all the phrases generated using the phrase generation algorithm where the input is ”Die beste tipe instandhouding is voorkomende instandhouding.”

C.1 Summary

In order to try to get the minimal span for an answer in the target languages, the phrase from the sentence which answers the question needs to be determined. To get all the possible phrases, we developed a novel phrase generation algorithm 6.2. In this Appendix, we provide an example of some of the output produced by the algorithm.

Appendix D

Annotation Evaluation Guideline

D.1 Summary

For any annotation that is made, an annotation guideline is created to assess the quality of the annotations. In this appendix, we show the evaluation flow chart which determines the final rating for the questions D.1 and answers D.2 respectively.

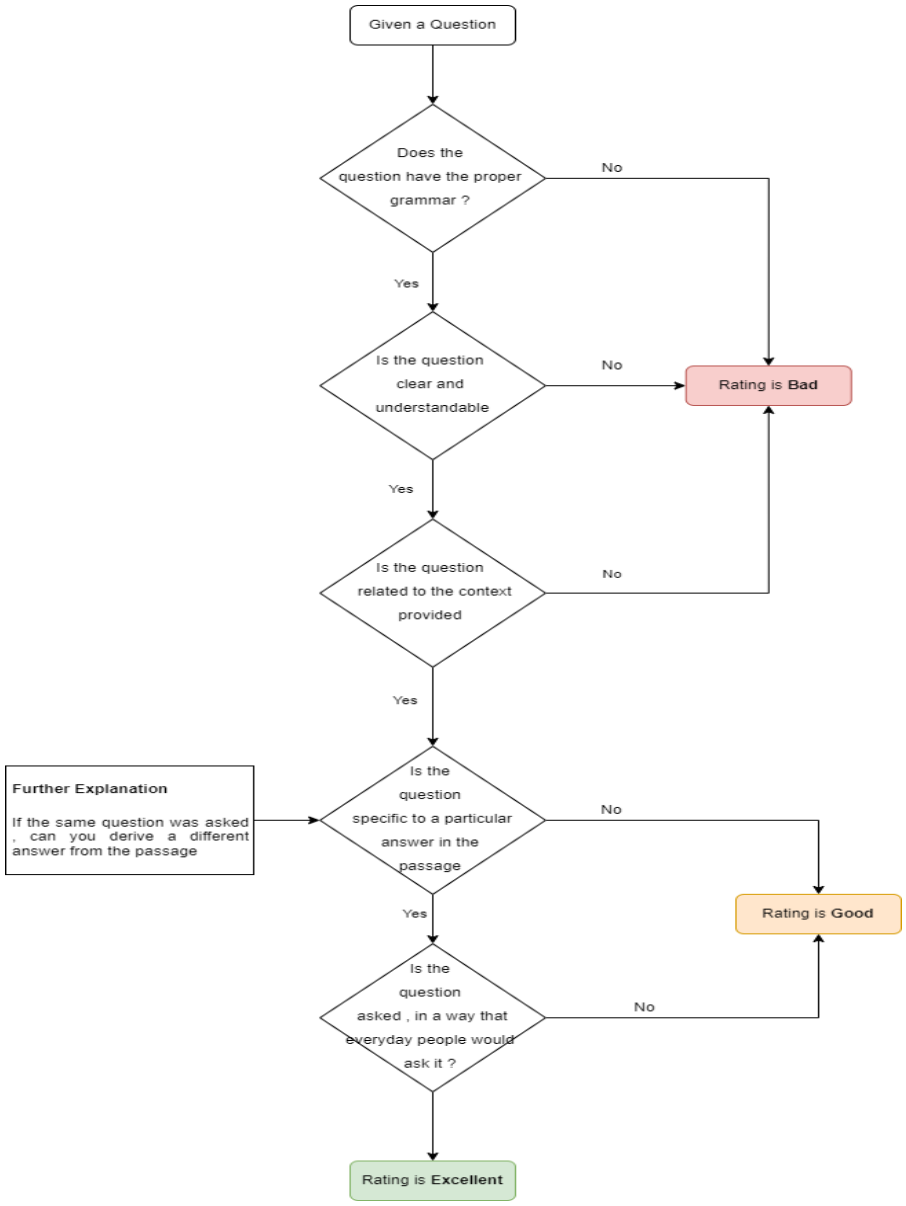

Figure D.1: The flow chart used to assess the quality of the questions

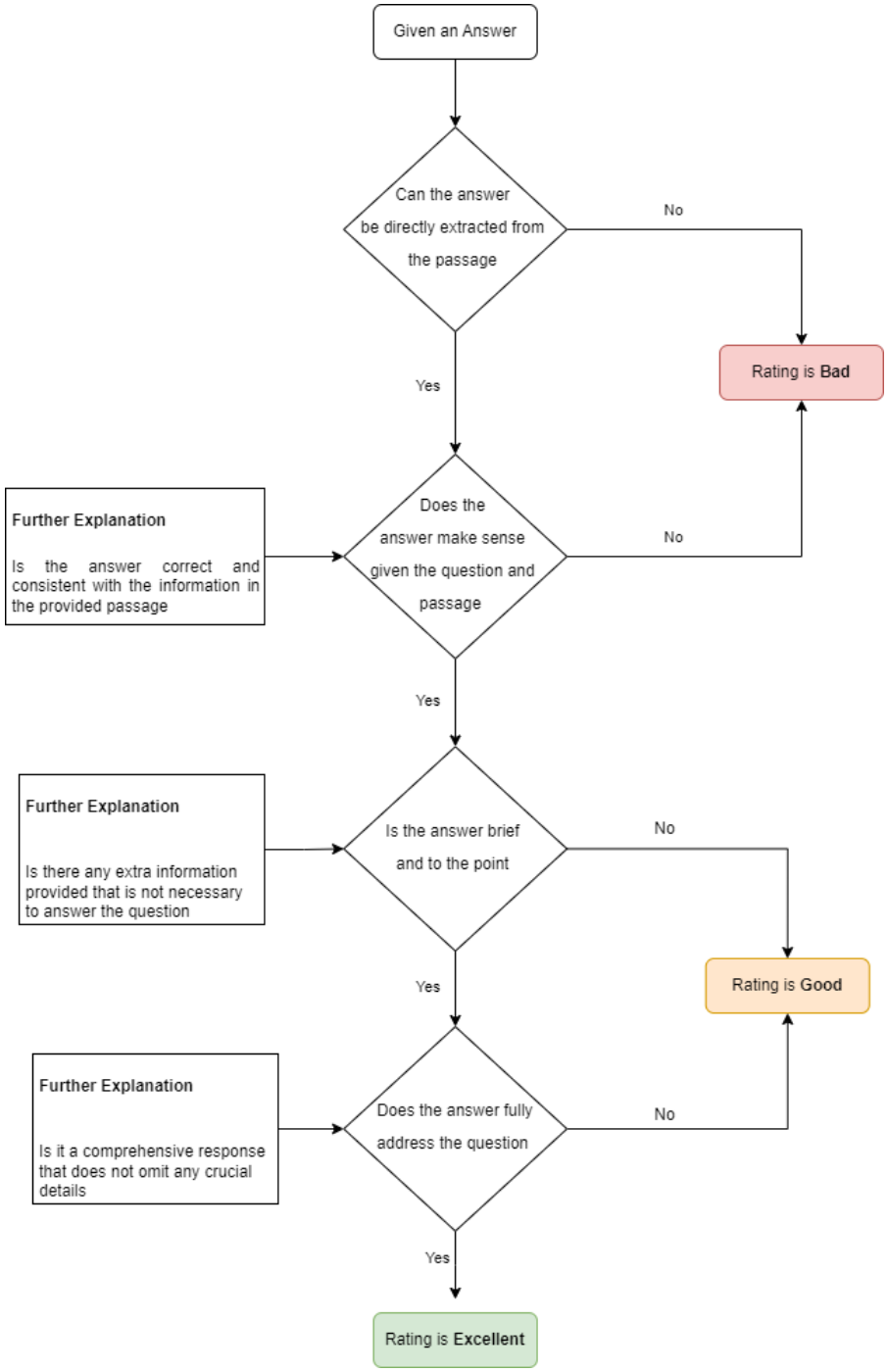

Figure D.2: The flow chart used to assess the quality of the answers

Appendix E

Detailed Dataset Split

Table E.1: Dataset split distribution for the different Languages

Language	Subset	Sample %	Num. of Samples
English	Training	40	2111
	Development	40	2110
	Testing	20	1056
Afrikaans	Training	40	1311
	Development	40	1311
	Testing	20	656
Xhosa	Training	40	1084
	Development	40	1084
	Testing	20	542
Zulu	Training	40	1094
	Development	40	1094
	Testing	20	548

E.1 Summary

In this appendix, we provide details of the final size of the different subsets of the dataset in the different languages.

Appendix F

Detailed Results of the Experimentation on the Domain-specific English Dataset

F.1 Summary

This appendix provides the detailed results obtained for the different experimentation done for the domain-specific fine-tuning. The results are divided into the results obtained for the different templates designed for the prompt-based method and the different fine-tuning methods. The results are obtained for different subsets of the data and then the average of the F1 score is calculated for various sizes of the dataset.

Table F.1: The average F1 scores obtained from the few-shot setting using different prompts, as a percentage

Num. of Examples	Template	Sample F1 Score					Average
		1	2	3	4	5	
16	1	75.5	58.1	46.7	37.2	41.8	51.9
	2	74.0	43.4	45.1	51.9	41.2	51.1
	3	62.9	49.6	42.5	42.1	43.7	48.2
	4	69.5	48.4	47.4	47.0	42.1	50.9
	5	58.3	39.0	48.3	46.6	41.8	46.8
	6	72.6	43.7	47.0	41.8	43.4	49.7
32	1	56.9	50.7	55.3	58.5	53.6	55
	2	65.3	51.8	58.6	43.1	46.2	53.0
	3	61.5	51.6	57.1	50.4	37.9	51.7
	4	57.2	48.8	56.5	48.7	47.2	51.7
	5	53.5	52.1	55.3	46.3	51.5	51.7
	6	60.2	51.5	52.9	47.6	51.5	52.7
64	1	61.9	59.5	52.8	56.5	51.0	56.3
	2	59.5	52.2	56.1	52.1	49.2	53.8
	3	55.1	49.1	52.0	53.2	48.5	51.6
	4	57.2	49.8	53.7	53.5	48.6	52.6
	5	54.9	51.9	50.7	50.5	46.0	50.8
	6	59.8	50.6	57.0	54.3	50.4	54.4
128	1	62.2	60.4	56.0	60.4	51.6	58.1
	2	58.1	60.4	59.2	51.4	51.5	56.1
	3	56.6	52.7	57.4	56.8	52.2	55.1
	4	61.5	61.1	55.9	55.0	53.1	57.3
	5	61.0	59.4	56.3	49.4	48.9	55.0
	6	57.6	57.5	56.3	52.1	51.6	55.0

Table F.2: The average F1 scores obtained from the few-shot setting using different prompts, as a percentage

Fraction Sampled(%)	Fine-tuning Method	Sample F1 Score					Average
		1	2	3	4	5	
0.5	Standard	31.9	33.9	36.7	35.7	13.6	30.4
	Null Prompt	53.9	37.5	36.6	33.8	37.5	39.9
	FewshotQA	60.1	55.8	33.2	35.4	47.7	46.4
1	Standard	58.1	29.6	43.6	34.5	42.0	41.6
	Null Prompt	63.8	53.0	71.3	55.5	47.0	58.1
	FewshotQA	54.1	59.0	51.6	42.0	49.7	51.3
2.5	Standard	49.2	55.3	54.2	53.4	56.9	53.8
	Null Prompt	64.8	56.0	53.6	57.6	52.3	56.9
	FewshotQA	60.2	57.2	54.4	52.5	52.4	55.3
5	Standard	65.1	60.6	62.7	65.4	59.2	62.6
	Null Prompt	59.0	58.4	58.0	59.7	51.3	57.3
	FewshotQA	56.1	58.0	57.1	59.4	46.43	55.4
10	Standard	67.2	69.0	66.2	65.0	-	66.9
	Null Prompt	62.7	60.3	59.1	58.1	-	60.1
	FewshotQA	61.7	60.3	57.9	56.3	-	59.1

Table F.3: The average F1 scores obtained from the few-shot setting using the SQuAD dataset, as a percentage

Num. of Examples	Template	Sample F1 Score					Average
		1	2	3	4	5	
16	Null Prompt	49.7	57.3	63.3	56.3	56.3	56.6
	FewshotQA	47.6	59.2	63.3	59.7	64.3	56.9
32	Null Prompt	67.3	70.3	62.1	59.7	64.3	64.7
	FewshotQA	69.3	62.8	60.1	61.4	51.5	61.0
64	Null Prompt	56.2	60.9	58.4	66.3	67.0	61.8
	FewshotQA	50.0	63.2	60.7	64.7	63.4	60.4
128	Null Prompt	65.2	70.6	66.6	72.0	61.0	67.0
	FewshotQA	67.1	64.6	62.8	69.2	61.1	65.0

Appendix G

Detailed Results of the Experimentation on the Cross-lingual Domain-specific Dataset

G.1 Summary

This appendix provides the detailed results obtained for the different experimentation done for the cross-lingual domain-specific prompt-based fine-tuning. The results for each target language is provided where different subsets of the data are recorded, and the averages for the different templates are recorded.

Table G.1: The average F1 scores obtained from the few-shot setting using different prompts for the cross-lingual English-Afrikaans, as a percentage

Num. of Examples	**Template**	**Sample F1 Score**			**Average**
		1	**2**	**3**	
16	1	47.2	59.1	41.6	49.3
	2	33.1	42.3	35.7	37.0
	3	31.5	40.3	49.1	40.3
32	1	56.5	54.8	47.9	53.1
	2	43.7	51.0	47.0	47.2
	3	43.8	42.9	44.3	43.7
64	1	57.9	50.9	58.8	55.9
	2	47.7	40.9	51.3	46.6
	3	52.2	42.7	52.1	49
128	1	54.5	56.3	54.2	55
	2	52.3	53.6	48.8	51.6
	3	49.2	46.4	45.0	46.9

Table G.2: The average F1 scores obtained from the few-shot setting using different prompts for the cross-lingual English-Xhosa, as a percentage

Num. of Examples	**Template**	**Sample F1 Score**			**Average**
		1	**2**	**3**	
16	1	29.1	25.7	16.2	23.7
	2	24.1	18.8	19.5	20.8
	3	12.5	9.4	11.4	11.1
32	1	25.3	13.9	21.5	20.2
	2	22.9	18.3	14.3	18.5
	3	16.8	18.5	17.0	17.4
64	1	18.7	19.2	18.6	18.8
	2	18.0	17.3	16.8	17.4
	3	18.4	14.7	15.3	16.1
128	1	18.22	21.2	-	19.71
	2	16.5	19.9	-	18.2
	3	14.5	18.4	-	16.45

Table G.3: The average F1 scores obtained from the few-shot setting using different prompts for the cross-lingual English-Zulu, as a percentage

Num. of Examples	Template	Sample F1 Score			Average
		1	2	3	
16	1	20.2	19.0	18.3	19.2
	2	17.0	15.6	14.2	15.6
	3	12.6	14.3	9.8	12.2
32	1	24.1	21.1	15.8	20.3
	2	22.4	17.3	15.2	18.3
	3	19.4	20.7	16.0	18.7
64	1	25.2	21.0	20.4	22.2
	2	19.9	15.8	16.6	17.4
	3	16.7	16.2	15.7	16.2
128	1	20.4	20.0	-	20.2
	2	15.1	16.8	-	16.0
	3	-	15.9	-	15.9

Table G.4: Detailed summary of the different cosine similarity scores for the different vectors based on lang2vec utility

Target Language	Syntax	Phonology	Family	Geography	Average
Afr	0.82	0.69	0.50	0.87	0.72
Xho	0.48	0.70	0.00	0.85	0.51
Zul	0.45	0.66	0.00	0.86	0.49