Enhancing Agriculture QA Models Using Large Language Models

Xiaoyan He*

Mathematics, University of California San Diego, La Jolla, USA

Abstract. Agriculture is a complex process that requires a great deal of knowledge and experience. Yet, the complexity and often chaotic nature of web data presents a considerable challenge in obtaining suitable results. Given these production requirements, there is an urgent need to develop a model that enables machine reading comprehension and caters to automatic question-answering scenarios within the scope of agricultural production. In this paper, we construct a dataset for the experiments of document QA in agricultural scenarios. We import two model (ask-my-pdf and chat-pdf) as our baseline and use them to do the single and multiple document QA task. Then, we proposed several methods to improve the performance of the model in agriculture scenarios. At the end of the experiment, we achieved 33.3% improvement in F1 score compared to baseline and 92.8% overall answer accuracy in single document QA. For our final multiple documents QA model, we achieved 53% improvement in F1 score compared to baseline and 83.3% overall answer accuracy in the task.

1 Introduction

1.1 Research Background

Agricultural production is a complex process that requires a great deal of knowledge and experience. To get the most updated knowledge and experience, agricultural practitioners are veering away from traditional methods of consulting with experts on agricultural techniques, favoring web-based resources instead. Yet, the complexity and often chaotic nature of web data presents a considerable challenge in obtaining suitable results. Given these production requirements, there is an urgent need to develop a model that enables machine reading comprehension and caters to automatic question-answering scenarios within the scope of agricultural production.

Currently, the research in the field of Large Language Models and Machine Reading Comprehension is highly active and has achieved great success. Machine Reading Comprehension (MRC) represents an exciting line of research within the domains of artificial intelligence and natural language processing. It manifests itself in the form of text-based question answering (QA) systems and equips computers with the capacity to sift through vast volumes of text to pinpoint specific answers. This not only affords users efficiency and convenience but also minimizes the cost associated with information access. On the other hand, Large Language Models (LLMs), such as OpenAI's GPT-3 [1] and GPT-4 [9], are designed to understand and generate human-like text. Trained on vast amounts of data, they possess an exceptional ability to understand context, generate coherent responses, and exhibit a remarkable degree of accuracy in tasks such as translation, summarization, and question-answering. Large language models capture the nuances of language by leveraging deep learning architectures and can handle a wide array of tasks without task-specific training data. These capabilities make them a potent tool for a multitude of applications, including the development of question-answering systems in specific domains.

In this paper, we are trying to explore the application of Large Language Models in constructing a QA model for agriculture scenarios. We construct a dataset for the experiments of document QA in agricultural scenarios. We import two model (ask-my-pdf and chat-pdf) as our baseline and use them to do the single and multiple document QA task. Then, we proposed several methods to improve the performance of the model in fragment retrieving and generating concise and accurate answers. In multiple document QA, we propose two possible methods to select the candidate documents, which are manually labelling the classes of documents and cluster the documents based on the embeddings of the texts. Then, we use the question-class pair to train a classifier and use the classifier to select the candidate documents for the question. In prompt tuning, we also propose two possible methods to generate the prompt template for the question, which are ensemble and in-context learning method. For ensemble method, we use several models to classify the question and use the majority vote to decide the prompt template for the question. For in-context learning method, we use the question and the desired answer to train a sequence-to-sequence model that generate the prompt template for the question.

The rest of the paper is organized as follows. Section 2 introduces the model we used in this experiment. Section

* Corresponding author: x6he@ucsd.edu

3 provides a brief overview of the related works. Section 4 describes the dataset we used in this experiment, and section 5 concludes the paper and discusses the future work.

1.2 Related Works

In this section, we will introduce the related works in the field of Large Language Models and Machine Reading Comprehension.

1.2.1 Some Previous Works about Machine Reading Comprehension

The task of Machine Reading Comprehension (MRC) was firstly experimented by Lifu Huang et al [5]. They proposed a new dataset named CoSMOS, which is a large-scale dataset of machine reading comprehension questions in multiple choices format and establish the baseline performance. The paper [1] stated GPT-3, a pre-trained language model with 175 billion parameters, achieved strong performance on many NLP tasks. Such model now can have a solid performance not only on the tasks in form of multiple choice but also on extraction of span. The paper [6] proposed a Semi-Supervised and Contrastive Learning-Based model in agricultural scenarios QA Task and it's also the paper that gave us several insign in our experiment.

1.2.2 Works about How to Import Knowledge from Outside and Retrieve Documents

How to import knowledge from outside is an important task in the field of NLP, especially in the field of Machine Reading Comprehension. The paper [4] introduced a new model named REALM, which allows the model to retrieve and attend over documents from a large corpus such as Wikipedia. In the paper [8], the author proposes a knowledge-enabled language representation model (K-BERT) with knowledge graphs (KGs). The model directly injects the triples of knowledge graph into sentences and uses them as the token. The model first retrieves the entities in the given sentence, and if there is a corresponding entity in KG, the model takes it out and hang it on the entity word of the main trunk like a branch to get a sentence tree. Then flatten it and feed into BERT with a soft position id to remain the information of tree structure. Then, in order to make the model retrieve the documents more effectively when facing large corpus, the paper [10] and [3] proposed some methods namely RALM and HyDE to help the model retrieve the most relevant text fragments from the document even without labels. The idea of HyDe is based on contrastive learning, that is, the model will generate a hypothetical answer for the question to retrieve the most N relevant text fragments from the document. The actual answer will also feedback to the generator of hypothetical answer to generate the answer that closer to the correct answer to help retrieve the text fragments more accurately.

1.2.3 Models Used in the Experiment

The paper "Attention Is All You Need" [11] proposed the Transformer model, which is a novel neural network architecture based solely on attention mechanisms, dispensing with recurrent neural networks (RNNs) and convolutional neural networks (CNNs) entirely and has a strong performance on NLP related field. In our experiment, we import BERT [2] which is a pre-trained transformer model to do some tasks of classification. The model ask-my-pdf(https://github.com/mobarski/ask-my-pdf/tree/main) is built on the top of GPT model and use HyDE method to retrieve the candidate paragraph. When question comes, the model will generate a hypothetical answer for the question and use the hypothetical answer to retrieve the most relevant text fragments from the document. Then, the model will use the actual answer to feedback to the generator of hypothetical answer to generate the answer that closer to the correct answer to help retrieve the text. The model chat-pdf (https://www.chatpdf.com/) is built on the top of GPT-3 model and use the idea of zero-shot learning to generate the answer for the question. The model will directly use the question to retrieve the most relevant paragraph from the document and then generate the answer for the question. The one difference from ask-my-pdf model is that chat-pdf model supports the interaction between user and the model that the answer generate by the model will be influenced by the context of the previous QA between user and model.

1.2.4 Works about Prompt Learning

In "A Survey of Large Language Models" [14], it states that prompt tuning is an effective method to improve the performance of the model. The paper [7] briefly discusses all kinds of prompt tuning and their application scenarios. The paper [13] proposed a new method of prompt tuning named Chain-of-Thought, that increases ability of models in the tasks require strong reasoning. The paper [12] focus on soft prompt, that learns a prompt on one or more source tasks and then uses it to initialize the prompt for a target task. In the paper [15], the author proposes to use the large language model as a prompt generator, that through training with input and output pair, the large language model learn to generate the prompt template for the question.

2 Methodology

In this paper, we use choose two pre-trained models as our baseline. One is ask-my-pdf model, and the other is chat-pdf model (https://www.chatpdf.com/). Both models are built on the top of GPT model and can be accessed freely through internet. The major difference between two model is that ask-my-pdf uses Hypothetical Document Embeddings (HyDE) that helps to create effective fully zero-shot dense retrieval systems when no relevance label is available (shown as Fig. 1). On the other hand, chat-pdf model is built based on GPT-3 model and it's a fully zero-shot model.

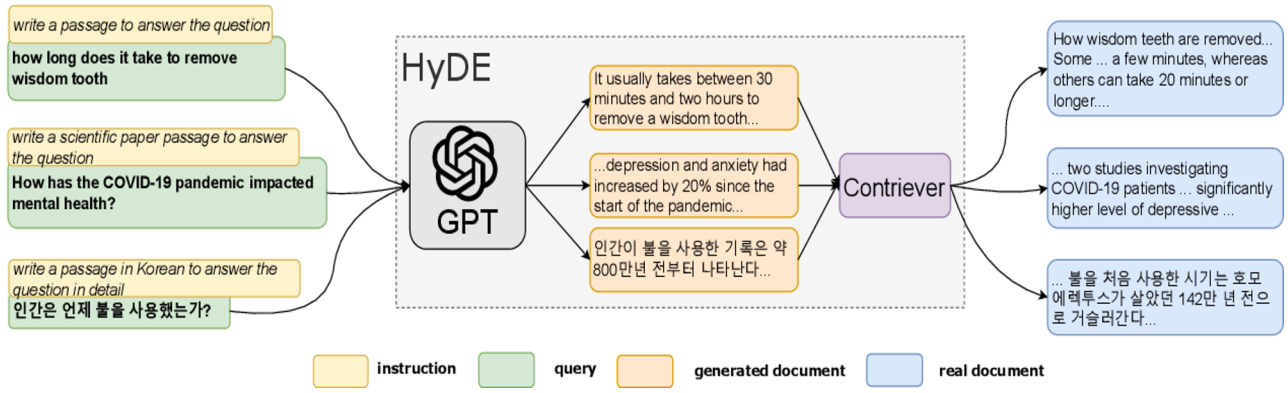

Fig. 1. The idea of HyDE.

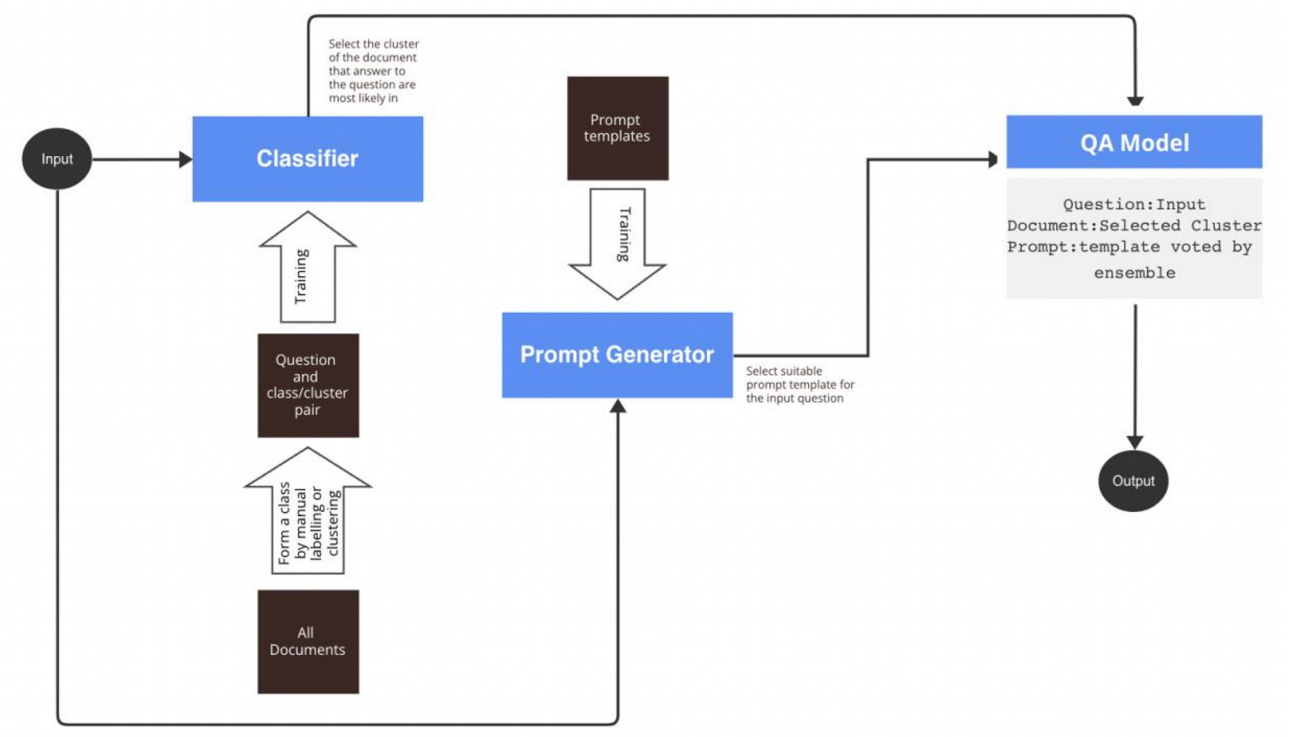

Fig. 2. The flow chart of the final model.

These two models are directly used as our baseline for single and multiple document QA We propose several methods to improve the performance of the model in multiple document QA as Fig. 2 shows the final model we proposed.

Input. The input of the model is the question we want to ask. It can be a single question or a list of questions. In our propsed model, the input will first feed into the classifier module and prompt generator module to select the candidate documents for the question and prompt template for the question. Then, we feed the candidate documents and the prompt template into the QA model together with the question to generate the answer.

Classifier Module. For multiple document QA, we want to first reduce the workload of retrieving the documents. We don't want to feed all the documents into the model and let the model to select the candidate paragraph. Thus, we want to use a classifier to select the candidate documents for the question and only retrieve the answer through these documents. We propose two possible methods to select the candidate documents, which are manually labelling the classes of documents and cluster the documents. For manually labelling the classes of documents, we will manually assign the document to a class and train a classifier to select a class of documents based on the question we feed to the model. The second method is to cluster the documents based on the embeddings of the text instead of manually label them. The classifier is trained base on the question-class/cluster pair of the dataset. And in the final model, we choose the second method, and the reason will be discussed in the following experiment sections.

Prompt Generator Module. This module serves to give the suitable prompt for the language model. Prompt learning is a recent innovation in natural language processing research that has played a crucial role in enhancing machine reading comprehension and document question-answering (QA) capabilities. Fundamentally, prompt learning involves teaching a machine learning model to understand and respond to a task by providing a

text "prompt" that frames the problem in a way the model can understand. In the context of machine reading comprehension and document QA, this approach has improved the model's ability to extract relevant information from a text and respond to queries effectively. We also have two difference prompt generators for this module. The first is an ensemble model, that we prepare several prompt templates and use several models to classify the question and use the majority vote to decide the prompt template for the question. The second is in-context learning model, that we use the question and the desired answer to train a sequence-to-sequence model that generate the prompt template for the question.

QA Model. This part is the large language model that we use to generate the answer, we use the candidate documents selected by the classifier and the prompt template generated by the prompt generator to feed into the model to generate the answer for the question. For QA model, we use ask-my-pdf in our final model. Because chat-pdf model cannot change the prompt, thus we can only perform the experiment on ask-my-pdf model. In following experiment section, we will perform the experiment each method mentioned above separately to both ask-my-pdf and chat-pdf model and compare the result with the baseline and finally combine the classifier and prompt generator together to test the result.

3 Experiment

3.1 Dataset

The dataset we used in this experiment is an agricultural QA dataset, which is constructed by us. The dataset contains several documents that are from different institutions. All the documents contain agriculture related texts. We extract the questions and answers from the documents and construct the dataset. The dataset contains 100 pairs of question and answer. For each question, the answer could be found on a specific paragraph of the document. Following is an example of the dataset.

> The most important for achieving a high infiltration is to maintain a topsoil with a good soil structure containing many cavities and pores (e.g. from earthworms). Cover crops and mulch application are suitable to create such a favourable top soil structure. Further, they help to slow down the flow of water, thus allowing more time for the infiltration.

Fig. 3. A paragraph from a document in the dataset.

Table 1. A question-answer pairs based on the paragraph in Fig. 3.

Question	What's the most important thing for achieving a high infiltration?
Answer	The most important for achieving a high infiltration is to maintain a topsoil with a good soil structure containing many cavities and pores (e.g. from earthworms)

Our dataset was constructed by meticulously selecting agricultural texts from various sources, including academic journals, government reports, and online agricultural forums. The selection criteria focused on topics such as crop management, soil health, and pest control. Each document was subjected to a rigorous cleaning process to remove irrelevant information and ensure the accuracy of the content. The annotation process involved a team of agricultural experts who labeled the questions and answers based on the context provided in the documents. To ensure the dataset's diversity, we included texts from different geographical regions and farming practices. The final dataset comprises 100 question-answer pairs, each meticulously curated to reflect the complexity of agricultural scenarios (Table 1).

The dataset covers a wide range of agricultural scenarios, including but not limited to:

Crop Cultivation: Questions and answers related to planting techniques, crop rotation, and yield optimization.

Pest Management: Strategies for identifying and controlling pests, as well as the use of integrated pest management (IPM) practices.

Soil Health: Information on soil testing, nutrient management, and the impact of soil health on crop growth.

Agricultural Machinery: Usage and maintenance of agricultural equipment, including tractors, harvesters, and irrigation systems.

These scenarios were chosen to represent the diverse challenges and opportunities faced by farmers in their daily operations.

3.2 Evaluation

We employ several evaluation metrics for the performance of models. Firstly, we use Exact Match (EM) and F1 score in token-level. EM score is the metric that measures if predictions that exactly match the ground truth answers. The EM score is binary, either the answer is exactly correct or it's not. Therefore, it's a strict metric because it requires the model to predict the answer precisely as it appears in the ground truth, without any variation. This can make it challenging for models in more complex tasks, where there can be multiple valid answers or where the answer might be expressed in different ways. Token-level F1 is a commonly used MRC task evaluation metrics. The equation of token-level F1 for a single question is as follows:

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

Beside these two metrics, we also create a manual score scale that evaluate each answer based on a 5-point scale shown as follows. Also, we use candidate paragraph

retrieval accuracy and accuracy of answer given the model select the correct candidate paragraph to evaluate the performance of the model (Table 2).

Table 2. The manual score scale.

Score	Description
5	The answer is exactly correct, and the answer is concise
4	The answer is exactly correct, but the answer is not concise
3	The answer is partially correct, but the answer is not concise or contains unrelated information
2	The model correctly finds the candidate paragraph for the correct answer but doesn't provide the correct answer
1	The answer provided is totally wrong/Model mistakenly reject to answer

3.3 Baseline

We use the dataset we constructed to evaluate the performance of the two models. There are two major tasks for each model, which are single document QA and multiple documents QA. For single document QA, we only give the model the document that the answer to this question comes from and ask the model for the answer. For multiple documents QA, we give the model all the documents in the dataset and ask it question. Note that for ask-my-pdf, we use GPT-3.5 model and set the fragment size as 400, fragments before and after both as 2. Then we use the evaluation metrics described above to calculate the performance of two models and generate the baseline.

Table 3. Result of the single document QA.

Evaluation	ask-my-pdf	chat-pdf
EM	0.108	0.031
F1	0.477	0.338
Manual Score	4.111	3.746
Candidate Selection Accuracy	0.922	0.881
Accuracy Given Correct Candidate	0.946	0.966
Overall Answer Accuracy	0.872	0.851

Table 4. Result of the multiple document QA.

Evaluation	ask-my-pdf	chat-pdf
EM	0.031	0.031
F1	0.406	0.266
Manual Score	3.601	3.135
Candidate Selection Accuracy	0.817	0.706
Accuracy Given Correct Candidate	0.961	0.944
Overall Answer Accuracy	0.785	0.667

With the result in Table 3 and 4 of candidate fragments selection accuracy and answer accuracy given by selecting the correct candidate, we calculate the overall answer accuracy by multiplying these scores and get accuracy of 0.872 for ask-my-pdf and 0.851 for chat-pdf in single document QA and 0.785 for ask-my-pdf and 0.667 for chat-pdf in multiple documents QA. We can see that for both tasks, the performance of ask-my-pdf is better than chat-pdf. More than that, we can see that the performance of both models on the multiple documents QA has a significant decline with 14.9% of ask-my-pdf in F1 score and 21.3% of chat-pdf in F1 score.

3.4 Adding Classifier

We Base on the result above, we found the performance of both models on the multiple documents QA has a signficant decline. As we know retrieve the correct candidate text fragment is harder than single document QA, since the corpus is larger and has more disturbance term. Thus, for multiple documents QA task, we think it's possible to use the classifier to select several candidate documents based on the question feed into the model and reduce the workload of retrieving through document. We first classify all the documents into 3 classes: Guideline, News, Encyclopedia. Then, we label the questions according to the document it belongs to. Then we transfer the raw text of document and its title into embeddings and train a BERT classifier based on the title and embeddings of the texts. However, during the training process of the classifier, we found there are too many marginal cases between the Guideline class and Encyclopedia class. Thus, instead of manually labelling all the documents, we decide to use cluster method. Similarly, we extract the raw texts of the documents and the title, transfer them into embeddings and cluster them into 4 clusters by K-Mean algorithm. Then, we assign each cluster a label and use the question-cluster pair to train a classifier. Finally, we will use the classifier to select the candidate cluster of documents for the question and feed them into the model to generate the answer.

Table 5. Accuracy of the classifier trained by different method.

Evaluation	Manually labelled classes	Embedding Clustering
Classifier Accuracy	0.889	0.952

Table 6. Result of the multiple document QA with classifier trained by manual labelling.

Evaluation	ask-my-pdf	chat-pdf
EM	0.031	0.031
F1	0.369	0.346
Manual Score	3.444	3.325
Candidate Selection Accuracy	0.769	0.754
Accuracy Given Correct Candidate	0.969	0.958
Overall Answer Accuracy	0.745	0.722

Table 7. Result of the multiple document QA with classifier trained by embedding clustering.

Evaluation	ask-my-pdf	chat-pdf
EM	0.031	0.038
F1	0.409	0.375
Manual Score	3.770	3.635
Candidate Selection Accuracy	0.865	0.833

Accuracy Given Correct Candidate	0.954	0.952
Overall Answer Accuracy	0.825	0.793

From Tables 5-7, we can see that the classifier trained by embedding clustering has a better performance in selecting the correct groups of documents than the classifier trained by manual labelling.

3.5 Prompt Tuning

Besides the accuracy of selecting the candidate paragraph, we also pay attention to the accuracy and conciseness of the answer. We found that the model is likely to give a long answer even for the questions that can be answered in few words. Thus, we want to find a way to generate a concise and accurate answer for the question. We think prompt tuning is a good choice to achieve this goal.

In our baseline, we use the same prompt for all the questions. However, we found that the model is likely to give a long answer even for the questions that can be answered in few words. Thus, we developped several prompt templates and designed several approaches to match the suitable prompt template for the question based on the paper [7] and [13]. First, we implemented an ensemble model that can vote the most suitable prompt template for the question. For each model in the ensemble, we use BERT to classify the question and assign the question a label, then use the majority vote to decide the prompt template for the question.

Then, we also implemented an in-context learning model that for each question, we construct a specific prompt for it based on the question and the desired answer. Then, we feed the tuple of question, answer and prompt into the model to train it. When testing, we first feed the question into a language model (we use text-davinci) to generate a hypothetical answer for the question. Then, we use the hypothetical answer and the question to generate a prompt template for the question. Finally, we feed the question and the prompt into the model to generate the answer. For multiple document QA, we first use the cluster method experimented above to select the candidate cluster of documents for the question, then use the ensemble model to generate the prompt template for the question and feed the question and the prompt into the model to generate the answer, this is also the final model that we stated above (Tables 8 and 9). Since chat-pdf model cannot change the prompt, we only perform the experiment on ask-my-pdf model.

Table 8. Result of the single document QA of ask-my-pdf model with prompt tuning.

Evaluation	ensemble	in-context
EM	0.231	0.112
F1	0.636	0.524
Manual Score	4.524	4.214
Candidate Selection Accuracy	0.960	0.952
Accuracy Given Correct Candidate	0.967	0.975
Overall Answer Accuracy	0.928	0.928

Table 9. Result of the multiple document QA ask-my-pdf model with final model.

Evaluation	ask-my-pdf
EM	0.176
F1	0.621
Manual Score	4.167
Candidate Selection Accuracy	0.865
Accuracy Given Correct Candidate	0.963
Overall Answer Accuracy	0.833

4 Discussion

We infer that the performance gap between ask-my-pdf and chat-pdf is that ask-my-pdf use the HyDE method to retrieve the candidate paragraph and use GPT-3.5 model to generate the answer, which is more powerful than GPT-3 model. Also, during the experiment, we found that chat-pdf model is more likely to generate a long answer even for the questions that can be answered in few words, and contains some unrelated information, it can also be partially reflected by the f1 score and our manual score. We think it's because the model is trained on the dataset that contains the interaction between user and model, so it's more likely to generate a long answer.

For multiple documents QA, we think it's because retrieve the correct candidate text fragment is harder than single document QA. Thus, it's hard for the model to select the right fragment. Finally, we notice that the accuracy of answer given by selecting correct candidate fragment is high for both models in both cases. Thus, we think the remaining challenges are mostly in how to efficiently select the correct fragment and generate high quality answers.

From the result above and compared to the baseline, we confirm that manually labelling the documents is not a good choice as it cannot avoid the misclassification of marginal cases as we can see the candidate fragment selection accuracy of ask-my-pdf is lower than the baseline. However, for chat-pdf, the candidate fragment selection accuracy has some slight improvement. We think this is because the model is built on GPT-3 which has a weaker ability to retrieve the candidate fragment than GPT-3.5. Thus, if the classifier selects the correct class of documents, the model can retrieve the correct fragment more accurately.

On the other hand, we can see that the performance of both models on the multiple documents QA has a significant improvement compared to the baseline when using cluster instead of manually labelling especially for chat-pdf which uses a less powerful GPT model. The overall answer accuracy for chat-pdf in this experiment is 79.3% which has a 18.9% improvement compared to the baseline. The F1 score and manual score also have improvement of 40.1% and 15.9% respectively. We think it's because the cluster method can find some similarity and hidden features between the documents through embeddings that human may not notice. Thus, it can avoid some marginal cases and achieve a higher accuracy in select the correct cluster of documents. Thus, we can see that the candidate fragment selection accuracy of both models has a significant improvement.

Through the result above, we see the significant improvement for all the metrics. The overall answer accuracy comes to 0.928 which improves 6.5% compared to the baseline. For single document QA, F1 score has 33.3% improvement, manual score has 10% increase, and EM score has a significant increase of 113.9% and EM score of 0.231 means 1/5 of the answer exactly match the correct answer we expected and indicates that by selecting the correct prompt template, the language model can produce a concise and correct answer in specific agricultural scenario even without training or fine-tuning the language model. For multiple document QA, EM score also has a significant improvement, F1 score increases by 53%, and the overall answer accuracy is 0.833 which has a 6.1% increase.

Also, for the in-context learning model, it also achieves a good performance in single document QA, but not as much as the ensemble does. We don't think it turns out that using ensemble to select prompt is a better choice. Rather, we think by the measure of F1 score, answer accuracy, and manual score, the in-context learning model still has a close performance compared to the ensemble. And the performance is limited because we don't have a large enough dataset to train it. We think in practice, in-context learning has a larger potential because we don't have enough time to cover all the prompt for all the agricultural related questions and such in-context learning gives the possibility to use the large language model itself to generate the prompt. Also with a larger dataset, we can fine-tune the generator for the hypothetical answer so that the model can generate a better prompt.

5 Conclusions

In this paper, we introduce the application of Large Language Models and Machine Reading Comprehension in agriculture scenarios. We firstly introduce the several techniques and methods that have been used in the field of Large Language Models and Machine Reading Comprehension to retrieve the relevant text fragments from the document and generate the answer for the question. Then, we proposed several methods to improve the performance of the model in agriculture scenarios. We also construct a dataset for the experiment. Finally, we evaluate the performance of the model and discuss the result.

In the future, we will continue to improve the performance of the model in agriculture scenarios. As the dataset we constructed is relatively small, we will try to collect more data and test the performance of the model on a larger dataset. Also, we will try to discover a new approach to classify the documents and retrieve the document more accurately. Besides, the prompt learning is a new method that has a great potential in the field of NLP. We think it's meaningful to explore how to assign the most suitable prompt template for the question and improve the performance of the model. Based on the idea of in context learning, it's worth to train a sequence-to-sequence model that that generate prompt templates based on the pair of question and desired answer.

Acknowledgments

Authors wishing to acknowledge assistance or encouragement from colleagues, special work by technical staff or financial support from organizations should do so in an unnumbered Acknowledgments section immediately following the last numbered section of the paper.

References

1. Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
2. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
3. Gao L, Ma X, Lin J, et al. Precise zero-shot dense retrieval without relevance labels[J]. arXiv preprint arXiv:2212.10496, 2022.
4. Guu K, Lee K, Tung Z, et al. Retrieval augmented language model pre-training[C]//International conference on machine learning. PMLR, 2020: 3929-3938.
5. Huang L, Bras R L, Bhagavatula C, et al. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning[J]. arXiv preprint arXiv:1909.00277, 2019.
6. Huang Y, Liu J, Lv C. Chains-BERT: A High-Performance Semi-Supervised and Contrastive Learning-Based Automatic Question-and-Answering Model for Agricultural Scenarios[J]. Applied Sciences, 2023, 13(5): 2924.
7. Liu P, Yuan W, Fu J, et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing[J]. ACM Computing Surveys, 2023, 55(9): 1-35.
8. Liu W, Zhou P, Zhao Z, et al. K-bert: Enabling language representation with knowledge graph[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(03): 2901-2908.
9. Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report[J]. arXiv preprint arXiv: 2303.08774, 2023.
10. Ram O, Levine Y, Dalmedigos I, et al. In-context retrieval-augmented language models[J]. Transactions of the Association for Computational Linguistics, 2023, 11: 1316-1331.
11. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
12. Vu T, Lester B, Constant N, et al. Spot: Better frozen model adaptation through soft prompt transfer[J]. arXiv preprint arXiv:2110.07904, 2021.
13. Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in neural information processing systems, 2022, 35: 24824-24837.

14. Zhao W X, Zhou K, Li J, et al. A survey of large language models[J]. arXiv preprint arXiv:2303.18223, 2023.
15. Zhou Y, Muresanu A I, Han Z, et al. Large language models are human-level prompt engineers[J]. arXiv preprint arXiv:2211.01910, 2022.