Utrecht University

MSc Thesis

Master of Artificial Intelligence

Student number: 7069146

Will transformer give you answers? An effective way to conduct multilingual real-world ConvQA tasks with transformer

---

Author:
Shaomu Tan[1,2]

Supervisor:
Dr. Denis Paperno[1]

External Supervisor:
Wei Zhong[2]

Second Reader:
Dr. Marijn Schraagen[1]

[1]Utrecht University, Utrecht, The Netherlands
[2]ABN AMRO N.V., Amsterdam, The Netherlands

July 14, 2022

Contents

Abstract

Background The goal of question answering (QA) systems is to automatically provide answers to given questions. Modern QA systems commonly extract the possible answer intervals from documents that contain explicit answers. However, in the real-world case, machines may only be able to answer questions using relevant background knowledge. The background knowledge documents contain implicit answers to questions, thus it requires the machine to understand latent semantic information and consider the questions simultaneously in order to answer questions. Moreover, as the conversation goes more extensive for multi-turn QA tasks, it becomes more challenging to provide answers by considering both the current question and historical contextual information.

Methods In this thesis, we propose a neural retrieval-reading system with customized modules to investigate the possibility of using background knowledge to answer questions and explore a few directions to leverage historical contextual information in real-world Conversational QA (ConvQA) scenarios. In order to be more relevant to the industrial scenario, we conducted all experiments using a real-world multilingual customer service dataset provided by a Dutch corporation. We first implemented the sequence-to-sequence[] (seq2seq) model to generate answers by reading documents containing potential background knowledge retrieved by a dense retriever. In addition, we proposed and validated the use of a text summarizer to refine the content of historical context to improve the quality of machine retrieval and reading. Furthermore, we explored whether it is possible for neural readers to improve reading quality by introducing an additional attention module to force the machine towards focusing more on valuable historical information.

Results Overall, our system can efficiently use background knowledge and historical contexts in real-world multilingual ConvQA scenarios. The experimental results show that the machine can create significantly better answers when background knowledge is taken into account. This implies that using a retrieval-reading model to efficiently exploit background knowledge significantly outperforms using only a generative reader or a retriever without considering background knowledge. Specifically, the qualitative experimental results show significant improvements in both correctness and readability of the generated answers when compared with the retriever; the quantitative experimental results have the same conclusion, i.e., there is an improvement in f1 rouge score over 2%. Furthermore, text summarization improves performance by refining the history context information for both the retrieval and reading tasks. It improves both retrieval and reading performance by up to over and around 2% f1 rouge-1 score respectively. However, our experiments demonstrate that adding an additional attention module to the encoder of the seq2seq model for the historical context makes the model harder to train and slower to converge.

1 Introduction

1.1 Background

Contextualized embedding-based models have recently shown powerful capabilities by improving the state-of-the-art on various natural language processing (NLP) tasks, such as question answering (QA), sentiment analysis, and text classification. As a result of the rise of Pre-trained Language Models (PLMs) such as Bert[15] and RoBERTa[44] in NLP, more research in the field of QA has started to focus on understanding the latent semantic information in documents. For example, Machine Reading Comprehension (MRC) aims to answer questions by reading a given document, and ad-hoc neural retrieval compares document similarities on the document level and then retrieves the most similar ones. These studies have explored and achieved near-human results in the Open-domain QA and MRC field, i.e., SQuAD[57], QuAC[11], CoQA[58], and MS-MARCO[49].

Conversational AI products, such as Siri, Cortana, and IBM Watson, are increasingly used across industries to assist humans with complex tasks such as customer service and recommendations[18, 61, 34, 78]. The question answering technology behind them is a research domain aiming to answer questions in the natural language form automatically. Question answering research generally includes studies in the open domain and closed domain. The former studies how to make machines answer questions in various domains, such as "Who is the first human to go to space?"; "when the first electronic computer was built?". The closed domain QA tasks, on the other hand, investigate QA in a relatively closed domain. For example, in the banking domain, machines are expected to answer questions like "how to block a credit card?". Generally, QA systems are usually designed to answer factoid questions, such as questions starting with what, when, who, and where, especially for open domain QA tasks. Non-factoid questions are more likely to ask about descriptions and instructions, such as "why the earth is magnetic?", "how do I withdraw a cross-border transfer I made yesterday?", etc.

1.2 Research approach

Modern QA systems often rely on information retrieval to retrieve documents containing explicit answers and extract them. However, in real-world cases, the answer to a question may come from multiple documents instead of one, or even does not explicitly exist in the retrieved documents. In particular, for non-factoid questions, it is difficult or sometimes even unrealistic for the machine to retrieve documents that clearly contain the complete actual answers from historical data, which places a higher demand on the machine. Frequently asked questions (FAQ) or How-To documents help users with simple guidance on the basics in real-world applications, however, when faced with complex and very personal non-factoid questions especially in customer service tasks, there is no guarantee that these documents will cover a wide range of issues.

In addition, previous research has pointed out that IR systems using millions of historical QA pairs cover well open domain in which new users can ask questions[12]. However, in actual industrial use, we cannot guarantee the effectiveness of these data in the practical application of QA in certain specific areas. And because their sources may be uncertain, it is also difficult to access whether these data have been identified by certain measurements, thus the blind use of these data may lead to some unpredictable consequences, such as sexism and racial discrimination.

Therefore, due to the limitations of the data in practical applications, simply extracting answers from a given golden passage may not be feasible. The first goal (RQ1) of this thesis is to explore whether the machine is capable of answering questions using relevant background knowledge. To investigate this, we first build a state-of-the-art system to retrieve potential background knowledge, then we explore the possibility of using neural sequence-to-sequence generative models to leverage

the potential knowledge and then generate answers.

Humans seek answers in many conversational QA (ConvQA) scenarios by asking several questions for more comprehensive information. As the conversation goes more extensive, multiple rounds of conversation yield more historical contexts. Specifically, this historical contextual information refers to conversations that took place between the questioner and the respondent prior to the current question, also known as historical QA pairs. For example, in the customer service QA scenario, customers are more likely to ask follow-up questions about previous answers. Thus, this history dependence issue requires the machine to consider and encode the historical information additionally, which poses additional challenges for existing QA systems. In addition, QA data in the real world often tends to be colloquial and informal, covering non-valuable semantic content, such as greetings and personal information. When such irrelevant information appears in historical contexts, it is difficult for the model to capture the focus of attention from the complex information.

Therefore, the other goal (RQ2) of this thesis is to investigate the impact of historical context information on machine retrieval and reading quality. To avoid ambiguity, in contrast to the reading task performed by humans, machine retrieval and reading refers to the task of retrieving documents and exploiting the statistical distribution of words to predict the answer to a given question. Generally, the composition category of historical contextual information determines its content and thus influences what information the QA system encodes. We consider the composition category of historical contexts as historical questions, historical answers, and both of them. Where the former consists of all questions asked by the user prior to the current query, and the latter is the system's response to those historical questions. We first focus on what types of historical contextual information can help machines answer questions better (RQ2.1); hence we conducted machine retrieval experiments using three different historical information compositions. Second, considering real-world conversations may contain irrelevant fragments in the conversation such as greetings and confirmations, we propose a text summarizer to refine the historical information content in both machine retrieval and reading task (RQ2.2).

Furthermore, not all historical contexts are helpful for understanding the current question, and some may even confuse the machine to trivialize the current question, especially when historical information occupies the majority of the input. Our final goal ((RQ2.3)) is to investigate whether it is possible to improve machine reading performance by forcing machines to pay less attention to low-value historical information. In order to do that, we propose an additional history reweighting module that can be extended in the generative model. The main idea is to learn how vital different historical contexts are, then reweight their embedding according to the importance weights we learned.

In order to be more relevant to the industrial scenario, our research are conducted using a multilingual ConvQA dataset that based on real-world customer service conversations provided by a Dutch banking corporation. Even customer utterances can be linguistically meaningful as answers to previous questions, and staff responses can sometimes be seen as follow-up questions to previous responses. To avoid ambiguity, we define customer utterances as questions and customer service staff utterances as answers in this study. Our data consists of real conversations between the bank's customer service staff and customers, and it includes more than 131,000 conversations and over 339,000 questions and corresponding answers. In addition, our dataset is featured as multilingual; it consists mainly of Dutch and English and a small number of other languages such as German. Compared to other publicly available academic datasets in the field of ConvQA, such as QUAC[11] and OR-QUAC[55], the vast majority of questions in our dataset are non-factoid questions, and the questions and answers in our data are significantly longer than those in public datasets.

The research questions of this thesis are designed as follows:

RQ1: Is the machine capable of answering questions using candidate background knowledge?

RQ2: How do machines leverage historical contextual information for real-world ConvQA tasks?

RQ2.1: How do different historical information compositions affect a machine's retrieval performance?

RQ2.2: For real-world ConvQA tasks, refining the historical contexts help the machine perform retrieval and reading tasks?

RQ2.3: Is it possible to improve machine reading performance by forcing machines to pay less attention to low-value historical information?

1.3 Research aspects

- Our hypotheses and models will be validated on real-world industrial data. This dataset is multilingual and more challenging than other datasets, e.g., our data has longer questions and responses, and the dataset contains multilingual data with a large amount of non-factoid questions.
- We propose a neural retrieval-reading system with customized modules to investigate the possibility of using potential background knowledge to answer questions in the real-world ConvQA scenario.
- Inspired by previous approaches[11, 55, 53], we further explore a few directions to leverage historical contextual information with our industrial ConvQA dataset.

2 Related work

In this section, we first introduce the background of Question Answering and ConvQA in section.2.1 and section.2.2. After that, according to the two main research directions of textual QA, we then focus on the current research progress of IR-based QA and MRC in section.2.3 and section.2.4, respectively. In addition, some of the research directions in dialogue systems are very close to the current research in QA; therefore, we will summarize the research progress and results of these studies in Section.2.5.

2.1 Question Answering Overview

Question answering (QA) aims to answer questions concisely and automatically, and it generally involves various NLP techniques such as natural language understanding, information retrieval, machine learning, and knowledge graphs. Depending on the source of the answer, the whole QA study can be divided into two main paradigms[32, 64, 84]: knowledge-based QA and textual QA. Knowledge-based QA approaches provide answers by reforming questions into logical queries and then searching the corresponding answers in the structured database. In contrast, textual QA approaches obtain answers from unstructured texts, such as a large number of documents, Wikipedia, or web texts queried from search systems.

Textual QA systems are often considered more accessible to implement[84] because it does not require structured information such as a database. There have been many studies at this stage with outstanding results in the direction of textual QA. Some even approached human performance, e.g., close to 30 models outperformed human performance on SQuAD[57]'s leaderboard. In addition, in terms of constructing QA datasets, most of the datasets such as SQuAD[57], NarrativeQA[35], TriviaQA[30] used in previous studies are based on information-verifying qualities, i.e., the questioner asks questions to the respondent based on knowing the answers to the questions, and the respondent's answers are used to assess their ability[14]. This is usually happened when the questioner is given a document and then they are asked to ask a question about the document. However, several textual QA studies such as QuAC[11], and MS Marco[49] have also introduced the information-seeking feature, i.e., the questioners have no prior knowledge of the answers to the questions they are asking[11, 14]. This means the questioners do not know the actual answers to the questions they asked. This knowledge-intensive attribute makes QA research more oriented to real-world scenarios, where answers and background knowledge are obtained by asking questions[21, 38].

Recently, several textual QA studies are focusing on two sub-directions: Information-Retrieval based QA (IR-QA) and Machine Reading Comprehension (MRC). The former focuses on how to use information retrieval to return documents that may contain answers and then extract possible answers from the retrieved documents. MRC, on the other hand, only investigates how to perform the machine reading task more efficiently, which can be seen as a subtask of the IR-QA direction. This means that for the MRC system, the machine is provided with a golden document explicitly containing the answer, and the machine only needs to answer the question based on the provided document. As our research will cover both information retrieval and machine reading comprehension directions, we will summarize and present research findings of IR-QA and MRC in sections 2.3 and 2.4.

In addition, depending on the type, the questions to be answered by the QA system can be mainly classified as factoid and non-factoid questions[64, 6, 32, 78, 84]. Factoid questions are questions that can be simply represented in short text and usually have fewer answer forms. For example, the following factoid questions have very few answer forms. Further, even though the answers to factual questions may take different forms, they all contain the actual answers. For instance, for question 2, although its answers have different expressions, all of them include the true answer,

i.e., "ABNANL2A". Therefore, for factoid questions, it is feasible and common to use extractive QA approaches to extract answers or predict answer intervals from documents.

Factoid Question

1. "Who is the first human to go to space?"
   Answer: "Yuri Gagarin".

2. "What is the BIC code of the ABN AMRO Bank?"
   Answer 1: "ABNANL2A."
   Answer 2: "It is ABNANL2A."
   Answer 3: "The BIC code of the ABN AMRO is ABNANL2A."

In contrast, non-factual questions typically cover many questions about request instructions and descriptions. The answers to non-factual questions can often be syntactically formulated in multiple forms even though their sentence-level semantics are guaranteed to be similar. For the customer service use case, non-factoid questions are more likely to ask the responder to provide a specific detailed instruction. Admittedly, the answers to some non-factoid questions may be semantically different, depending on how the respondent wants to help the questioner. For example, the following example shows that freezing the credit card can be handled by phone app or offline appointments, and it depends on the policies of different banks. Furthermore, the answer to the question can be a follow-up question to get more details, provide more accurate information, or provide guidelines to the user. As a result, non-factoid QA tasks are more oriented to real-world applications, especially in customer service applications. However, most QA studies at this stage rarely focus on these tasks.

Non-Factoid Questions

1. "How can I freeze my ABN AMRO credit card?"
   Answer1: "Would you like to handle it offline? In this case you can make an appointment in your city, which city do you currently live in?"
   Answer2: "Did you lose your card? In that case we recommend you to deactivate your card first, then apply for a new card. You can login to our app and select 'block the card', then request for a new card."

2.1.1 Multilingual Question answering

At this stage, several studies have started to introduce multilingual and cross-lingual QA dataset for research[4, 69, 72, 28, 45, 13], and most of them contribute to the better development of multilingual QA systems by introducing parallel multilingual and cross-lingual QA data. Among them, [4] proposed the XQuAD dataset by translating SQuAD data into ten languages, and they measured the generalization ability of multilingual models by transferring multilingual models trained on other languages to the English QA task. MKQA[45], on the other hand, proposed a larger QA dataset containing more than 260k QA pairs on 26 languages pairs. One limitation of these studies is that most of them cover only factoid questions, and these data are not conducted in multiple rounds of conversation. In addition, only very few datasets have a large amount of data[72, 45, 13]. Furthermore, most of these multilingual QA datasets contain less than 100k of actual QA data when duplicates of QA data in other languages are removed[28, 4, 45, 13], which means that these studies are not very relevant for practical applications in industry.

2.2 Conversational Question Answering

Based on the QA setting, conversational question answering (ConvQA) introduces contextual information by introducing dialogues, which can usually be considered QA questions with multiple turns. On the other hand, like single-turn question answering tasks, the ConvQA task also requires

the system to answer the user's question in a simple natural language form. The introduction of the multi-turn dialogues introduces context information to the system, which means that when understanding the current user question, systems have to consider previous questions and corresponding answers as context information as well. Such historical context dependency imposes additional difficulties on the QA system, requiring the system to rely on other historical QA information to understand the current question.[84, 78, 20]

Historical context dependency example

1. Q1: "Hello, I lost my bank card. What should I do?"
   A1: "Sorry to hear that. Do you have access to our app?"
2. Q2: "Yes."
   A2: "Then you can use the card overview menu in the app, to block your current damaged card, and apply for a new one."
3. Q3: "How long will this take?"
   A3: "The new card should arrive withing 5 working days."

To be more specific, the user's current question sometimes can be the follow-up question to the historical answer[78, 14, 84]; thus, answering them becomes problematic if historical contexts are ignored. In the historical context dependency example, without considering the first QA pair(Q1, A1), it is impossible to give a reliable answer only when looking at the second question Q2. The same issue happened in Q3, where the machine needed to understand what "this" meant in order to truly understand how long it would take for the user to receive the new card. Thus, when the current question involves information about history QA pairs, it is difficult to understand without considering the history.[54, 84, 51]

In addition, the machine is forced to encode longer texts when the historical context is also taken into account. Nevertheless, most of large pre-trained language models have the limitation of processing only a restricted amount of tokens, e.g., 512 token lengths for Bert[15]. In the field of conversational QA, this drawback becomes severe, as we need to consider the context, the current question, and candidate documents together, which will likely cause the length of the text to be processed to exceed the limitation[80]. Considering only a few neighboring historical contexts may alleviate this problem[83, 51, 11, 54], since the questioner's current question may involve only proximate historical contexts. Previous studies[20, 78, 75] have introduced and discussed the difficulties of topic return and topic shift in ConvQA tasks. In a multi-round conversation, the user may ask about a previously discussed topic, which will lead to a shift in the topic of conversation; or the user may continue to seek answers from other previous rounds of conversation (topic return). Therefore, when facing problems such as topic return and topic shift, only encoding limited preceding historical contexts still makes the machine remain in a dilemma: 1. the limitation of long text due to considering all contexts[80]; 2. the problem of sacrificing some performance by considering only the neighboring contexts[51].

Another issue that cannot be ignored is: in real-world ConvQA scenarios, conversations may contain irrelevant fragments such as greetings and information irrelevant to understanding the current question, or sentences with errors and misspellings[17]. Such issues are complex for the machine to tackle because some irrelevant historical contexts are essential for some questions but also can be redundant for others. Further, questioners sometimes describe their questions in too much detail, leading to lengthy questions. Therefore, applying appropriate summarization to historical contexts may be a potential direction to mitigate these issues. It reduces the length of the text to some extent while preserving most of the necessary semantic information[2]. On the other hand, summarization for historical context is capable to removes unnecessary information like greetings.

However, no existing research has attempted to use text summarization to alleviate the long text limitation and refine the historical context in the ConvQA domain. Some studies[68, 3, 12] about

the question rewriting are similar to our idea to some extent. Generally, the question rewriting in the field of ConvQA investigates how to rewrite the combination of historical contexts and the current question into a form that machines can better understand. For example, when a demonstrative pronoun is involved in the current question, machine rewriting approaches usually look for the noun referred to by the referent from the historical context in order to rewrite the current question into a more comprehensible version. However, these approaches[68, 3, 12] usually require the rewritten questions as training data for supervised learning. Their implementation also requires large-scale neural network training, which is hard to acquire in real-world applications.

2.3 Current research progress in IR-based QA

Most modern Information Retrieval based QA (IR-QA) approaches are based on the retrieval-reading architecture. As shown in Figure 1, the retrieval-reading architecture can be divided into two essential components, retriever and reader, according to their functions. The retriever is an information retrieval module that retrieves and returns similar documents. At the same time, the reader is a predictor that reads those returned documents and gives answers to questions using specific methods. We will introduce the current research progress of retriever and reader in the following paragraphs.

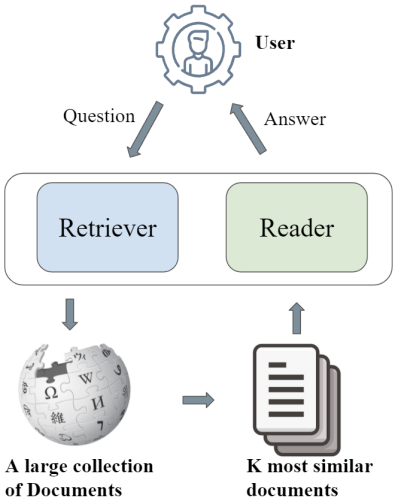

Figure 1: Retrieval-reading architecture

2.3.1 Sparse retriever

Sparse retriever refers to those retrieval models that use sparse representation, such as TF-IDF and BM25. One of the earliest systems with modern architecture is DrQA[9], which was used for the retrieval-reading task and proved to achieve state-of-the-art performance at that time. DrQA proposes a retrieval module based on bigram hashing and TF-IDF to retrieve relevant documents from five million Wikipedia documents, with a multilayer RNN[10] module to predict the answer span. With the advent of contextualized embedding, researchers started to replace the reader module with the more advantageous transformer encoder model. BERTserini[74] proposed an end-to-end model based on BM25 and Bert using Wikipedia as their corpus. They also compared the differences in retrieval at the various granularity of document length, i.e., comparing the performance of prediction at the document level, paragraph level, and sentence level, which brings more possibilities for researchers on information retrieval in the QA domain.

2.3.2 Dense retriever

In addition to sparse retrievers like TD-IDF and BM25, another innovation in IR-based QA tasks is the use of neural ad-hoc retrieval models, sometimes also called the dense retriever. An obvious advantage of a dense retriever is the use of dense embedding to represent the semantics of a sentence or a document, allowing us to find content containing similar semantic information when retrieving documents[84]. In addition, dense retriever can usually be trained as a module with reader together, i.e., forming an end-to-end system, which will effectively help the whole system be trained uniformly and thus improve the whole system's performance. Besides, it is possible to encode and save all the documents before the daily use of inference process so that the dense retriever can perform fast offline inference similar to sparse retriever.

A typical dense retriever is the Dense passage retriever (DPR[33]). DPR is a dual-encoder framework that encodes the question and the document independently. It retrieves similar documents to a question through computing the inner product between the question and all documents. As a result, it outperforms BM25 on several mainstream open-domain QA datasets, and shows superior capabilities when comparing semantic relations and dealing with syntactic relations. Meanwhile, several studies (RAG[38], ORConvQA[55], FID[27]) have adopted DPR's retrieval module and combined it with the faiss[29] system to return similar documents. Their results have shown the effectiveness and efficiency of this combination.

2.3.3 Extractive reader

Another integral module of the retrieval-reading architecture is the reader, responsible for reading the returned documents and giving answer predictions. At this stage, most reader modules are based on extractive models[9, 74, 21, 55]; that is, they read the documents and then predict the probability of each token to be the start and end of the answer span. Undeniably, the reading module has almost the same task as MRC, allowing many IR-based QA models to explore models that perform well in MRC. For instance, DrQA[9] and BiDAF[63] investigated RNN-based models akin to the attentive reader[22] to predict answers, where DrQA predicts answers for each document split into paragraphs and aggregates them. In contrast, ORConvQA[55] and BERTserini[74] implemented Bert-based readers to encode both the question and the candidate document together and then predict the start and end probability for each token in the document.

2.3.4 Generative reader

On the other side, unlike extractive QA, some recent papers are also starting to experiment with generative models as the reading module to explore IR-based QA problems. Especially when extractive QA does not yield good results on non-factoid questions, i.e., we can hardly extract complete answers from the retrieved documents. Generally, these models follow the sequence-to-sequence[65] architecture, which means using an encoder to read the current question and the candidate document, and then using a decoder to generate free-form answers. Specifically, RAG[38] uses DPR as the retriever and BART[37] for answer generation at the sentence and token levels. At the same time, FID[27] follows BM25 and DPR to return documents similar to the question, then uses the encoder module of the T5[59] model to encode all the documents before integrating them into the decoder to predict the answers. Although the use of generative models in IR-based QA problems is still a relatively new research direction, these limited results have demonstrated the benefits and feasibility of generative models, which are certainly worthy of further exploration.

In addition, for the multilingual QA task, MKQA[45] performed experiments with retrieval-extractive QA and retrieval-generative QA models on their dataset for the open domain QA task. Specifically, they use a monolingual dense retriever to retrieve relevant documents in English, then use a machine translation model to translate them into other languages, and finally use the extractive QA models to predict the answer spans. For the generative QA models, they only input the retrieved English documents as external knowledge into the generative models. Therefore one of

their research limitations is that they do not use the same translation procedure for the generative models, even though the generative model is multilingual, yet it might be expected to perform worse on cross-lingual tasks.

2.3.5 IR-based QA in the conversational setting

Until now, few studies[55] have focused only on combining information retrieval-based QA in the conversational setting. Most studies either focus only on enhancing the IR-based QA architecture in the single-turn QA scenario[38, 21, 33, 47, 27, 48, 9], or they are conducted only in the context of conversational machine reading comprehension[51, 8, 58, 11, 76, 24, 54, 53, 80, 31, 83]. Additionally, no existing study has attempted to investigate using generative models to provide answers in the multilingual conversational IR-QA scenario at this stage. A similar study is QReCC[3], where they use the generative model to rewrite sentences for the information retrieval, however they use the extractive model to predict answer intervals.

ORConvQA[55] proposes an end-to-end system architecture to solve the conversational information retrieval-based QA problem. Specifically, the model uses a dual-encoder-based retriever architecture to retrieve k most relevant documents to the current user question and then uses the re-ranking and reading modules to predict the answer span in each candidate document to answer the question. The selection of the final question depends on three different scores: retrieval score, re-ranking score, and reading score, and this design allows the end-to-end model to be trained simultaneously. The retrieval score represents how confidently the retrieved document is relevant to the current user question, while the re-ranking score indicates how likely a document's ranking needs to be changed, and finally, the reading score is the confidence level that the token in each document is predicted to be the answer span.

Even though ORConvQA[55] tried to connect these two domains and achieved good results on mainstream datasets, they do not consider some problems specific to conversational situations, such as dynamic history embedding, to solve historical context dependency. Even though they adopted the sliding window approach to divide long passages for reading, their research did not indicate the advantages or disadvantages of this approach to the final performance.

2.4 Current research progress in Machine Reading Comprehension

With the rise of neural models, more and more studies focus only on the reading task in the QA domain, which is the driving force behind the development of Machine Reading Comprehension (MRC) research. MRC focuses on tasks given a document, a question, and the goal is to provide answers by reading the input. In other words, MRC systems generally provide answers by predicting the answer span from a given passage; thus, a vital assumption of MRC is that the true answer can be found in the given passage. Even though the MRC task, to some extent, simplifies the QA task by providing a golden passage with answers, many competitions or datasets have emerged in the NLP community in recent years to address the MRC problem, such as SQuAD[57], TriviaQA[30], QuAC[11], NewsQA[67]. In contrast to IR-based QA tasks, these MRC tasks focus only on how machines perform the reading task; thus, it does not perform the retrieval task.

Numerous studies started to address the MRC problem using neural approaches as the neural network and attention mechanism[70] emerged. Some studies use traditional machine learning algorithms to answer MRC questions by extracting named entities or hand-crafted features from the context. An example is the logistic regression model proposed in SQuAD[57], which considers nine features to predict answers, including dependency tree paths, and span POS tags. On the other hand, more and more end-to-end deep learning models based on techniques such as CNN, RNN, and transformer are used to provide more effective and robust systems. For example, the Bi-Directional Attention Flow network (BiDAF[63]) utilizes multi-stage hierarchical architecture

and CNN to represent context at the character and word level. It then uses an attention mechanism to generate query-aware context representation and finally predicts the answer span using multilayer LSTM[23]. A new level of performance has been witnessed for transformer-based models like Bert[15] and Albert[36] on the machine comprehension task, and approaches based on them behaves close to the real human label in datasets such as SQuAD.

For multilingual MRC tasks, currently several studies generally exploit multilingual PLMs in their research[28, 77]. For example, BiPaR[28] compares the performance of multilingual model and two monolingual models on MRC tasks in their experiments, while [77] uses the construction of cross-lingual data pairs through back-translation to enhance the performance of multilingual models. Even though these studies have investigated the use of additional translation modules to enhance their monolingual or multilingual systems, their methods are not efficient and easy to implement.

2.4.1 Conversational Machine Comprehension

Another innovative area that deserves attention is the inclusion of more contextual information in the MRC setting. Conversational Machine Comprehension (CMC) aims to add context to the MRC task by introducing multiple rounds of dialogues, thus making the MRC task more adaptable to practical applications of Conversational AI. As evidence, CoQA[58] and QuAC[11] datasets introduced multi-round QA, information-seeking feature, and unanswerable questions on top of the MRC task to provide a more realistic task setting to the NLP community. Formally, the CMC task can be defined as giving a current question q_k, a passage P containing the answer, and all historical QA pairs H_k preceding the current question, and the system needs to give the corresponding answer.

Previous work[20] summarizes the framework of current neural CMC systems, indicating that it consists of four modules: 1. History selection module; 2. Encoder; 3. History modeling module; 4. Answer prediction module. Specifically, the history selection module will be responsible for selecting the parts of all historical QA pairs that are useful for understanding the current question in order to avoid the possible consequences of topic shift and topic return. After that, the encoder will be responsible for encoding the current question, selected history and the document. The history modeling module integrates embeddings of the passage, current query, and the history. Finally, the prediction module will predict the answer span or generate free-form answers based on the received embeddings.

We summarize and compare the methodologies of several papers in the field of CMC, and we will focus on how they encode and use contextual information. First of all, most models are based on the following two methods or models: 1. large-scale pre-trained language model, e.g., Bert[15], Roberta[44]. 2. Attention-based sequence models such as BiDAF[63], DrQA[9]. We observed that methods based on pre-trained LMs had achieved higher results than the others (according to QuAC and CoQA leaderboards), representing the advantage of such large-scale pre-trained models in the QA domain. Several approaches[11, 24, 25, 53] implemented self or cross attention layers for the history modeling and context integration to integrate passage, question and history together. We also observed that studies using transformer-based models typically generate the contextualized embedding by encoding passages, questions, and histories simultaneously; thus, these models barely make use of the additional contextual integration layer.

2.4.2 History selection and modeling in CMC

Compared to MRC, CMC needs to consider historical information while understanding the current issue because of the history dependency, especially when the topic shift and topic return issues appear in the conversation[53, 54]. Therefore, how the context is selected and encoded is critical to the CMC task. At this stage, most approaches[31, 83, 58, 51, 11] simply prepend all historical contexts to the current question, or prepend only k previous historical QA pairs due to the length limitation. Their results have shown disadvantages because some historical QA pairs do not help

the system to understand the current question, but bring more noise instead.

However, some approaches started to explore the possibility of dynamically selecting and encoding histories. For example, previous work[54] uses History Attentive Embedding (HAE) to encode the current question and passages while considering whether each token in them is also present in the historical context; then, it adds this information to the encoding calculation of the Bert. While History Attention Mechanism (HAM)[53] incorporates the location information of historical QA pairs based on HAE, i.e., when the token in the passage appears in the i th historical QA pair simultaneously, the historical information of i is also encoded into Bert. In addition, HAM also computes the sequence-level embedding for each history pair, then computes the weights of each history pair by a single-layer attention network, and finally uses those weights in the token-level embedding. Finally, in HAM, the final prediction of the answer span is performed by aggregation operation.

A significant limitation of HAM and HAE is that these history encoding strategies can only be implemented on extractive QA models. Because they only reweight tokens in the candidate passages, thus only passage embeddings will be updated. However, for generative QA models, they predict the answer by considering not only candidate passages but also query and history contexts. Such limitation is reasonable for CMC tasks since the golden passage is provided to the model. Therefore, exploring additional historical modeling strategies for IR-QA tasks and generative QA tasks is a feasible research direction.

2.5 Dialogue system

Dialogue systems are systems designed to communicate and interact with human users. According to the function and intention of these systems, they are commonly divided into task-oriented dialogue systems and open-domain dialogue systems (also called chatbots)[32]. Task-oriented dialogue systems are designed to fulfill specific user commands and requests, such as ordering food, checking the weather, etc.; thus, they are often implemented in restricted domains. Chatbots, on the other hand, are more focused on how to communicate with humans in a casual way. Generally, chatbots are not expected to answer users' questions accurately but rather to chat with users pleasantly or humorously. Consequently, many chatbots are often demanded to give a more human-like response[1, 81, 82]. It is undeniable that conversations in dialogue systems, especially for task-oriented systems, are more akin to question answering conversations. The reason behind this is because those conversations are conducted in a question-and-answer format, and they end with the user receiving a satisfactory answer. These properties of dialog systems have made their task analogous to conversational question answering; therefore, in this subsection, we will outline and review some of the research progress and findings for dialog systems.

Similar to the research in QA, several dialogue systems are developed using information retrieval models or generative models to respond to users' utterances. Information retrieval-based dialogue systems[7, 66, 46, 26, 73] adopt retrieval models to provide responses by searching for the most similar question in a corpus. We observed a similar conclusion as in the IR-QA studies, that is, the dual-encoder-based retrievers can produce better results than the traditional word frequency-based models (e.g.,: TF-IDF, BM25). On the other hand, there are some studies[81, 79, 1, 82] utilizing generative models to answer users' utterances in dialogue system studies. Specifically, given user utterances, these studies use sequence-to-sequence models[60, 81, 1, 82] or decoder-only language models[79] to predict the responses.

Furthermore, some previous studies[16, 62, 73, 60] have investigated the use of systems similar to retrieval-reading architectures to enhance the system by introducing retrievers based on generative models to consider more external knowledge such as Wikipedia articles or knowledge bases. For example, [16] proposes Generative Transformer Memory Network to retrieve knowledge candidates from Wikipedia and then incorporate those candidates with a sequence-to-sequence model

to obtain an end-to-end system that can select beneficial knowledge candidates using the negative log-likelihood loss function. In contrast, [60] proposes a retrieve-and-refine system that is very close to the retrieval-reading architecture, where they use a poly-encoder retriever to retrieve possible results from candidate responses or knowledge base, and then combine the encoder-decoder system to generate the final response. However, their experimental results show that this retrieve-and-refine system does not outperform the pure retrieval system on the dialogue system dataset.

Eventually, these research on dialogue systems is very similar to IR-QA and CMC approaches with similar task goals: i.e., predicting responses in multi-round dialogue situations using user queries and historical contexts, and sometimes using external knowledge. Nevertheless, although these studies have explored the use of retrieval-reading architectures to enhance the performance of dialogue tasks, few studies have focused on how to encode and select historical information more efficiently.

3 Data

In collaboration with a Dutch corporation, this research will use real industrial multilingual question answering data to validate all hypotheses. The data comes from real conversations conducted between bank customer service agents and customers via a mobile app and consists of Dutch and English and a small amount of other languages such as German. However, we do not have language labels for these data, hence we cannot provide accurate statistical information on what percentage of the data is in Dutch and English. Table.1 shows some statistics of our dataset and compares them with the QuAC[11] dataset. Overall, our dataset provides over 300k QA pairs containing over 100k complete dialogues. In addition to this, we observe that our dataset is more challenging than other ConvQA datasets such as QuAC, and CoQA[58], because the vast majority of questions in our dataset are non-factoid questions, and the questions and answers in our data are significantly longer than those in public datasets. Moreover, although the average history rounds are similar to the QuAC dataset, the maximum is much larger than the QuAC dataset, posing a more significant challenge for our study because some conversations will have excessively long history context lengths.

An English conversation example in our QA dataset

1. Q1: I download the bank app, and activated it, then the app showed me that I can't use it anymore because no permission to use Mobile Banking. My parents transferred me money on Friday, and I still didn't get it. What's the problem?
   A1: Were you able to use our app before or is this the first time you've installed the app?
2. Q2: I downloaded it for the first time, and it was ok at first. then I quit the app, and it was written that I can't use it anymore, because no allow to mobile banking.
   A2: Okay, can you still log in with your card and e.dentifier on our website in Online Banking?
3. Q3: yes, I think so.
   A3: Could you please try this?
4. Q4: Yes, I can log in online. I'm sorry for changing the subject, but this question is really important. My parents transferred me money on Friday and I still didn't get it. What is the problem? It will be nearly a week tomorrow. The bank from where the money was transferred told us that the bank got the money on Friday, but there is no money in my bank account.
   A4: From which country/ bank did your parents transfer the funds?
5. Q5: [country].
   A5: Then you will receive it within 5 working days. The moment it is in our bank it is directly on your account.
6. Q6: ok, thank you! and what about the mobile banking?
   A6: You can call us on [phone-number] so we can check that!
7. Q7: What is the best time to call? Because I called a lot before and always couldn't get the answer.
   A7: We are available from 08.00 in the morning.
8. Q8: Ok, thank you a lot.
   A8: Good luck and have a nice day.
9. Q9: By the way, to check money on the bank account you use e.identifier?
   A9: It is possible to check your account in Internet Banking or in the app. The first time to log in on both you need an e.dentifier.

	Our dataset	QuAC dataset
# Questions	339,478	90,922
# Dialogues	131,725	12,567
Below: Max, Mean, Min		
# Tokens per question	233, 24.5, 1	23, 6.5, 1
# Tokens per Answer	645, 30.0, 3	30, 12.6, 1
# Questions per Dialogue	46, 2.6, 1	12, 7.2, 4

Table 1: Data statistics summarizing in our dataset and the QuAC dataset

Above shows a real English conversation example in our dataset, it contains 9 rounds conversations conducted between a customer and a human agent. Sensitive information such as name, address, phone number is masked in this example, for example, the [phone-number] denotes the phone number provided by the agent, and the [country] denotes the counrty name mentioned by the customer. In real training and reasoning phases, we mask any sensitive information related to user information such as name, bank card number, address, etc. However, we keep non-sensitive information provided by the agent such as the bank-related phone number, website link that can help to answer the question.

The reason why our dataset is challenging:

- Conversations contain more redundant information, e.g., confirmatory information, repetitive conversations, irrelevant information to the question, questions and answers with errors and misspelling, etc.
- It is challenging to understand conversational information, e.g., 1. According to our observations, our data contains a high percentage of non-factoid questions such as questions about specific processes, inquires about the description of a product, and questions that rely on particular expertise. 2. In addition to the majority of our dataset being in English and Dutch, we also have a small number of conversations that are predominantly in French, German, etc.
- Our data contains a large number of phone numbers, website addresses, etc. Most of these information are enriched with real answers, which makes our task even more challenging because the majority of QA tasks exclude them.

Additionally, the reasons above indicate our QA data is noisier than other academic datasets. This certainly brings more challenges to our research and brings us more thoughts about the reality of the ConvQA problem. Furthermore, our QA dataset does not belong to the typical information retrieval-based QA datasets such as MS-MARCO[49], TriviaQA[30], or conversational machine comprehension datasets such as QuAC and CoQA. This is mainly because the former types of datasets provide many articles such as Wikipedia corpus that may contain answers and knowledge, while the CMC datasets provide the golden passage for each question that indeed contains answers. In contrast, our dataset only provides QA pairs, where each question's answer can be considered as a golden passage for that question. We also extend our passage collection by adding the official FAQ answers to augment the data.

4 Method

In this chapter, we describe the methodologies involved in this study and how we connected each component into a total system. We will first introduce the overview of the QA system in section.4.2, and then introduce the retriever, reader and other extension modules separately from section.4.3 to section.4.7 respectively.

4.1 Task Definition

The Conversational information retrieval-based QA task can be defined as follows: Given a large collection of passages D, a query U_k including the current question Q_k and its previous historical context H_k, where the $H_k = \{\mathrm{Q_i}, \mathrm{A_i}\}_{i=1}^{k-1}$ contains all QA pairs(questions and answers) before the Q_k in this conversation. The task goal is to provide an answer A_k for U_k using D. The answer A_k can be generated using generative model or greedily selected using the top-1 retrieved answer from the retriever. Furthermore, It is important to note that the complete A_k answer may come from single or multiple documents, or may not even exist in all existing documents. In addition, since our data is obtained from real-world customer service, we hardly have the Wikipedia-like articles used in academic data as D. Instead, we use answers from QA data as our D to retrieve; hence, in our case, a single element in D is a single answer in the dataset.

4.2 System Overview

Figure.2 shows our QA system. In general, we incorporate reranker, History summarization module (HSM), and Dynamic History Re-weighting Module (DHRM) in the retrieval-reading architecture to enhance the system. Furthermore, the retrieval-reading module consists of a neural bi-encoder retriever and a neural sequence-to-sequence generative reader. In our subsequent experiments, we will verify the utility of different modules for the overall retrieval and reading tasks.

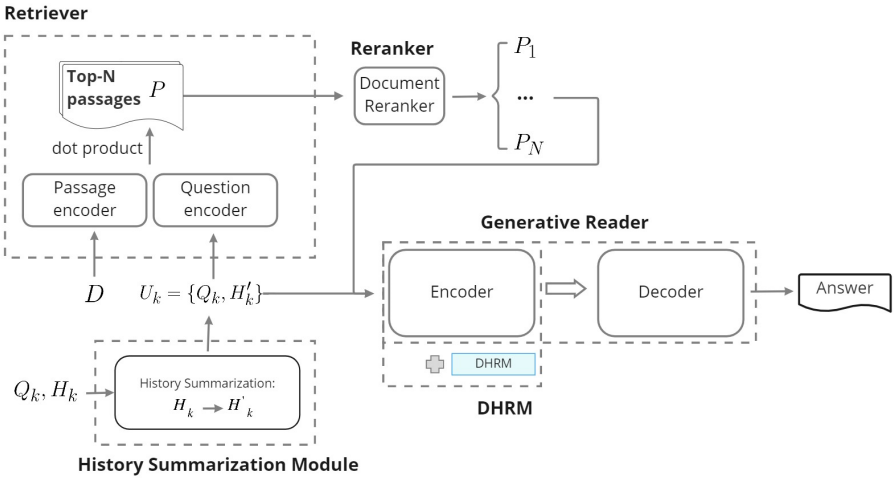

Figure 2: Architecture of the QA system in this study

4.3 Document Retriever

The retriever is an essential component of our QA system that will be fed the current question and historical context to retrieve relevant documents. High-quality Retrieved documents can provide more relevant background knowledge for subsequent generative readers, and ideally even its output can be used directly for our prediction of answers. To validate what model can yield more satisfactory results with real-world data, we investigate both sparse and dense retrievers in our retrieval task. Specifically, we implement the BM25 in the pyserini[41] and bi-encoder architecture in the DPR[33] as our document retrievers to trace the most top-N similar passages P in D.

BM25, as a type of sparse retriever, uses a sparse text representation by introducing the additional text length feature while considering term frequency (TF) and inverse document frequency (IDF). The equation 1 shows how BM25 retrieves documents given a question. Given the current question Q_k and its history H_k, we first concatenate them as a query input U_k, where u_i denotes the i-th token in the U_k. Then, When calculating the retrieval score between Q_k and a document d in D, the BM25 considers IDF score $IDF(u_i)$, the TF score $f(u_i, d)$, and the averaged document token length $\overline{L_D}$. Where $\overline{L_D}$ represents the averaged document token length computed using m documents in D. k_1 and b are hyperparameters in the BM25. It is worth noting that this sparse text representation vector does not take into account the order of occurrence of words in the document, thus BM25 can also be considered a bag-of-words approach.

$$Retrieval\_Score(U_k, d) = \sum_{i=1}^{n} IDF(u_i) \cdot \frac{f(u_i, d) \cdot (k_1 + 1))}{f(u_i, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{\overline{L_D}})} \tag{1}$$

$$\overline{L_D} = \frac{1}{m} \sum_{j=1}^{m} \#tokens \in D_j \tag{2}$$

In addition, as a neural approach, the bi-encoder model is based on the Bert encoder to compute the embeddings of both questions and documents dynamically, then use the maximum inner product (MIPS) to measure the relevance between them to provide the retrieval result. Formally, the bi-encoder retrieval process in our study can be represented as the process illustrated in the equation 3 and 4. We feed the current query U_k and a candidate documents d to the bi-encoder, where the question encoder encodes the query input and the document encoder encodes the document inputs individually. After that, the dot product is used to compute the relevance score between the E_{U_K} and E_d. During the retriever inference step, the bi-encoder retriever output N similar documents $P = \{\mathrm{p_i}\}_{i=1}^{N}$ for the input.

$$E_{U_K} = Q_{encoder}(U_K), E_d = D_{encoder}(d) \tag{3}$$

$$Retrieval\_Score(U_k, d) = E_{U_K}^{\top} E_d \tag{4}$$

One merit of the bi-encoder architecture is that it allows us to pretrain all the documents and questions first so that the overall concurrent training becomes more effective in the subsequent co-training with the generative reader. In addition, document representations can be calculated and cached before the inference step. Thus such offline encoding facilitates the inference process to speed up significantly for daily use in industry. We will make use of the Faiss[29] system to speed up this retrieval process. Furthermore, Since our experiments will be conducted on multilingual data, we replaced Bert with multilingual-Bert in the dual encoder model for efficient multilingual model training. We do not consider training at least two or more monolingual models to process data in different languages, as this would greatly increase training time and consume computational resources. Recent studies have demonstrated that multilingual LMs like multilingual-bert can save considerable computational resources without sacrificing much performance. Moreover, the multilingual environment does not have a significant impact on the sparse retriever, since most

of the data are based on Dutch and English only, and the words in these two languages are not particularly similar, therefore, we do not need to make any changes to BM25.

4.4 Generative Reader

On the basis of the retriever, We further explore the possibility of using the generative model to fill in the blank in the ConvQA field. The architecture of the sequence-to-sequence generative reader contains two sections: encoder and decoder. The encoder forms the contextualized representation given the query input. In contrast, the decoder is a classic auto-regressive model that aims to generate the answer. Equation 5 illustrates the formula for how the generative reader computes the probability. once the retriever provides N similar documents $P = \{\mathrm{p_i}\}_{i=1}^{N}$, the reader aims to read the concatenation of the query input U_k and its corresponding retrieved documents P_k to predict the answer autoregressively. This means it generates each token a_i in the answer A_k (total contains m tokens) by considering all background documents simultaneously. Moreover, We implement both the mBART[43] and mT5[71] models as our readers to validate which model will yield better results in our multilingual reading task.

$$Prob(A_k|U_k, P_k) = \prod_{i=1}^{m} prob(a_{k_i}|U_k, P_k, a_1, ...a_{i-1}) \tag{5}$$

As Equation 6 shows, consequently, we integrate the retriever and reader components into a complete system. Since P_k contains N candidate documents for U_k, we represent each candidate document as $P_{k_1}, P_{k_2}, ..., P_{k_N}$. Moreover, inspired by [33], we pretrain the retriever before the concurrent training to save the training time and make the whole system converge faster.

$$\begin{aligned} Prob(A_k|U_k) &= Prob(P_k|U_k) \prod_{i=1}^{m} prob(a_m|U_k, P_k, a_1, ..., a_{m-1}) \\ &= Prob(P_{k_1}, P_{k_2}, ..., P_{k_N}) \prod_{i=1}^{m} prob(a_i|U_k, P_{k_1}, ..., P_{k_N}, a_1, ..., a_{m-1}) \end{aligned} \tag{6}$$

4.5 History Summarization Module

Realistic industrial QA datasets have more noise, as evidenced by the presence of more confirmatory information in the conversation, redundant information, and information that is not relevant to understanding the current question. These noises can impair QA systems in retrieving relevant documents to a certain extent, thus we propose a History Summarization Module(HSM) to refine the historical context in an attempt to improve the both retrieval and reading performance. Our HSM can be easily used as an extension module for any retriever and readers and since it can only be implemented in an unsupervised way in our study, we implement TF-IDF based extractive summarization method because it is commonly used as a strong baseline.

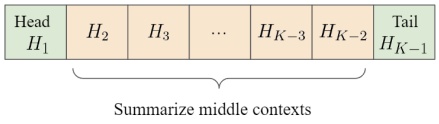

Figure 3: History Summarization Strategy

It is crucial to use the text summarizer in conversational QA tasks effectively. As shown in the Figure.3, we choose to keep the head H_1 and tail H_{k-1} pairs of historical contexts, and only summarize the content in the middle. This is because, intuitively, the head of a conversation normally contains the user's primary intent, while the tail is most likely to be relevant to the current question since it is the most recent context. Specifically, we first transform all tokens in the documents D

through stemming as our preprocessing step, which will bring better results for our summarization task. Then we calculate the TF and IDF values for tokens in D to measure the TF-IDF score of each sentence in the historical context that is to be refined, and finally remove the sentences below the threshold to compose complete sentences.

4.6 Dynamic History Re-weighting Module

Another focus of our research is dynamically adjusting the importance weights for historical context information to improve reading performance. More specifically, we hypothesize that some history turns are redundant for the current question, while others can help us understand and answer the current question. Thus, our intuition is that by dynamically assigning fewer weights to low-value historical contexts in the reading process, the reader can be more capable of handling the ConvQA tasks. Figure.4 and algorithm.1 shows how Dynamic History Re-weighting Module (DHRM) works with a generative reader. In general, DHRM learns the importance weight of historical QA pairs, then re-weights those historical turns and tokens in the passages. Technically, DHRM can be extended on any encoders of the generative reader.

We define the input U_k as the combination of current question Q_k, all history context turns $H_k = \{\mathrm{Q_i}, \mathrm{A_i}\}_{i=1}^{k-1}$, and retrieved candidate passages P_k, where H_i denotes the $i-th$ QA pairs in the conversation. After passing the input U_k to the encoder, we receive the contextualized representation; then, we implement mean pooling to compute the sequence level embedding for current question QS and all historical context QA turns, e.g. HS^1 denotes the sequence level embedding for H_1. The reason to acquire the sequence level embedding is that we will need to measure how important a historical context QA turn is, which will be processed by using the bahdanau attention[5].

Further we implement a bahdanau attention layer and a softmax layer to calculate the attention weights $\left\{\alpha^{\mathrm{i}}\right\}_{i=1}^{k-1}$ between the sequence level embedding of current question QS and the sequence level embedding of all historical context QA turns $\left\{\mathrm{HS^i}\right\}_{i=1}^{k-1}$. Hence, we hypothesize the attention socres α^i indicate how important to consider the historical context QA turn H_i to understand the current question Q_k. The next essential step is to utilize the attention weights. This process can be divided into two parts, one is to utilize attention scores to reweight corresponding historical context embedding. For example, DHRM utilizes α^1 to reweight all tokens in the H_1, and this step can be found as the history reweighting layer in the figure.4. The other step is to reweight the tokens in the candidate passages that also appeared in the specific historical context QA turn H_i, we demonstrate this step as the passage reweighting layer in the figure.4.

For example, in the figure.4, PT_1 denotes the first token in candidate passages P_k, and it also occurs in the second history turn; thus we multiply w^2 to the embedding of PT_1. Overall, DHRM first learns the importance scores of different historical contextual turns by introducing an additional attention mechanism to learn their relationship to the current question. Then, based on those scores, DHRM re-weights the corresponding historical rounds and re-weights the tokens in the candidate passages if they also appear in those historical rounds.

Algorithm 1 Dynamic History Re-weighting Mechanism

1: Input U_k: it includes the current question Q_k, historical contexts $H_k = \{\mathrm{Q_i}, \mathrm{A_i}\}_{i=1}^{k-1}$, and candidate passages P_k.
2: Feed U_k to the encoder to get the contextualized embedding.
3: Utilize mean pooling for the current question embedding and every historical contexts' embeddings. This process outputs $[QS, HS^1, HS^2, ..., HS^{k-1}]$.
4: Implement the Bahdanau attention layer to calculate the attention score for each historical context's embedding. After that, pass the attention scores to a softmax layer to compute their attention weights ($[\alpha^1, \alpha^2, ..., \alpha^{k-1}]$, they are ranging from $[0, 1]$).
5: Reweighting tokens in the historical contexts by multiplying its corresponding attention weights.
6: Reweighting tokens in the candidate passages. For example, if the token PT_1 also appeared in H_2, then multiply α^2 to the embedding of PT_1.

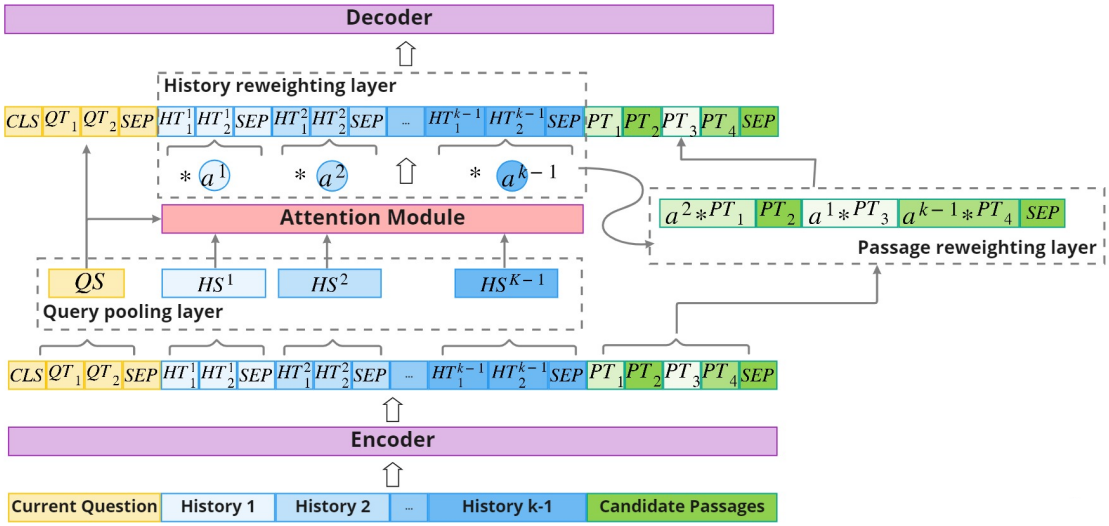

Figure 4: Dynamic History Re-weighting Mechanism architecture. To show how DHRM works, we visualize its architecture. In this graph, we denotes historical contexts $H_k = \{\mathrm{Q_i}, \mathrm{A_i}\}_{i=1}^{k-1}$ as History 1 to $k-1$.

4.7 Passage reranking Module

The retriever can retrieve the most relevant N candidate passages based on a given query input in the inference phase, where these passages are ranked in order of likelihood. when the retriever performs poorly, those retrieved passages can only contain real answers when N becomes significantly large. Nevertheless, feeding a large number of candidate passages to the generative model increases the computational resource consumption significantly and relies on a larger GPU memory. [55] has shown that adding a neural network-based reranker can improve the ranking results of passages and thus provide better background knowledge for subsequent reading tasks, even though the reranker and the dense retriever are based on the same language model without sharing parameters. The reranking strategy can be effective when we cannot obtain satisfactory retrieval results, especially when we cannot take advantage of the large number of background documents being retrieved. Therefore, we will also verify whether adding a passage reranker can improve the ranking of the retrieved candidate passages when we cannot incorporate a large number of candidate passages in the generative model.

We adopted the neural reranker architecture from [50] and modified it to fit our multilingual scenario by changing the bert to the multilingual-bert. In general, the intuition of reranker is similar to the sequence classification task. To illustrate, the reranker predicts whether the candidate passage contains the actual answer of a query. During the training phase of the reranker, the query input and one of its retrieved candidate passage is fed to the m-bert encoder. After the final layer from the encoder, a cross entropy layer is added to calculate the relevance score between the query and the candidate passage. During the inference phase, the reranker is fed with the query input and N candidate passages to calculate N relevance scores. Hence, we could use those relevance scores to rerank retrieved passages to yield better ranking results.

5 Experiments setup

This chapter provides experimental setup and technical details regarding our experiments and more detailed evaluation descriptions. We separate all experiments into retrieval tasks in section.5.1 and retrieval-reading tasks in section.5.2. The former covers the entire document-based retrieval experiments, and the latter carries out the reading experiments on top of the retrieval tasks.

5.1 Document retrieval task

5.1.1 Experiments of historical context composition

The document retrieval experiment is an essential part of this study. The composition category of historical contextual information determines its content. It can mainly be composed of three components, that is, historical questions, historical answers, and historical QA pairs. Therefore, we first investigated the impact of three different historical context compositions on the quality of machine retrieval. We consider the bi-encoder retriever from the dense passage retriever (DPR) as our potential base retriever. Furthermore, we implemented the Gradient-Cached DPR [19] (GC-DPR) to overcome the GPU memory restrictions because of the limitation of computational resources. GC-DPR is capable of saving at least half of the GPU memory without sacrificing the performance of retrieval, which makes our experiments more efficient. Lastly, we replaced the bert-base-cased model with the bert-base-multilingual-cased to suit our multilingual environment.

The following is a combination of three different historical contexts on the multilingual GC-DPR model; for simplicity, we refer to their prefixes as Retriever. It is important to note that the multilingual GC-DPR retriever has two different encoders, one of which encodes only query information (historical information and current question), and the other encodes only document information.

Retriever w/QAs: The query input includes the current question and all historical QA pairs.

Retriever w/Qs: The query input is the current question and all historical questions.

Retriever w/As: The query input is the current question and all historical answers.

5.1.2 Experiments of the retrieval task

To further validate the dense retriever's advantage, we also implement the sparse retriever (BM25) for the retrieval task. In the comparison experiments, we evaluate both models' retrieval accuracy of top-200 retrieved documents. Moreover, since we observed that historical QA pairs gave the best results in the initial retrieval experiments; thus, we apply historical QA pairs as the historical context component for all models in all the following retrieval experiments.

BM25: To compare the performance between non-neural retrieval approaches and neural-based retrievers, we implemented the bm25 algorithm adapted from pyserini[41] for our retrieval task.

Dense Retriever: We adapt the Retriever w/QAs in the previous subsection as our dense document retriever.

Dense Retriever + summarization: For this setting, we extend the dense retriever with a text summarizer to refine historical contexts. It is worth noticing that we do not perform refinement on other components of inputs, i.e., the current question and documents.

Dense Retriever + reranker: Inspired by [55], we extend the dense retriever with an extra neural document reranker to get more satisfactory background documents as well as improve the retrieval generalization ability.

5.2 Retrieval-Reading task

In the retrieval-reading task, we experimented with two different generative models, mT5 and mBart, as our readers. We adopted the dense retriever model(in section.5.1.2) in this task. In addition, to investigate whether our History Summarization Module (HSM) and Dynamic History Reweighting Module (DHRM) will yeild better reading quality in the reading task, we added it to the mT5 model.

Dense Retriever + mT5: We implement the mT5 model as our generative reader to encode the retrieved documents and the query input.

Dense Retriever + mBart: In this setting, we implement the mBart model as our reader.

Dense Retriever + mT5 + Summarization: For this setting, we extend the mT5 reader with a text summarizer to refine historical contexts. It is worth noticing that we do not perform refinement on other components of inputs, i.e., the current question and documents.

Dense Retriever + mT5 + Summarization + DHRM: Similarly, we extend the Dense Retriever with a text summarizer to refine historical contexts for this setting. Moreover, we incorporate the DHRM to our reader to validate whether it can yield better performance.

To further validate that background knowledge can help the machine generate more accurate answer predictions, we will conduct two additional experiments for comparison. First, we aim to investigate to what extent the output of information retrieval model can be compared with our retrieval-reading system. Therefore, we chose to implement only the retriever to output top-1 answers as the answer prediction. In addition, to verify that the combined retrieval-reading system can predict answers better than a pure reader, we set up an additional experiment using only query input as the generative reader input. In previous studies, using only generative readers like GPT2 or encoder-decoder models for QA and dialog system tasks can also largely outperform using only retrievers[79, 1, 59]. Therefore, our comparison experiments are intended to verify that using background knowledge can help the retrieval-reading system to make better predictions than using the generative reader alone.

Retrieved top-1 document as the final answer: For this setting, we implement the Dense Retriever + summarization from the section.5.1.2 to output the most possible answer.

mT5 without background knowledge: For this setting, we implement the mT5 model using only the query input which includes the current question and its historical contexts.

5.3 Distributed training

In this study, we utilize distributed computing to accelerate the entire training and inference process. Specifically, we combine distributed data parallelism and model parallelism to implement uniform distributed training through the Zero Redundancy Optimizer[56] (ZeRO) strategy and Pytorch platform[52]. In addition, We also introduce gradient accumulation[39], learning rate warm-up[42] techniques to make the model converge better.

Distributed data parallelism (DDP) first divides the overall data into multiple chunks based on the number of GPUs, then allows different data chunks to be deployed on different GPUs, and finally enables the gradients to be shared across multiple GPUs before the optimization step, thus ensuring that the model on all GPUs can have the same parameters[39]. This strategy accelerates the training and inference process at a rate that approximates the number of gpu's, which saves the computational resources. On the other hand, model parallelism (MP) divides a model into multiple parts and deploys them on different GPUs, but it does not change the original data input. Dividing a heavy model into multiple parts can make the whole training process more efficient, hence model parallelism is especially useful when the model is too large.

Nonetheless, using distributed data parallelism or model parallelism alone still faces limitations when the model takes up considerable memory and a batch of data takes up a large amount of memory simultaneously. ZeRO proposes an approach that combines both DDP and MP to save GPU memory at a lower GPU communication cost. Therefore, we applied ZeRO strategy in our experiments, which saved us significant computational power.

5.4 Evaluation metrics

For the evaluation metrics, we measure the average document rank for experiment 5.1.1. The average document ranking indicates the average ranking of the true answer to a question among a given large number of documents when the retriever is searching for the answer to that question. For experiment 5.1.2, we utilize top-n retrieval accuracy and rouge-1, rouge-2, and rouge-L scores[40] as our evaluation metrics. The former measures whether the true answer to the question is contained in the top-n document given by the retriever, while the latter is done by comparing the similarity between the true answer and the retrieved top-1 answer on the unigram, bigram and the Longest Common Subsequence criteria. In addition, we do not use metrics such as retrieval accuracy in experiment 5.1.1 because the retrieval performance needs to be verified by first encoding the entire documents in the dataset offline, which consumes significant computational resources and requires considerable time.

For the evaluation of the retrieval-reading experiment(5.2), similar to the previous experiment, we perform evaluation by using the quantitative criteria (rouge scores). Finally, to better verify whether a retrieval-reading model that considers background knowledge can generate better responses than a retriever, we invited internal experts from the company to conduct a questionnaire-based double-blinded experiment (examples in section.9). Specifically, given a question and its historical contexts , they are required to rate three different candidate answers (1.ground-truth answers; 2.top-1 answers returned by the retriever; 3.answers predicted by the retrieval-reading model) on the scale of relevance, correctness, and readability. The meaning of relevance, correctness, and readability in our questionnaire was explained in the preamble based on the language that the subjects had mastered. Moreover, to ensure the fairness of the experiment, we did not label the sources of these responses, and we randomized the order in which the answers from different sources appeared in the questionnaire.

We invited a total of eight subjects to participate in this questionnaire, four Dutch-speaking and four English-speaking, and each of them was given a different question for the assessment. Additionally, in order to reduce the questionnaire evaluation time, the subjects started with the real questions, which means that we did not give them practice or demonstration questions, but each subject can go back to the previous questions to change their rating. We invited a total of eight subjects to participate in this questionnaire, four speaking Dutch and four speaking English, and each of them was given different questions for the assessment. Each participant was given a maximum of 30 minutes to complete the assessment with three different responses to the eight questions, with an average completion time of 12 minutes and 84 seconds per participant. Finally,

we conducted a within-group Student's t-test to verify whether the three candidate answers were significantly different in terms of criteria.

6 Results

This chapter presents the result of our experiments. We will first demonstrate the result of our retrieval task in section.6.1 and section.6.2, and then we will illustrate the finding of retrieval-reading experiments in section.6.3.

6.1 Results of historical context composition

Our experiments show that considering both historical QA pairs as historical contextual compositions yields better results than using them alone. Specifically, we randomly selected 4000 questions in the validation set and then used trained models to predict the ranking of their actual documents among 32,000 documents. Table.2 presents the model with QA pairs as historical contexts ranked approximately 14 and 1.4 lower than the model with Questions and Answers as contexts, respectively, in the average rank. This indicates that historical questions and answers are crucial for the machine to understand the current question and answer it.

	Avg. rank
Retriever w/Qs	170.55
Retriever w/As	158.09
Retriever w/QAs	**156.69**

Table 2: The average rank of the actual documents in 32k documents for 4000 questions. (Lower is better)

6.2 Results of document retrieval task

Table.3 shows the results of all our experiments on the machine retrieval task. Overall, the dense retriever produces much better results than the sparse retriever, and the addition of the text summarizer yields the best results, while the introduction of the reranker does not improve the system. Among them, the dense retriever achieves more than 4 times the retrieval accuracy on the test set than BM25, which is only 6%. In addition, the additional text summarizer improves the dense retriever in almost all metrics, except for rouge-1 recall (negligible decrease 0.27%). For the retrieval accuracy top200 and top500 metrics, its implementation improves the dense retriever's performance by 1.68% and 2.24%, respectively. For rouge scores, the introduction of the text summarizer improves mainly the precision and f1 score metrics by roughly 5% and 3%, correspondingly.

However, experiments show that extending the reranker for the bi-encoder retrieval system does not help it to obtain a better ranking of retrieved text. Note that reranker does not change the retrieval accuracy because it only reorders the retrieved top-n documents. Thus we use reranker to sort the top 500 documents, and then measure the average rouge scores of the top 200 documents that are sorted. The results show that adding reranker to the system reduces all rouge scores (up to 0.68%) compared to the model without it.

Models	Retrieval Accuracy (Top 500)	Retrieval Accuracy (Top 200)	Avg. Rouge-1 score (Top 200)	Avg. Rouge-L score (Top 200)
BM25	-	6.00	-	-
Dense Retriever	34.34	24.75	17.96, **26.74**, 18.35	13.67, 20.80, 14.04
Dense Retriever + summarization	**36.58**	**26.43**	**23.16**, 26.47, **21.63**	**18.76**, **21.63**, **17.62**
Dense Retriever + summarization + reranker	36.58	26.12	23.02, 25.79, 21.22	18.74, 21.06, 17.32

Table 3: Evaluation of Retrieval Task on test set. Values in the bold font denotes they are the highest compared to values of other models. For rouge scores, the order of demonstration is Precision, Recall, F1 score. For both retrieval accuracy and rouge scores, higher means better. − means there is no need to continue to complete the corresponding experiment, this is mainly to save expensive computing resources. For example, we only validated the BM25 with top-200 retrieval accuracy because it performs much worse than Dense Retriever.

6.3 Results of Retrieval-Reading task

For the retrieval-reading task, we conduct both qualitative and quantitative evaluations to measure the result. Because the performance of retrieval task affect the reading task quality, thus we adopt the best retriever (Dense Retriever+summarization) in the retrieval task in all retrieval-reading experiments. We will present the quantitative and qualitative evaluation result respectively in this section.

6.3.1 Quantitative evaluation result

Table.4 and table.5 illustrates the complete quantitative evaluation result of the retrieval-reading task using three different rouge scores. Table.4 compares the performance of different generative models and presents whether incorporating more retrieved background documents can improve the final reading quality. In contrast, table.5 compares whether our additional custom modules improves the quality of the reader's output.

First of all, table.5 shows that When we employ only the generative reader without background knowledge, the results can slightly outperform the retriever-only model on all f1 rouge scores. This means that solely using the encoder-decoder model alone can serve as a strong baseline for our task. In addition, table.4 presents that when the number of background knowledge documents increases, the overall quality of the answers generated by the reader also improves. Specifically, when the mt5 reader considers 5 more retrieved documents, its results achieve more than 1.5 improvement on all rouge f1 scores. This result is consistent with our finding in the retrieval task that more retrieved documents indicate these documents are more likely to contain true implicit knowledge. In addition, results show that the mT5 model shows an advantage over the mBart model, especially the improvement of 1.94%, 1.15%, and 0.92% in the precision, recall, and f1 scores of rouge-2, respectively.

Furthermore, the text summarizer for historical contexts significantly improves the prediction quality of the reader for almost all rouge scores except rouge-2 recall. In particular, for all rouge scores in terms of f1 score, which are 0.63%, 0.56%, and 0.56%, respectively. However, the addition of DHRM does not result in better performance of the generative reader, i.e., a slight decrease in

all metrics. This finding indicates our hypothesis of dynamically adjusting the weights of different historical contexts cannot yield improvement on retrieval-reading task. Finally, we evaluated using only the top-1 document of the retriever as our answer to the metrics. The results show that our best model outperforms it in almost all metrics, except for the recall scores of rouge-1 and rouge-L. Such improvement is especially apparent for all rouge scores in terms of precision, which are 6.65%, 3.95%, and 6.68% respectively.

Models	Rouge-1 score	Rouge-2 score	Rouge-L score
Dense retriever + mT5 w/5 passages	27.87, 23.71, 22.78	8.84, 7.70, 7.50	23.49, 20.28, 19.34
Dense retriever + mT5 w/10 passages	**29.56**, 25.24, **24.39**	**10.69**, **10.05**, **9.01**	**25.17**, 21.69, **20.88**
Dense retriever + mBart w/10 passages	25.00, **26.97**, 23.14	8.75, 8.9, 8.09	20.9, **22.83**, 19.48

Table 4: Model comparison result of Retrieval-Reading Task on test set. This table compares the performance of the mT5 and mBart models, and compares whether incorporating more background documents helps. Values in the bold font denotes they are the highest compared to values of other models. For rouge scores, the order of demonstration is Precision, Recall, F1 score.

Models	Rouge-1 score	Rouge-2 score	Rouge-L score
Retrieved Top-1 document as final answer	23.43, **27.59**, 22.09	7.28, 8.84, 7.21	18.93, **22.66**, 18.01
mT5 without background knowledge	25.99, 23.58, 22.28	8.44, 8.15, 7.55	22.1, 20.55, 19.19
Dense retriever + mT5 w/10 passages	29.56, 25.24, 24.39	10.69, **10.05**, 9.01	25.17, 21.69, 20.88
Dense retriever + mT5 w/10 passages + Summarization	**30.08**, 25.98, **25.02**	**11.23**, 9.65, **9.57**	**25.61**, 22.35, **21.44**
Dense retriever + mT5 w/10 passages + Summarization + DHRM	29.30, 25.83, 24.68	10.46, 9.08, 8.94	24.79, 22.11, 21.01

Table 5: Quantitative Evaluation result of Retrieval-Reading Task on test set. Values in the bold font denotes they are the highest compared to values of other models. For rouge scores, the order of demonstration is Precision, Recall, F1 score. We implemented the best retrieval model for the Retrieved Top-1 model.

6.3.2 Qualitative evaluation result

Figure.5 and table.6 demonstrate the qualitative evaluation result of the retrieval-reading task on the basis of a double-blinded human experiment. In the experiment, a total of 64 questions (32 in English and 32 in Dutch) and their corresponding historical contexts were randomly selected from the test set as examples for the questionnaire. We then provided participants with those examples and asked them to rate(from 0 to 10) the three candidate answers for each question on the scale of relevance, correctness, and readability. The three types of candidate answers are: 1.ground-truth answers; 2.top-1 answers returned by the retriever; 3.answers predicted by the generative reader(we use the best retrieval-reading model in this experiment). Since the experiment was double-blinded, thus we did not tell the participants the source of these answers.

The results in figure.5 show the answer scores of retriever, generative reader, and human in a

stepwise manner for the three criteria. In addition, we also performed a within-subjects pairwise student's t-test on the results, and more details can be observed in table.6. Among them, we can observe that for correctness and readability, the predictions of the generative reader are statistically significantly better than those of the retriever. Lastly, human answers significantly outperform answers of both models in all metrics.

Finally, we randomly sampled a few examples from the test set and analyzed the three candidate answers by eyeballing. One major observation is the answers provided by the generative reader rely heavily on the background documents provided by the retriever. Thus, it is difficult for the generative reader to provide significantly better predictions when the documents provided by the retriever are not relevant to the input question. The same finding can be observed from our qualitative evaluation results, i.e., there is no statistically significant improvement in the scores of the generative model on the relevance metric. Moreover, when both models are unable to understand the input question, they sometimes tend to provide answer predictions in different languages. For example, when the current question is in Dutch, their predictions are probably in English, and these predictions are not relevant to the question. Therefore, in our experiments, the generative model outperforms the answers provided by using retriever directly in both qualitative and quantitative analysis, yet the predictions are much worse than the real human answers.

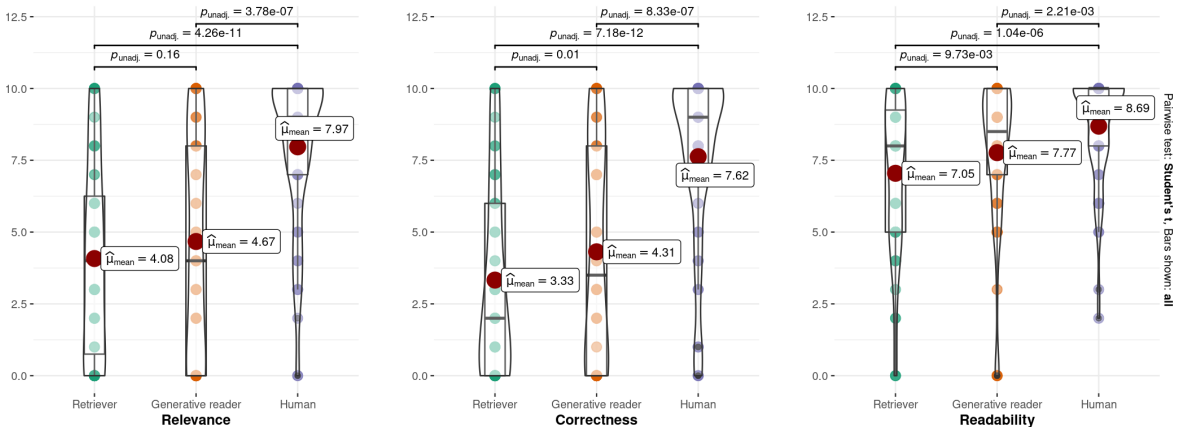

Figure 5: Qualitative Evaluation result of Retrieval-Reading Task on test set. The figure shows the results of experts' ratings on relevance, correctness and readability for three candidate answers. Where these scores are in the range of [0, 10], higher means better. It also illustrates the pairwise t test result where p_{unadj} represents p-value without adjustment.

	Relevance p-value	Correctness p-value	Readability p-value
Retriever - Generative reader	1.60e-1	1.00e-2 $*$	9.72e-3 $**$
Retriever - Human	4.26e-11 $**$	7.18e-12 $**$	1.04e-06 $**$
Generative reader - Human	3.78e-07 $**$	8.33e-07 $**$	2.21e-03 $**$

Table 6: Statistical significance test for qualitative evaluation of Retrieval-Reading task on test set. Table shows for A-B, whether B yields statistically significant improvement than A over Relevance, Correctness, and Readability score. where $**$ means $p < 0.05$, $*$ means $p < 0.01$.

7 Discussion

In this section, we discuss observations in the results chapter (chapter.6). For consistency, we will continue to follow the order of each section of the results chapter, and we will discuss possible future work at the end in the section.7.4.

7.1 Historical context composition

The results of experiment 6.1 show that both historical questions and historical answers are important, where the latter is more significant than the former, which is reflected in the validation results shown in table.2. Therefore, considering historical QA pairs on our data helps to better perform the retrieval task. This is consistent with the intuition that the current question in a conversation is likely to be an additional follow-up question to the historical answer, or possibly a supplementary and extended question to the historical question. In addition, our findings are similar to some studies conducted on public academic datasets, such as [58, 83].

Furthermore, our experiments show that Retriever w/QAs has an average ranking of 156.69 in the validation. Even though this result achieves the best score compared to other settings, there still are many irrelevant documents that have a high ranking. This issue will affect the performance of the retrieval-reading task experiments because it is not practical for the generative reader to input more than 50 retrieved background documents. It would greatly increase the computational resource usage, and we cannot guarantee that the generative reader can actually use such a large number of documents to answer the questions. Therefore, we introduced a reranker in the section.5.1.2 attempting to improve the ranking order of retrieved documents in order to better help the retrieval-reading task.

7.2 Document retrieval task

On the task of document retrieval (experiment 6.2), first of all, it is clear from our results that dense retriever shows a strong superiority over sparse retriever, which means that the neural-based retriever can serve as a strong baseline on real data in industry. This is consistent with findings of some previous studies[33, 38]. Moreover, when the QA dataset is strongly verbalized and informalized, the introduction of a textual summarizer to refine the historical context can effectively improve the performance of the retriever and thus offers a good value in real-life applications. Specifically, our approach can improve the F1 score by up to 3.58% on the retrieval task using TF-IDF; hence, in contrast to previous studies on question rewriting[68, 3, 12], our approach is very easy to implement in the real world application and it does not require the large computational resources that neural models require.

Finally, we verified that the introduction of reranker does not achieve better document ranking for retrievers that use neural model-based retrievers. The evidence derives from the fact that there is a small decrease (less than 0.5%) in all rouge metrics after adding the reranker. This may be due to the fact that both reranker and retriever are based on the same neural model, i.e., multilingual bert, and they both update encoder parameters by similar learning tasks in principle, so the introduction of reranker does not bring more advantages. Thus, Our findings are not consistent with the conclusions in [55].

7.3 Retrieval-Reading task

In the retrieval-reading task, our results of experiment 6.3 show that the seq2seq-based generative model can take advantage of the latent background knowledge to make better predictions. Even though the generative reader without introducing background documents has shown comparability in the results, the system with the addition of background knowledge and additional summarization can have a large improvement in all rouge scores, especially in the precision (up to 4.09% for rouge-1). Our findings are consistent with those of some previous systems[21, 38, 27, 16], yet also contradict those of others[45, 60]. The qualitative experiment result indicate the retrieval-reading model yields statistically significantly improvement than using only the retriever in terms of correctness and readability criteria. Moreover, when the generative model takes more background documents into account, its predictions are more similar to the actual answers. This finding is consistent with our intuition because more background knowledge is more likely to contain implicit information about the answer to the question, and generative models have the ability to exploit this implicit information.

In addition, when the textual summarizer is introduced to refine the historical context, the generative model can also produce predictions that are closer to the actual answer. This is likely because the refined context brings less noisy information to the reader, thus allowing the reader to learn more concise and organized historical information. Therefore, the text summarizer can improve the performance of the machine on both retrieval and reading tasks in a very efficient way. It can be easily deployed on almost any QA system that has a retriever or a reader, and is potentially adaptable to most multilingual environments.

Finally, we observed that adding additional attention mechanisms (DHRM) to the generative model to force it to pay less attention to historical information did not meet our expectations. This may be due to the fact that our data has fewer historical context rounds in general and thus does not have enough data to learn and update parameters for additional attention networks. In addition, this additional attention module may force the generative model to ignore historical information deliberately. Historical information that is useful for understanding the current question is also disregarded. Therefore, a possible future direction for our study is to observe what questions are more likely to be influenced by DHRM through attention visualization. In addition, performing error case analysis on DHRM and other models in ablation analysis might also give us more evidence. Although no existing studies have applied similar mechanism to the generative reader as we have done in the ConvQA domain, only HAM[53] and HAE[54] have applied similar strategies to the extractive reader and achieved good improvements. Our findings are contrary to them, suggesting that for generative models we have to consider other more efficient approaches in the future work.

7.4 Furture work

For future work, the text summarization module is still a direction worthy of attention. Specifically, this study only focused on using extractive summarizer to refine the historical information, however, for subsequent work, using a mature unsupervised abstractive summarizer may be a perspective to improve this work. Additionally, it might be worthwhile to explore other extractive summarization methods in the ConvQA task, such as methods based on clustering, TextRank. Moreover, it might be an interesting direction to compare or mix the summarization and question rewriting in the experiment. For instance, instead of just streamlining the historical information, another attempt could be to rewrite the refined historical information and the current question into a condensed query, but for realistic data, we may only focus on unsupervised rewriting methods.

We also observed that some of the retrieved or generated answers were reasonable but were formulated differently from the actual answers, while others were predicted incorrectly. Therefore, in

this case, the rouge metrics may not be able to distinguish reliably and accurately between these two cases. Moreover, rouge metrics also cannot measure instances of cross-lingual answers, i.e., answers may be described in another language even though they are semantically close to the actual answer. Therefore, we will consider improving the use of more semantic metrics in the future work. An alternative possible future perspective is to enhance the data by using the method of machine translation. For example, monolingual data can be extended with another monolingual data by using back-translation, and it can also be used to generate cross-lingual data to enhance the generalization ability of the model.

Eventually, the continuous improvement of the information retrieval module may be a perspective worthy of attention, as its results greatly influence the quality of subsequent generative reader predictions. In real-world applications, the quality of the data will greatly affect the results produced by the retriever, and adding user intent recognition to the data may be one of the future directions to improve the overall system.

8 Conclusion

In this thesis, we propose a neural retrieval-reading system with customized modules to investigate the possibility of using potential background knowledge to answer questions and explore a few directions to leverage historical contextual information in real-world ConvQA scenarios. We conduct several experiments using our system on multilingual conversational question answering tasks with an industrial customer service dataset. Overall, The experimental results show that in comparison to using only the retriever, our model yields up to 3.43% improvement in f1 and 6.7% improvement in precision, and also significant improvement in correctness and readability. In addition, our model makes efficient use of candidate knowledge and historical contexts, which is demonstrated by up to 4.01%, 2.48%, and 2.74% improvement in the precision, recall, and f1 metrics.

RQ1: Is the machine capable of using candidate background knowledge to answer questions?

To answer the first research question, we proposed and implemented a neural dense retriever to retrieve relevant background documents and then feed those documents to a sequence-to-sequence generative reader to predict the answer. Our quantitative experiment results indicate that such retrieval-reading architecture model can make better predictions compared to the top-1 output using only the retriever. In addition, the qualitative experiment result has shown that the retrieval-reading model provided statistically significantly better predictions than the retriever's outputs in terms of correctness and readability criteria. However, the performance of our model is still significantly lower than the level of actual human responses. Therefore these findings show that machines can utilize potential background documents to perform ConvQA tasks better, yet they cannot reach the human benchmark.

RQ2: How do machines leverage historical contextual information for real-world ConvQA tasks?

We explore its possibility from three different directions for the second research question. Our first direction perspective (RQ2.1) is to investigate what historical contextual information helps the machine to yield better retrieval performance. In addition, we hypothesize that refining the historical context can improve both retrieval and reading quality (RQ2.2). Furthermore, we propose an additional attention module to dynamically adjust the importance weights of different historical contexts in an attempt to improve the machine reading performance (RQ2.3).

RQ2.1: How do different historical information compositions affect a machine's retrieval performance?

In our experiments, we divided the historical information compositions to three different settings according to their properties, namely 1. previous questions; 2. previous answers; 3. previous QA pairs. The results indicate that considering all historical QA pairs (both previous questions and answers) for the retriever yields the best performance, yet the retriever produces the worst result when only historical questions are considered. This suggests all historical contexts (previous QA pairs) contain essential information to help machines understand and answer questions. There is a significant decrease in the quality of retrieval when ignoring any of the historical questions and answers.

RQ2.2: For real-world ConvQA tasks, refining the historical contexts help the machine perform retrieval and reading tasks?

To answer this research question, we proposed a summarizer based on TD-IDF that refines the

intermediate historical contexts while preserving the head and tail historical contexts. We implemented the summarizer in both the retrieval and generation tasks. Our results verified that introducing a textual summarizer to refine the historical context can effectively improve the performance of both the retriever and reader. This additional summarization module offers a good value in real-life ConvQA applications.

RQ2.3: Is it possible to improve machine reading performance by forcing machines to pay less attention to low-value historical information?

For this research question, we introduced an additional attention module for the generative reader in order to force the machine dynamically adjusts the weights of different historical contexts. Our intuition is the machine reading performance can be improved by paying less attention to low-value historical information. However, our experiment result shows it is hard for machines to learn how to pay less attention to non-valuable contexts dynamically. This is evidenced by the fact that the machine becomes slower to converge during training, and the final results also show that adding this module produces worse quantitative evaluation result.

As for future applications, our system will be used in real-world applications to assist customer service agents in the banking industry. Our system can reason fast and provide its predicted results as a template option to internal staff to assist their work and improve their efficiency in providing responses. Therefore, we employ our system in actual daily use to generate answers that can be accessed as a convenient template to speed up the response process rather than as an automated response bot.

9 References

[1] Daniel Adiwardana et al. “Towards a human-like open-domain chatbot”. In: *arXiv preprint arXiv:2001.09977* (2020).

[2] Mehdi Allahyari et al. “Text summarization techniques: a brief survey”. In: *arXiv preprint arXiv:1707.02268* (2017).

[3] Raviteja Anantha et al. “Open-domain question answering goes conversational via question rewriting”. In: *arXiv preprint arXiv:2010.04898* (2020).

[4] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. “On the cross-lingual transferability of monolingual representations”. In: *arXiv preprint arXiv:1910.11856* (2019).

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[6] Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. “A survey on machine reading comprehension systems”. In: *arXiv preprint arXiv:2001.01582* (2020).

[7] Alexander Bartl and Gerasimos Spanakis. “A retrieval-based dialogue system utilizing utterance and context embeddings”. In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2017, pp. 1120–1125.

[8] Cen Chen et al. “CAT-BERT: A Context-Aware Transferable BERT Model for Multi-turn Machine Reading Comprehension”. In: *International Conference on Database Systems for Advanced Applications*. Springer. 2021, pp. 152–167.

[9] Danqi Chen et al. “Reading wikipedia to answer open-domain questions”. In: *arXiv preprint arXiv:1704.00051* (2017).

[10] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[11] Eunsol Choi et al. “Quac: Question answering in context”. In: *arXiv preprint arXiv:1808.07036* (2018).

[12] Zewei Chu et al. “How to ask better questions? a large-scale multi-domain dataset for rewriting ill-formed questions”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 7586–7593.

[13] Jonathan H Clark et al. “TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 454–470.

[14] Pradeep Dasigi et al. “A dataset of information-seeking questions and answers anchored in research papers”. In: *arXiv preprint arXiv:2105.03011* (2021).

[15] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[16] Emily Dinan et al. “Wizard of wikipedia: Knowledge-powered conversational agents”. In: *arXiv preprint arXiv:1811.01241* (2018).

[17] Fahim Faisal, Sharlina Keshava, Antonios Anastasopoulos, et al. “Sd-qa: Spoken dialectal question answering for the real world”. In: *arXiv preprint arXiv:2109.12072* (2021).

[18] Tingchen Fu et al. “Learning towards conversational ai: A survey”. In: *AI Open* 3 (2022), pp. 14–28.

[19] Luyu Gao et al. “Scaling deep contrastive learning batch size under memory limited setup”. In: *arXiv preprint arXiv:2101.06983* (2021).

[20] Somil Gupta, Bhanu Pratap Singh Rawat, and Hong Yu. “Conversational machine comprehension: a literature review”. In: *arXiv preprint arXiv:2006.00671* (2020).

[21] Kelvin Guu et al. “Realm: Retrieval-augmented language model pre-training”. In: *arXiv preprint arXiv:2002.08909* (2020).

[22] Karl Moritz Hermann et al. “Teaching machines to read and comprehend”. In: *Advances in neural information processing systems* 28 (2015), pp. 1693–1701.

[23] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[24] Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. “Flowqa: Grasping flow in history for conversational machine comprehension”. In: *arXiv preprint arXiv:1810.06683* (2018).

[25] Hsin-Yuan Huang et al. “Fusionnet: Fusing via fully-aware attention with application to machine comprehension”. In: *arXiv preprint arXiv:1711.07341* (2017).

[26] Samuel Humeau et al. “Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring”. In: *arXiv preprint arXiv:1905.01969* (2019).

[27] Gautier Izacard and Edouard Grave. “Leveraging passage retrieval with generative models for open domain question answering”. In: *arXiv preprint arXiv:2007.01282* (2020).

[28] Yimin Jing, Deyi Xiong, and Yan Zhen. “BiPaR: A bilingual parallel dataset for multilingual and cross-lingual reading comprehension on novels”. In: *arXiv preprint arXiv:1910.05040* (2019).

[29] Jeff Johnson, Matthijs Douze, and Hervé Jégou. “Billion-scale similarity search with gpus”. In: *IEEE Transactions on Big Data* (2019).

[30] Mandar Joshi et al. “Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension”. In: *arXiv preprint arXiv:1705.03551* (2017).

[31] Ying Ju et al. “Technical report on conversational question answering”. In: *arXiv preprint arXiv:1909.10772* (2019).

[32] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

[33] Vladimir Karpukhin et al. “Dense passage retrieval for open-domain question answering”. In: *arXiv preprint arXiv:2004.04906* (2020).

[34] Veton Kepuska and Gamal Bohouta. “Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home)”. In: *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*. IEEE. 2018, pp. 99–103.

[35] Tomáš Kočiskỳ et al. “The narrativeqa reading comprehension challenge”. In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 317–328.

[36] Zhenzhong Lan et al. “Albert: A lite bert for self-supervised learning of language representations”. In: *arXiv preprint arXiv:1909.11942* (2019).

[37] Mike Lewis et al. “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. In: *arXiv preprint arXiv:1910.13461* (2019).

[38] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive nlp tasks”. In: *arXiv preprint arXiv:2005.11401* (2020).

[39] Shen Li et al. “Pytorch distributed: Experiences on accelerating data parallel training”. In: *arXiv preprint arXiv:2006.15704* (2020).

[40] Chin-Yew Lin. “Rouge: A package for automatic evaluation of summaries”. In: *Text summarization branches out*. 2004, pp. 74–81.

[41] Jimmy Lin et al. “Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021, pp. 2356–2362.

[42] Liyuan Liu et al. “On the variance of the adaptive learning rate and beyond”. In: *arXiv preprint arXiv:1908.03265* (2019).

[43] Yinhan Liu et al. “Multilingual denoising pre-training for neural machine translation”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 726–742.

[44] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[45] Shayne Longpre, Yi Lu, and Joachim Daiber. "MKQA: A linguistically diverse benchmark for multilingual open domain question answering". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 1389–1406.

[46] Ryan Lowe et al. "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems". In: *arXiv preprint arXiv:1506.08909* (2015).

[47] Yuning Mao et al. "Generation-augmented retrieval for open-domain question answering". In: *arXiv preprint arXiv:2009.08553* (2020).

[48] Sewon Min et al. "AmbigQA: Answering ambiguous open-domain questions". In: *arXiv preprint arXiv:2004.10645* (2020).

[49] Tri Nguyen et al. "MS MARCO: A human generated machine reading comprehension dataset". In: *CoCo@ NIPS*. 2016.

[50] Rodrigo Nogueira and Kyunghyun Cho. "Passage Re-ranking with BERT". In: *arXiv preprint arXiv:1901.04085* (2019).

[51] Yasuhito Ohsugi et al. "A simple but effective method to incorporate multi-turn context with BERT for conversational machine comprehension". In: *arXiv preprint arXiv:1905.12848* (2019).

[52] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[53] Chen Qu et al. "Attentive history selection for conversational question answering". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2019, pp. 1391–1400.

[54] Chen Qu et al. "BERT with history answer embedding for conversational question answering". In: *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 2019, pp. 1133–1136.

[55] Chen Qu et al. "Open-retrieval conversational question answering". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2020, pp. 539–548.

[56] Samyam Rajbhandari et al. "Zero: Memory optimizations toward training trillion parameter models". In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2020, pp. 1–16.

[57] Pranav Rajpurkar et al. "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250* (2016).

[58] Siva Reddy, Danqi Chen, and Christopher D Manning. "Coqa: A conversational question answering challenge". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 249–266.

[59] Adam Roberts, Colin Raffel, and Noam Shazeer. "How Much Knowledge Can You Pack Into the Parameters of a Language Model?" In: *arXiv preprint arXiv:2002.08910* (2020).

[60] Stephen Roller et al. "Recipes for building an open-domain chatbot". In: *arXiv preprint arXiv:2004.13637* (2020).

[61] Elayne Ruane, Abeba Birhane, and Anthony Ventresque. "Conversational AI: Social and Ethical Considerations." In: *AICS*. 2019, pp. 104–115.

[62] Sashank Santhanam et al. "Local knowledge powered conversational agents". In: *arXiv preprint arXiv:2010.10150* (2020).

[63] Minjoon Seo et al. "Bidirectional attention flow for machine comprehension". In: *arXiv preprint arXiv:1611.01603* (2016).

[64] Marco Antonio Calijorne Soares and Fernando Silva Parreiras. "A literature review on question answering techniques, paradigms and systems". In: *Journal of King Saud University-Computer and Information Sciences* 32.6 (2020), pp. 635–646.

[65] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[66] Chongyang Tao et al. "Building an Efficient and Effective Retrieval-based Dialogue System via Mutual Learning". In: *arXiv preprint arXiv:2110.00159* (2021).

[67] Adam Trischler et al. "Newsqa: A machine comprehension dataset". In: *arXiv preprint arXiv:1611.09830* (2016).

[68] Svitlana Vakulenko et al. "Question rewriting for conversational question answering". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 355–363.

[69] Alessandro Vallin et al. "Overview of the CLEF 2005 multilingual question answering track". In: *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer. 2005, pp. 307–331.

[70] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[71] Linting Xue et al. "mT5: A massively multilingual pre-trained text-to-text transformer". In: *arXiv preprint arXiv:2010.11934* (2020).

[72] Rui Yan et al. "Multilingual COVID-QA: Learning towards global information sharing via web question answering in multiple languages". In: *Proceedings of the Web Conference 2021*. 2021, pp. 2590–2600.

[73] Zhao Yan et al. "Docchat: An information retrieval approach for chatbot engines using unstructured documents". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 516–525.

[74] Wei Yang et al. "End-to-end open-domain question answering with bertserini". In: *arXiv preprint arXiv:1902.01718* (2019).

[75] Mark Yatskar. "A qualitative comparison of CoQA, SQuAD 2.0 and QuAC". In: *arXiv preprint arXiv:1809.10735* (2018).

[76] Yi-Ting Yeh and Yun-Nung Chen. "FlowDelta: modeling flow information gain in reasoning for conversational machine comprehension". In: *arXiv preprint arXiv:1908.05117* (2019).

[77] Fei Yuan et al. "Enhancing answer boundary detection for multilingual machine reading comprehension". In: *arXiv preprint arXiv:2004.14069* (2020).

[78] Munazza Zaib et al. "Conversational Question Answering: A Survey". In: *arXiv preprint arXiv:2106.00874* (2021).

[79] Yizhe Zhang et al. "Dialogpt: Large-scale generative pre-training for conversational response generation". In: *arXiv preprint arXiv:1911.00536* (2019).

[80] Jing Zhao et al. "RoR: Read-over-Read for Long Document Machine Reading Comprehension". In: *arXiv preprint arXiv:2109.04780* (2021).

[81] Hao Zhou et al. "Eva: An open-domain chinese dialogue system with large-scale generative pre-training". In: *arXiv preprint arXiv:2108.01547* (2021).

[82] Li Zhou et al. "The design and implementation of xiaoice, an empathetic social chatbot". In: *Computational Linguistics* 46.1 (2020), pp. 53–93.

[83] Chenguang Zhu, Michael Zeng, and Xuedong Huang. "Sdnet: Contextualized attention-based deep network for conversational question answering". In: *arXiv preprint arXiv:1812.03593* (2018).

[84] Fengbin Zhu et al. "Retrieving and reading: A comprehensive survey on open-domain question answering". In: *arXiv preprint arXiv:2101.00774* (2021).

Appendices

A Questionnaire examples of retrieval-reading experiment

To further validate whether a retrieval-reading model that considers background knowledge can generate better responses than a retriever, we invited internal experts from the company to conduct a questionnaire-based double-blinded experiment. In the experiment, a total of 64 questions (32 in English and 32 in Dutch) and their corresponding historical contexts were randomly selected from the test set as examples for the questionnaire. The experiment result can be found in section 6.3, and below are some of the examples in the questionnaire. For faster viewing, we have bolded the core parts of questions and the reasonable answer fragments. Note that we have not bolded any fragments in the real questionnaire.

A.0.1 English Examples

Current Question is:
hi [NAME], so i have applied for a bank account and made an appointment for the id checking. then i talked with the amroep staff and he told that he need to see the brp document. long story short i couldn't show it to him at the appointment day and it just arrived yesterday. **is it possible to reschedule the appointment?** i would like to get a guidance about this condition.

Real Answer is:
That is possible! For this you need to **make an appointment with AMP group again**. You should have had an e-mail or text message, via that way you can contact them.

Retrieved Top-1 Answer is:
Thank you! That is indeed our email, but since you already got a visit from AMP it is not necessary to reply.

Generative Reader's Answer is:
Ah I see, well you can contact them on these details **to make an appointment with AMP** Group:Phone: 030-2408020 (8:00-21:00, za 8:00-16:30) WhatsApp: 0638035877.

Figure 6: English example 1

Current Question is:
Hello i **lost my bank card**. What must i do?

Real Answer is:
Sorry to hear you lost your debit card. **Do you have access to the ABN AMRO app**?

Retrieved Top-1 Answer is:
Ah excellent, then you can use the card overview menu in the app, **to block your current damaged card, and apply for a new one**. The new card should arrive within 5 working days.

Generative Reader's Answer is:
Then you can **replace the card** via the following link: https://abn.com/35lMUWt. They can **apply for a new card**.

Figure 7: English example 2

A.0.2 Dutch Examples

Original:
Current Question is:
Na gebruik van Tikkie krijg ik voortdurend een melding in mijn scherm" Tikkie blijft stoppen". Ondanks dat ik de app verwijderd heb en de telefoon helemaal afgesloten en opnieuw opgestart heb. Graag jullie hulp.

Real Answer is:
Waar zie je de melding? Is dat een sms of een pop up. kan je een screenshot maken en hier sturen?

Retrieved Top-1 Answer is:
Wat vreemd! Begrijpelijk dat je dit wilt weten. Via Whatsapp heb ik zelf helaas geen inzage in je gegevens. Het kan 6 werkdagen duren voor dit is verwerkt in ons systeem. Ik zou je adviseren om te bellen met ons op het nummer 088 22 62 611 zodat wij kunnen nakijken of dit goed geregistreerd staat.

Generative Reader's Answer is:
Wat vervelend dat dit gebeurt. Ik raad je aan om hiervoor even onze helpdesk persoonsgegevens te bellen op 088-2262611, zij kunnen je hier verder mee helpen.

Translated:
Current Question is:
After using Tikkie I constantly get a message in my screen "**Tikkie keeps stopping**". Despite deleting the app and shutting down the phone completely and restarting it. Please help.

Real Answer is:
Where do you see the message? Is that a text message or a pop up. can you take a screenshot and send it here?

Retrieved Top-1 Answer is:
That's weird! It's understandable that you want to know this. Unfortunately, I do not have access to your data via Whatsapp. It can take 6 working days for this to be processed in our system. I would advise you to call us on 088 22 62 611 so that we can check whether this is properly registered.

Generative Reader's Answer is:
How annoying that this is happening. **I advise you to call our personal data helpdesk** on 088-2262611, they can help you with this.

Figure 8: Dutch example 1

Original:
Current Question is:
Goedenmiddag. Wat is een swift code. Bic code. Hoe kan ik weten wat mij bic code is. Voor dat mensen uit buiteland geld kunnen storte??? Oke ik wacht. Hey.

Real Answer is:
Excuus dat je op ons antwoord moest wachten! Is je vraag nog relevant of ben je er inmiddels uit? Mocht ik nog iets kunnen doen, laat het dan weten.

Retrieved Top-1 Answer is:
de SWIFT code van ABN AMRO is hetzelfde als de BIC-code. Deze is ABNANL2A. Ik kan je denk ik niet verder helpen met deze website, omdat ik die ook niet ken. Als je hulp nodig hebt bij het invoeren neem dan contact op met de klantenservice van dat bedrijf. Als je de mogelijkheid hebt om via een overboeking binnen internet bankieren te betalen help ik je daar natuurlijk graag mee. Vraag dan bij het bedrijf hun gegevens op: Naam Rekening (IBAN)BIC of Swift codeBedrag Omschrijving. Daarmee kunnen we dan de betaling invoeren in Internet Bankieren.

Generative Reader's Answer is:
De BIC van ABN AMRO is ABNANL2A.

Translated:
Current Question is:
good afternoon. **What is a swift code. bic code**. How can I know what my bic code is. For people from abroad to be able to deposit money. † Okay I'll wait. hey.

Real Answer is:
Sorry you had to wait for our reply! Is your question still relevant or have you already figured it out? If there's anything else I can do, please let me know.

Retrieved Top-1 Answer is:
the ABN AMRO SWIFT code is the same as the BIC code. **This one is ABNANL2A.** I don't think I can help you further with this website, because I don't know it either. If you need help entering it, please contact that company's customer service. If you have the option to pay via bank transfer via internet banking, I will of course be happy to help you with that. Request their details from the company: Account Name (IBAN)BIC or Swift codeAmount Description. We can then enter the payment in Internet Banking.

Generative Reader's Answer is:
The BIC of ABN AMRO is ABNANL2A.

Figure 9: Dutch example 2

Current Question is:
Hallo, ik heb schoenen besteld via marktplaats en heb betaald met een tikkie nu zou diegene mij de track and trace sturen maar kreeg dus een ongeldige track and trace nu zegt de persoon dat het een tijdelijke was en een nieuwe heeft maar laat niks meer van zich horen kan ik mijn geld terugkrijgen?

Real Answer is:
Dat klinkt niet goed. Het is net als met een normale overboeking niet mogelijk om het geld terug te boeken via Tikkie. Als het om oplichting gaat dan raad ik aan om aangifte bij de politie te doen. In dat geval ontvangen wij ook graag meer informatie, zoals een screenshot van de afschrijving.

Retrieved Top-1 Answer is:
Dat is balen zeg! Helaas kan ik je hier niet mee helpen. Ik raad je aan om aangifte hiervan te doen via de politie. Geef dan duidelijk aan dat jij de verkoper bent en dat de koper niet betaald heeft. Hopelijk kunnen zij je hiermee helpen.

Generative Reader's Answer is:
Dat is balen zeg! Helaas kan ik je hier niet mee helpen. Ik raad je aan om aangifte hiervan te doen via de politie. Geef dan duidelijk aan dat jij de verkoper bent en dat de koper

Current Question is:
Hello, I ordered shoes via **marktplaats** and paid with a little bit, now that person would send me the track and trace but got an invalid track and trace now the person says it was a temporary one and has a new one but doesn't leave anything behind hear **can i get my money back?**

Real Answer is:
That does not sound well. Just like with a normal transfer, **it is not possible to refund the money via Tikkie**. If it concerns **fraud**, I advise you to **report it to the police**. In that case, we would also like to receive more information, such as a screenshot of the debit.

Retrieved Top-1 Answer is:
That sucks! Unfortunately I cannot help you with this. **I advise you to report this to the police**. Then clearly indicate that you are the seller and that the buyer has not paid. Hopefully they can help you with this.

Generative Reader's Answer is:
That sucks! Unfortunately I cannot help you with this**. I advise you to report this to the police.** Then clearly indicate that you are the seller and that the buyer

Figure 10: Dutch example 3