Generating domain-specific paraphrases of questions from FAQ

Ng, Jing Rui

2021

Ng, J. R. (2021). Generating domain-specific paraphrases of questions from FAQ. Final Year Project (FYP), Nanyang Technological University, Singapore.
https://hdl.handle.net/10356/148155

https://hdl.handle.net/10356/148155

Generating Domain-Specific Paraphrases of Questions from FAQ

Supervisor: Assoc. Prof Chng Eng Siong

Examiner:

Submitted in Partial Fulfilment of the Requirements
for the Degree of Bachelor of Computer Science and
Engineering of the Nanyang Technological University

by

Ng Jing Rui
U1722094J

2021

ABSTRACT

This project will introduce a paraphrase generation system that will generate domain-specific paraphrases of the questions of Frequently Asked Question (FAQ) corpuses. This project aims to minimise the cost associated with the manual generation of paraphrases and performs effective data augmentation to complement end-to-end FAQ retrieval that uses large language models by reducing overfitting. This is achieved by paying attention to several unique characteristics of the FAQ corpuses and through the use of two large language models, an off-domain labelled paraphrase dataset and abbreviations handling. The two language models used are T5 and Sentence Transformer. The approach proposed involves pre-processing, paraphrase generation, post-processing and candidate paraphrase selection. Firstly, T5 is used to fine-tune on the paraphrase dataset for the task of paraphrase generation. Secondly, abbreviations handling was incorporated into the pre-processing of the original question and post-processing of the generated paraphrase. Thirdly, Sentence Transformer Library is used for candidate paraphrase selection to ensure the semantic similarity of the paraphrase with the original question and the integrity of the paraphrase's class label. Lastly, a GUI application is provided for users to generate paraphrases of questions from a FAQ dataset. From our experiments, we conclude that the pre-processing, post-processing and candidate paraphrase selection are effective in the successful generation of paraphrases and subsequent filtering of these paraphrases to output a set of high-quality domain-specific paraphrases for augmenting the FAQ corpuses.

ACKNOWLEDGEMENTS

This project is only made possible with the help of the following people and I would like to express my utmost gratitude to them:

Firstly, I would like to express my sincere thanks and gratitude to Assoc. Professor Chng Eng Siong for his patience and guidance throughout the duration of this project amidst his busy schedule. His advice and support have been vital to the success of the project and I am truly appreciative to have been given this opportunity to explore and study this domain of Paraphrase Generation and FAQ Retrieval under him. Prof. Chng also engaged his team to have weekly meetings with me to share my progress, discuss further improvements and offer pointers to ensure the quality of the project.

Secondly, I would like to thank members of the team, specifically Damien Cheng, Thi Ly and Zin Tun for providing their valuable feedback and keeping us connected to the rest of the team and Prof. Chng.

Thirdly, I would like to express my gratitude to my family and friends who have being my strong pillars of support throughout the project.

Last but not least, I would like to thank Nanyang Technological University, School of Computer Science and Engineering for providing me with this opportunity to research on this project.

Table of Contents

Chapter 1: Introduction

1.1 Background

Frequently Asked Question (FAQ) pages are typically set up by businesses and organizations on their website as a source of information for the visitors. To provide a better user experience, FAQ retrieval system is often deployed on websites as a chatbot to answer user queries and provide them with the most relevant answer from the FAQ database. Formally, FAQ retrieval can be defined as a task of determining the FAQ pair $\{(q, a)\}$ that best matches the user query Q [1]. Matching of the user query Q can be performed against the question field q, answer field a or the concatenated question-answer field $q + a$ [2].

With reference to Figure 1.1.1, traditional approaches of FAQ Retrieval in chatbot involved the use of Artificial Intelligence Markup Language (AIML) [3] to recognize user's query via pattern matching. While such chatbots [4], [5] are able to retrieve the FAQ, the need to manually define the patterns by experienced personnel meant that expensive manual work had to be put in to set up the FAQ retrieval system. This presented an implementation hurdle for organizations setting up their FAQ retrieval system.

With reference to Figure 1.1.1, several recent works on FAQ retrieval have explored the use of neural network model such as BERT [6] to evaluate the similarity between the user query Q and the fields of the FAQ pairs [1], [7], thereby understanding which FAQ pair best matches the given user query. In contrast to the aforementioned traditional use of AIML, this neural network approach eliminates the need to manually define patterns for recognizing the user query. This translates to greater ease and economic savings on manual work for organizations when they set up a chatbot for the FAQ section of their websites.

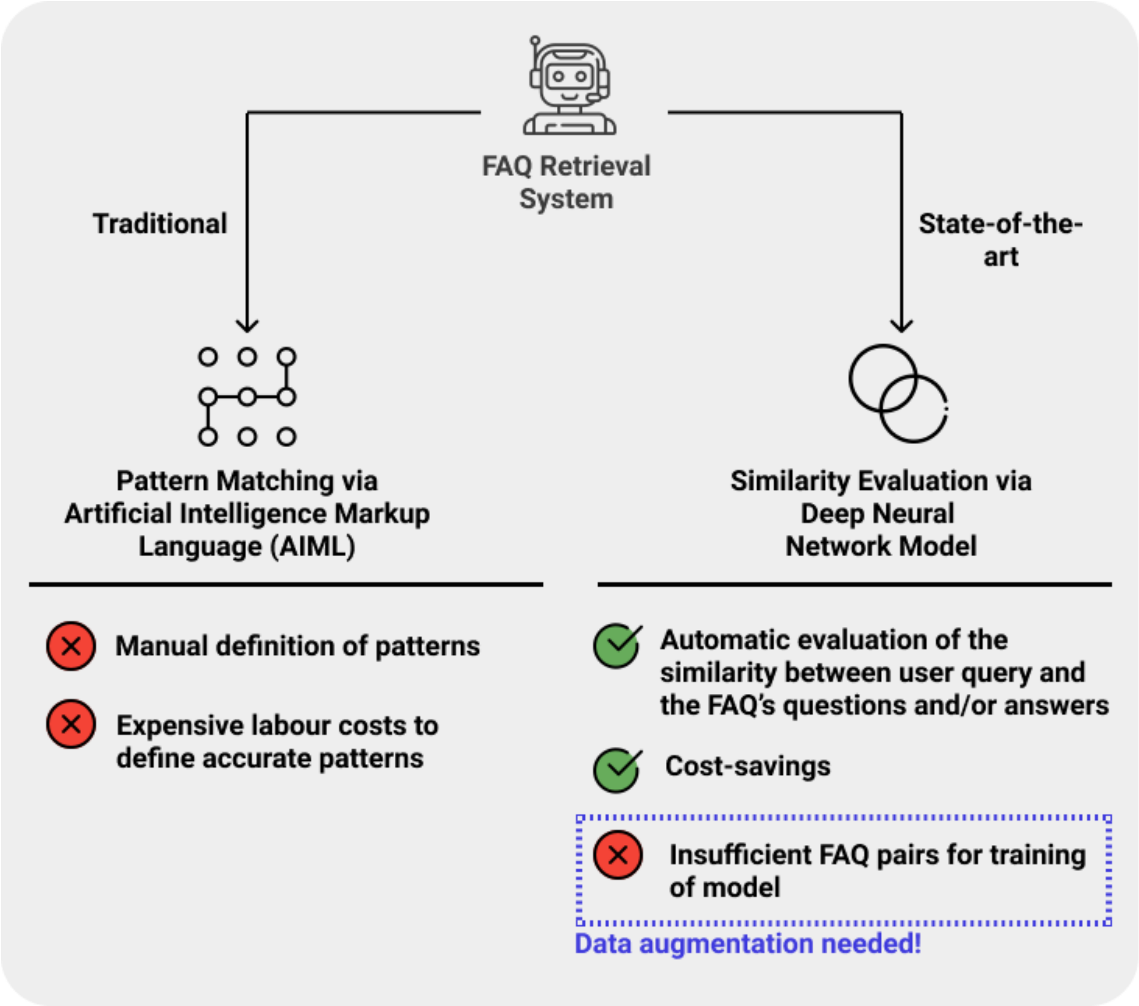

FAQ Retrieval System
Traditional
State-of-the-art
Pattern Matching via Artificial Intelligence Markup Language (AIML)
Similarity Evaluation via Deep Neural Network Model
Manual definition of patterns
Expensive labour costs to define accurate patterns
Automatic evaluation of the similarity between user query and the FAQ's questions and/or answers
Cost-savings
Insufficient FAQ pairs for training of model
Data augmentation needed!

Figure 1.1.1 Compare Traditional and State-of-the-art Approaches of FAQ Retrieval

However, as noted in [7] and shown in Figure 1.1.1, there are insufficient FAQ pairs in the FAQ page for training a neural model to be robust and accurate. This means that we need data augmentation to enable the neural model to be more generalized and not overfit. Furthermore, a fixed one-to-one mapping of question to answer typically exists within the FAQ page. This further prevents the model from training using triplets as performed in [1]. While we can augment the FAQ dataset by generating question paraphrases manually [8], this manual approach invokes annotation costs and require domain expertise while reintroducing the need for manual work that is undesirable. Therefore, in this thesis, we propose a paraphrase generation system that uses a large pre-trained neural model fine-tuned on an off-domain labelled paraphrase dataset via Transfer Learning to automatically generate paraphrases of the questions from FAQ datasets. The system will also incorporate a series of important steps to ensure that paraphrases preserve the domain-specific terms of the original question and to retain paraphrases through two reliable quantitative methods. The retained paraphrases are considered candidate paraphrases and are subsequently used to augment the FAQ dataset for model training. An overview of the workflow supported by this thesis is shown in Figure 1.1.2 below.

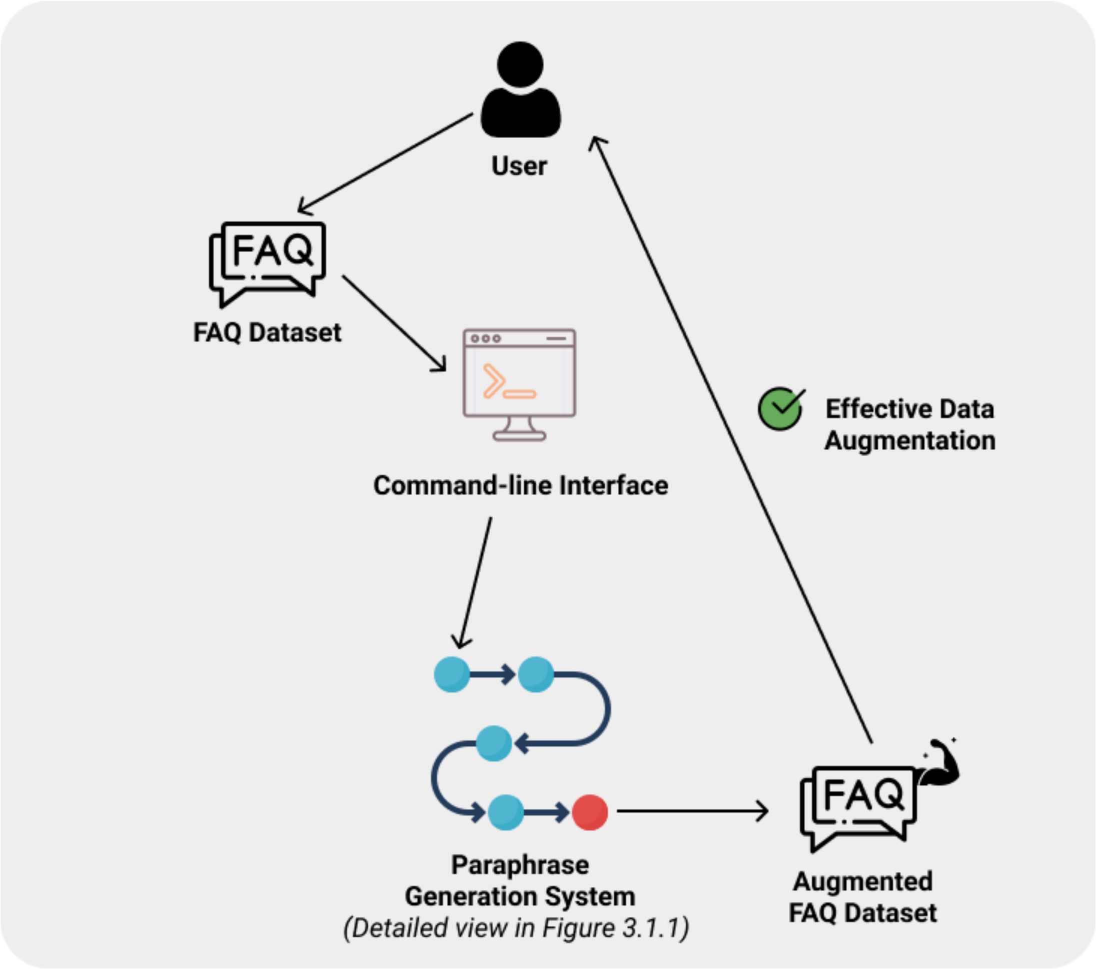

User
FAQ
FAQ Dataset
Command-line Interface
Effective Data Augmentation
Paraphrase Generation System
(Detailed view in Figure 3.1.1)
FAQ
Augmented FAQ Dataset

Figure 1.1.2 Overview of the Workflow Supported by this Thesis

Paraphrase generation refers to a task that outputs a sentence with a preserved semantic meaning while having a different syntax and/or lexical structure from the input sentence [9]. Paraphrase generation has been a long-standing Natural Language Processing (NLP) research area due to its prominent usage for data augmentation. Data augmentation via paraphrasing has shown to improve performance of downstream NLP tasks such as Machine Translation [10], [11], Question Answering [12], [13] and FAQ Retrieval [1], [7].

The early works of paraphrase generation were performed using rules definition [14], statistic machine translation [15] or on the linguistic structures of the input sentences [16]. Paraphrases can also be generated by simple replacement of synonyms [17], combining synonym replacements with other textual manipulations [18], or back translation [19] using neural network. Other approaches involves the use of pre-trained word embeddings or contextualized word embeddings [20] to find similar words to replace a target word. A more recent approach of paraphrase generation is to use large pre-trained model such as GPT-2 [21] in a supervised [22] or unsupervised setting [23]. Our approach while containing similarities to the use of large pre-train model in a supervised setting, differs in multiple aspects. This include ensuring that paraphrases are domain-specific and generating paraphrases of questions instead of generic sentences.

To evaluate the paraphrases generated by the above methods, the paraphrases can be evaluated through varying combinations of various popular metrics such as ROUGE-L [24], METEOR [25], BLEU [26], self-BLEU and cosine similarity between the paraphrase and the original question.

1.2 Objectives

The primary focus of this project is to implement a system that generates paraphrases of the question field q of the FAQ pairs $\{(q, a)\}$ to create domain-specific parallel data for augmenting FAQ dataset used by neural network model for training.

In this project, the following 4 objectives will be achieved:

1. To fine-tune the T5 model on the task of Paraphrase Generation using an off-domain labelled paraphrase dataset
2. To generate domain-specific paraphrases using the fine-tuned T5 model
3. To validate/filter paraphrases for shortlisting eligible candidate paraphrases
4. To provide a command-line interface for users to generate paraphrases of all questions in a given FAQ dataset shown in Figure 1.1.2 above.

By achieving the above insights, we will be able to effectively generate high-quality in-domain paraphrases in an autonomous manner, thereby eliminating the cost and need for domain-specific expertise to manually generate paraphrases. Furthermore, the proposed autonomous question generation system can also be coupled with the use of neural network model for FAQ retrieval to set up an end-to-end FAQ retrieval system using only the FAQ dataset, thereby lowering the implementation barrier to set up their FAQ retrieval system in the form of chatbot.

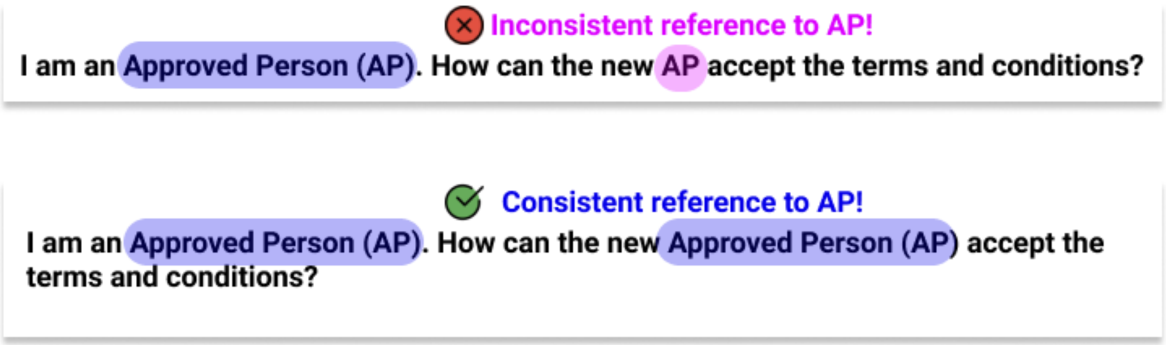
1.3 Scope and Assumptions

The scope of this thesis is restricted to the Ministry of Social and Family Development (MSF)'s BabyBonus FAQ. It is also expected that there is only one possible answer, instead of multiple possible answers, for every user query. While the related process and methods are tested only on MSF's BabyBonus FAQ data, the proposed techniques can be used on other FAQ data, when applicable.

The assumptions of this thesis are:

1. Domain-specific terms are typically made up of an abbreviation and/or its expansion. An example of domain-specific term is Approved Institution (AI) where the abbreviation expansion is "Approved Institution" and the abbreviation is "AI". Therefore, a generated paraphrase is considered domain-specific if it is able to retain the abbreviations and/or the expansions from the original question.

2. We also assume that references of domain-specific terms, if any, are consistent throughout the sentence as shown in Figure 1.3.1. For instance, “I am an Approved Person (AP). How can the new AP accept the terms and conditions?” is not valid as “Approved Person (AP)” and “AP” are used interchangeably.



☒ Inconsistent reference to AP!

I am an Approved Person (AP). How can the new AP accept the terms and conditions?

✓ Consistent reference to AP!

I am an Approved Person (AP). How can the new Approved Person (AP) accept the terms and conditions?

Figure 1.3.1 Inconsistent vs Consistent Reference of Domain-specific Terms

1.4 Report Organisation

This report is organized into five chapters as follow:

- Chapter 1 introduces the project and defines the objectives, scope and assumptions of the project.
- Chapter 2 provides literature review on FAQ Retrievals with Paraphrases, Paraphrase Generation and finally, the grammatical structure of the FAQ questions.
- Chapter 3 introduces the methodology and system architecture of the whole system proposed in this thesis.
- Chapter 4 highlights the experiments and results of the system and supports the design of the system architecture introduced in Chapter 3.
- Chapter 5 provides a conclusion to the report and discusses the future works.

Chapter 2: Literature Review

In this chapter, we performed a deeper comparison of my thesis with the recent works that relates to (1) FAQ Retrieval with Paraphrases and (2) Paraphrase Generation. For (1), we started from looking at how paraphrases have been used to support downstream NLP task of FAQ retrieval. For (2), we started by inspecting the various approaches of performing paraphrase generation. At the end of (1) and (2), we hope to highlight the various aspects that these recent works correlate or differs from my thesis. Finally, we will be inspecting the grammatical structure of the questions of a FAQ dataset.

2.1 FAQ Retrieval with Paraphrases

2.1.1 Automatically Generated Paraphrases

[1] demonstrated an unsupervised approach of FAQ retrieval that retrieves the FAQ pair $\{(q, a)\}$ for a given user query Q through an elaborate process of initial retrieval of top-k FAQ pairs followed by three consecutive re-ranking of the FAQ pairs where the last two re-ranking was carried out by two BERT model. Within the top-k FAQ pairs $\{(q, a)\}$, the first model *BERT-Q-a* measures the similarity of user query Q with the answer field a while the second model *BERT-Q-q* measures the similarity of the user query Q with the question field q. The similarity values are then used as a quantitative measure for re-ranking the FAQ pairs. Both BERT models are fine-tuned previously with the use of triplets. *BERT-Q-a* used triplets (q, a, a') while *BERT-Q-q* used triplets (p, q, q') where a' and q' are a negatively sampled answer and question field respectively while p is the generated paraphrase of q. Paraphrases are first generated by a fine-tuned GPT-2 model followed by filtering and ranking by their similarity to the input question q to give rise to paraphrase p used in the aforementioned triplets. The results of this study align with an earlier study [2] that matching the user query Q and question field of the FAQ pairs $\{(q, a)\}$ helped the most in fetching the correct FAQ pair, compared to other forms of matching. Therefore, this supports our project's focus on generating paraphrases of the questions, rather than the answers.

2.1.2 Manually Generated Paraphrases

[7] demonstrated an FAQ Retrieval approach that used TSUBAKI[27] to determine similarity between user query Q and the question field q as well as a BERT model, trained on classification task, to determine relevance between user query Q and the answer field a. TSUBAKI is a technique used for retrieval by matching words to their synonyms and considering the dependency structure of the sentence. This thesis differs from our project as paraphrases used in this thesis are not automatically generated and no processing or evaluation of these paraphrases was carried out.

2.2 Paraphrase Generation

2.2.1 Back Translation Approach

2.2.1.1 Statistical

Statistical Machine Translation (SMT) leverages on statistical analysis and prediction-based algorithms to define the rules most appropriate for the translation of a given sentence. SMT was first introduced as early as 1999 [28] and can be tackled through a variety of methods based on words [29],phrases [30], syntax [31] or hierarchical phrases [32] that considers both syntax and phrases. The most popular method is the phrase-based SMT. In the work of [15], a phrase-based SMT system was used to generate the best paraphrase T^* for a given sentence S based on probabilities by referring to parallel monolingual datasets:

$$T^* = \arg\max_T (P(T|S))$$

$$T^* = \arg\max_T (P(S|T)P(T))$$

However, as parallel monolingual datasets were rare and bilingual parallel datasets were more readily available, SMT eventually used bilingual corpora to generate paraphrases by performing back translation where a given sentence was translated from English to a pivot language and back to English [33]. SMT eventually starts to get replaced by Neural Machine Translation that provides better accuracy and speed, along with other numerous advantages.

2.2.1.2 Neural Network

While Neural Machine Translation (NMT) was introduced in 2013 [34], NMT systems then had lower accuracy and speed compared to SMT systems. The performance gap of NMT systems then was due to the slower training on large-scale datasets, slower inference time due to large number of parameters involved and the inability to deal with rare words. These issues were eventually addressed by Google in 2016 [35], thereby bringing forth the era where NMT systems outperforms SMT systems in terms of translation quality and speed. Furthermore, NMT was also able to learn the linguistic rules on its own in an end-to-end manner unlike SMT.

The subsequent work of [19] introduced the use of NMT systems to perform back translation for data augmentation for the downstream task of Question Answering by using two separate NMT models. Both NMT models are trained on the same dataset and hyperparameters but with different numbers of steps. In this case, the pivot language used is French, but can be in any other languages as well. The beam decoder of the first NMT model was used to obtain k French translations of a given English sentence. Each of these k French translations was then passed to the beam decoder of the second NMT model for a reverse translation back to English, giving rise to a total of k^2 paraphrases of the given sentence. While NMT-based Back Translation can be an effective form of data augmentation, NMT was found to perform badly for out-of-domain translations [36] where accessible large-scale training data are mostly out-of-domain. This is because in different domains, the same word can be expressed differently and with different meaning. This suggests that neural machine back translation is unlikely to be useful for generating paraphrases that remains domain relevant.

2.2.2 Word Replacement Approach

2.2.2.1 Easy Data Augmentation

Easy Data Augmentation (EDA) is a simple and effective data augmentation technique for text data. It was proposed by Wei and Zou [18].For a given sentence, one of the four main techniques described below are randomly performed:

1. **Synonym Replacement (SR):** Pick out n words at random from the sentence that are not stop words and replace each of these words with a random synonym.
2. **Random Insertion (RI):** For a random word in the sentence that is not a stop word, find a random synonym and insert that synonym randomly into the sentence. Repeat this n times.

3. **Random Swap (RS):** Swap the positions of two words in the sentence randomly. Repeat this n times.
4. **Random Deletion (RD):** At probability p, remove each word in the sentence at random.

Despite the simplicity of these techniques, EDA had shown great results in generating permutations of a given sentence and thereby reducing overfitting of models. However, the sentences generated by EDA are often not semantically or grammatically correct, thereby introducing unnecessary noise. Furthermore, the authors also noted that they expects EDA to provide negligible improvements where pre-trained models such as BERT are used [18].

2.2.2.2 Thesaurus

The work of [16] proposed a lexical substitution system that outputs a ranked list of synonyms for a given target word and its context. This system carries out a two-step process. The first step involves extracting synonyms of the target word from numerous thesaurus sources. Some of the thesaurus sources used are WordNet, Microsoft Encarta Encyclopaedia, Roget and a synonym dataset built from bilingual dictionaries. The second step performs a weighted combination of 7 ranking methods on the synonyms from the earlier step, and output the final list of synonyms ordered by their semantic closeness to both the input word and its context. Some of the ranking methods used are as follow:

1. **Lexical Baseline (LB):** This method ranks the synonyms higher if they are found in both Wordnet and Encarta.
2. **Machine Translation (MT):** This method translates the English sentence to a second language and back to English. Synonyms are ranked higher if they are located in the resulting sentence. Some of the second language tested are French, Italian, Spanish Chinese and German.
3. **Word Sense Disambiguation (WSD):** This method extends previous work [37] which observed that disambiguating the target word helped with the task of lexical substitution .

As compared to EDA, this work provides a more sophisticated approach of finding the best synonym to replace a target word. However, the words of domain-specific terms may still get replaced by their synonyms, which results in paraphrases that are no longer domain-specific. This goes against the focus of our work to generate domain-specific paraphrases.

2.2.2.3 Pretrained Word Embeddings

Word embeddings is a representation of text where each words are mapped to a predefined vector space as a real-valued vector. It can be broadly divided into pretrained and contextualized word embeddings.

word2vec [38] and GloVE [37] are the two earliest methods of constructing pretrained word embeddings of text where the latter is an extension of the former. These two methods can be represented by a function that maps elements v in a word of vocabulary V to real-valued vectors, $h \epsilon \mathbb{R}^d$:

$$f_{vocab}: v \rightarrow h$$

A later method of constructing word embeddings is FastText [39], that extends both word2vec and GloVE by including the consideration of the character sequence $(c_1, \ldots, c_t)$ of the word:

$$f_{subword}: (v, (c_1, \ldots, c_t)) \rightarrow h$$

By using word2vec, GloVE or FastText, words with identical or similar meanings will be mapped close to each other in the vector space. By leveraging on this feature, the pretrained word embeddings of a given word can be used to find the most similar group of words to replace the given word. The most similar group of words have the shortest distance from the given word in the vector space. For example, the word “happy” are likely to be mapped close to its synonyms such as “glad”, “joyous” and “delighted” in the vector space, thereby enabling the replacement of “happy” using one of these synonyms. This approach is encapsulated in the Textual Augmentation feature of a popular NLP Augmentation python library known as nlpaug [40]

However, pretrained word embeddings fail to capture polysemy where words have different meaning in different contexts. This means that the same word will have the same vector representation, regardless of the context the word is in. For example, the word “*solution*” will have the same vector representation for two given sentence “Work out the *solution* in your head” and “Heat the *solution* to 75 degree Celsius". Therefore, replacement of words using their pretrained word embeddings is not context accurate and may generate paraphrases that are no longer relevant to the domain or are semantically different from their original sentence.

2.2.2.4 Contextualized Word Embeddings

In comparison with pretrained word embeddings, contextualized word embeddings are able to capture the polysemous nature of words by taking into account the target word and the words surrounding the target word which forms the context. Contextualized word embeddings can be expressed as a function on the whole text where each word in the sequence is assigned a vector representation:

$$f_{contextual}: (w_1, \ldots, w_N) \rightarrow (h_1, \ldots, h_N)$$

BERT [6] and GPT-2 [21] are some of the popular methods of constructing contextualized word embeddings of a given text. The approach of using contextualized word embeddings to perform word replacement is also encapsulated in the nlpaug library mentioned in Section 2.2.3.3.

By taking into account the words surrounding the target word, the candidate words for replacing the target words can now go beyond synonyms of the target word to include non-synonyms while preserving the semantic meaning of the whole sentence as demonstrated in the work of [20]. This approach also addresses the issue where only a small percentage of words in the vocabulary have identical or similar meanings to each other by considering non-synonyms for a given context. Therefore, paraphrases can be generated more reliably by performing context-aware replacements of a target word using contextualized word embeddings. However, in our thesis, we aim to generate domain-specific paraphrases, thereby making this approach unsuitable due to the issue where specific words of domain-specific terms may get replaced remains unresolved.

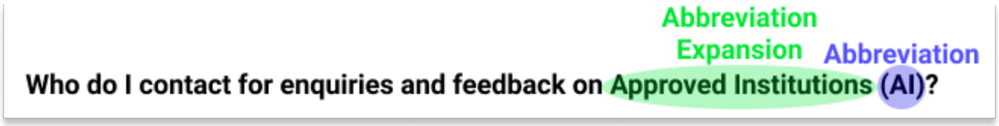
2.2.3 Using Pre-trained Language Models

In recent years, paraphrase generation has been tackled through the use of large language models such as GPT-2. In the work demonstrated in [22], GPT-2 was used in a supervised manner by fine-tuning the multiple GPT-2 model on each of the three labelled paraphrase dataset. The fine-tuned GPT-2 model can then dynamically generate paraphrases for a given sentence. The sentence is also known as a conditional input which is formatted with a unique token “>>>>” following after the sentence. Paraphrases generated are filtered in a two-step process. In the first step, Universal Sentence Encoder [41] is used to obtain the sentence embeddings of the paraphrase and its original sentence to compute the cosine similarity score between these two embeddings. Only paraphrases with a cosine similarity score of between 0.85 and 1 are accepted through the first step. The cosine similarity score ranges from 0 to 1.0 inclusive. In the second step, paraphrases with a ROUGE-L score of above 0.7 are eliminated. By filtering the paraphrases through this two-step process, candidate paraphrases are obtained. This work closely relates to our project as we are using a large pre-trained model in a supervised setting and filtering paraphrases using the cosine similarity score the first step. In our case, we are using the T5 model which will be introduced in Chapter 3.3. We differ from this work by focusing on generating paraphrases for questions which have inherently different grammatical structure from a sentence and tackling domain shift that was not addressed in this work. Furthermore, we filtered our paraphrases differently in the second step.

In another work demonstrated in [23], GPT-2 was used in an unsupervised manner for paraphrase generation but was fine-tuned for the task of reconstructing sentences. The data for the fine-tuning was prepared by taking sentences from a dataset and corrupting each sentence by removing stop words, shuffling the remaining words with a probability of 0.2 and replacing 20% of the words with synonyms from syn-net. Every training example was passed in the format of “corrupted sentence [SEP] original sentence”. Subsequently, paraphrases can then be generated by corrupting the input sentence in the same way and passing the input in the format of “corrupted sentence [SEP]” to the fine-tuned GPT-2 model. Candidate paraphrases in this work are obtained by only retaining paraphrases with a cosine similarity score of above 0.75 between the paraphrase’s embedding and the original sentence’s embedding. The cosine similarity score also ranges from 0 to 1 inclusive. The Sentence Transformer Library [42] was used to generate the aforementioned embeddings. The candidate paraphrases are then subjected to evaluation using human evaluation and metrics such as ROUGE-L, METEOR, BLEU and self-BLEU, but not used as a form of further filtering. This work differs from our project as we intend to use a large pre-trained model in a supervised setting but coincides with our project’s objectives in the aspect of generating domain-specific paraphrases. This work also fails to consider that there are likely to be similar sentences in the original dataset, and therefore filtering of paraphrases solely by their semantic score is unlikely to be sufficient.

2.3 Grammatical Structure of FAQ’s Question Field

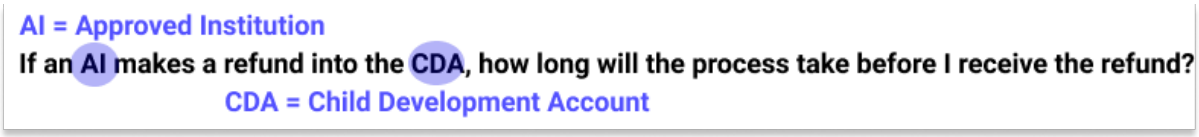
I will be using the BabyBonus FAQ dataset as an example FAQ dataset. The question field of the FAQ pairs usually contains domain-specific terms that are expressed in the form of abbreviations and its expansion. An example can be observed in Figure 2.3.1 below.



Who do I contact for enquiries and feedback on Approved Institutions (AI)?

Figure 2.3.1 Domain-specific Terms

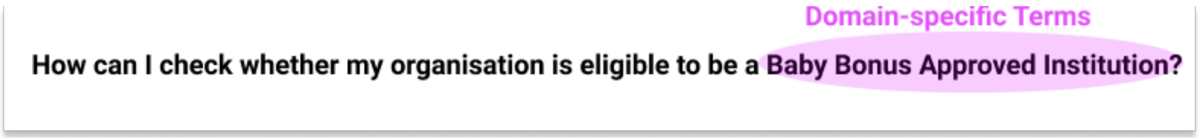
It is also possible for the question field of the FAQ pairs to contain only the abbreviation. An example can be observed in Figure 2.3.2 below.



If an AI makes a refund into the CDA, how long will the process take before I receive the refund?

Figure 2.3.2 Abbreviations only

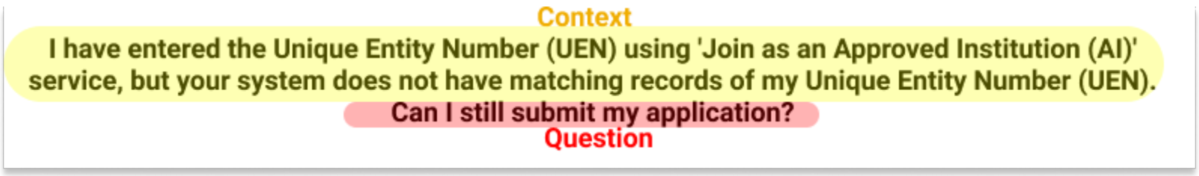
However, domain-specific terms may not be expressed in abbreviations but instead capitalized as shown in Figure 2.3.3 below.



How can I check whether my organisation is eligible to be a Baby Bonus Approved Institution?

Figure 2.3.2 Capitalized Domain-specific Terms

While the question field of most FAQ pairs contain only the questions, some question field contains both context and question as shown in Figure 2.3.4 below.



I have entered the Unique Entity Number (UEN) using 'Join as an Approved Institution (AI)' service, but your system does not have matching records of my Unique Entity Number (UEN). Can I still submit my application?

Figure 2.3.2 Context followed by Question

Chapter 3: Methodology & System Architecture

3.1 System Architecture

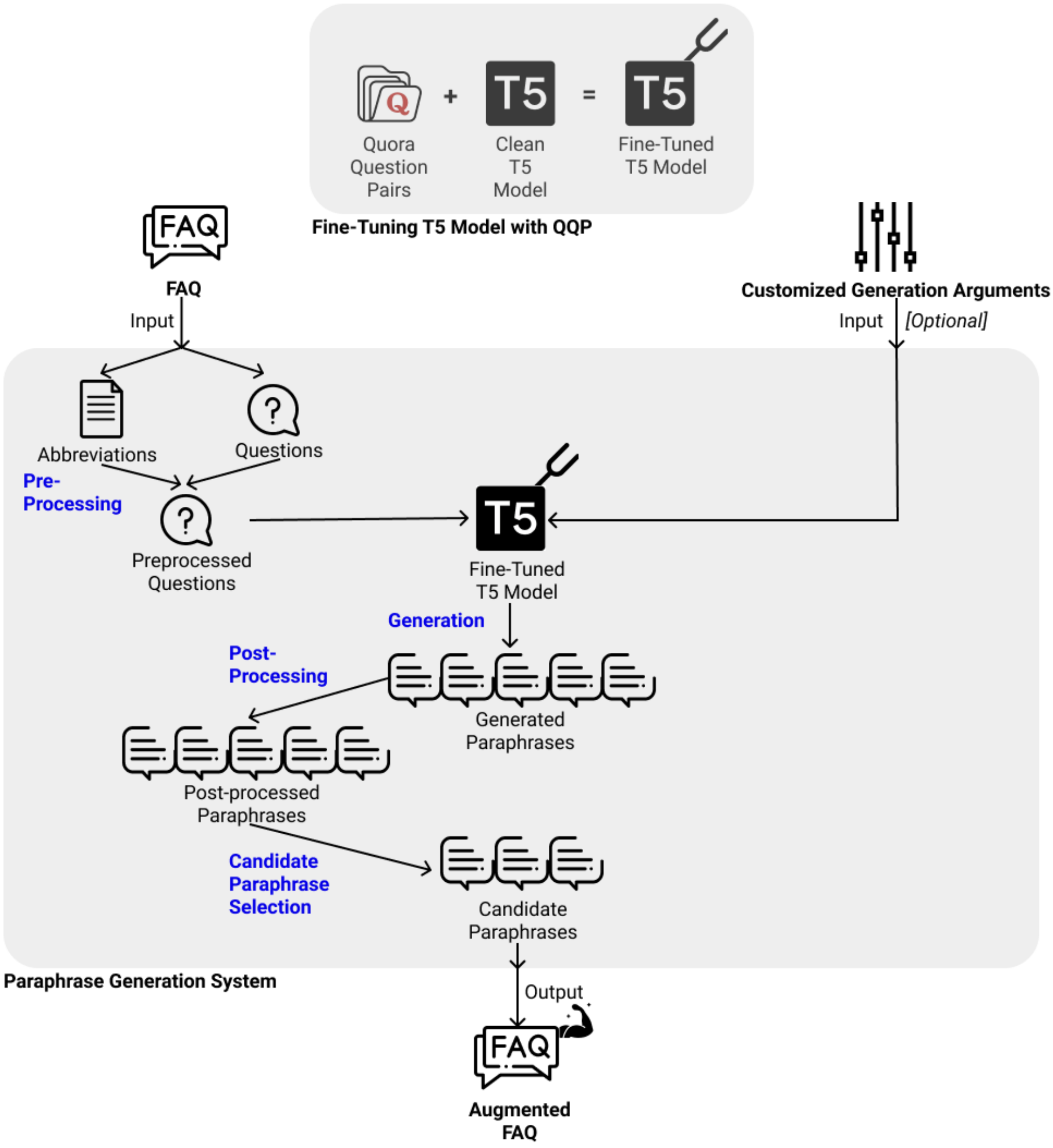


Figure 3.1.1 System Architecture

The system architecture of the project is shown in Figure 3.1.1. The system is designed in a manner such that the user only needs to upload the FAQ dataset to obtain an augmented dataset as the system output. The user may choose to pass in specific generation arguments. If no generation arguments are passed in, their corresponding default values will be used.

Various parts of the system architecture will be examined in greater details in the subsequent sections of Chapter 3.

3.2 Datasets

3.2.1 FAQ Dataset

In this thesis, we are using the BabyBonus FAQ extracted manually from the Ministry of Social Service (MSF) website. The BabyBonus FAQ contains around 290 FAQ pairs $\{(q, a)\}$ where each question q is mapped to a single answer a only. Each question q will be subsequently used to generate paraphrases p. Candidate paraphrases will then be paired with the answer of their original question q, thereby forming new FAQ pairs. Suppose there are n original FAQ pairs, and a total of m candidate paraphrases, then there will be $(n + m)$ FAQ pairs with n answers and $(n + m)$ questions.

3.2.2 Quora Question Pair

Quora Question Pair (QQP) [43] is a labelled paraphrase dataset that was released by Quora in 2017. It contains around 400k question pairs, each with a binary flag. Each question pairs consist of *sentence1* and *sentence2*. If the binary flag is 1, the question pairs are paraphrases of each other, otherwise the question pairs are duplicates of each other. To prepare the dataset for fine-tuning the T5 model, question pairs with a binary flag value of 1 are retained. This gives us about 140K paraphrase pairs.

Quora Question Pair is an appropriate dataset for fine-tuning the T5 model as it aligns with our project's focus of generating paraphrases of questions. Due to the aforementioned project focus, we did not consider the other larger labelled paraphrase dataset available publicly.

3.3 Large Language Models

In our study, instead of using GPT-2 model, we will be using the T5 model [44] that was pre-trained on Colossal Clean Crawled Corpus (C4) and have achieved cutting-edge results on multiple NLP tasks while enabling fine-tuning for different downstream tasks. "t5-base" model will be loaded twice as the model and tokenizer respectively. The model will then be fine-tuned on the 140k paraphrase pairs of the QQP.

3.4 Fine-Tuning of T5

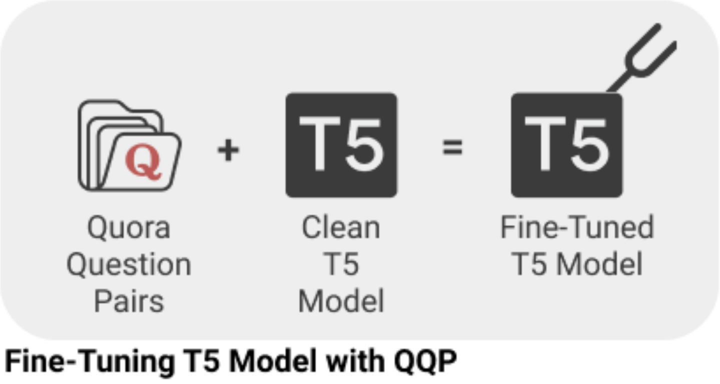


Figure 3.4.1 Fine-Tuning T5 Model with QQP

Figure 3.4.1 shows the overall structure of fine-tuning T5 with QQP. Each paraphrase pair consists of *sentence1* and *sentence2* which are the input and target respectively. To prepare the training examples, each paraphrase pair is formatted as follow:

Input: “paraphrase: *sentence1* </s>”

Target: “*sentence2* </s>”

where “paraphrase:” is used as the task prefix of paraphrase generation while “</s>” tag is added at the end of the input and target.

The fine-tuning of the T5 model are carried out using the following hyperparameters’ values, similar to that of [45], as shown in Figure 3.4.2 below.

Hyperparameter Name	**Value**
Max Sequence Length	512
Learning Rate	3e-4
Weight Decay	0.0
Adam Epsilon	1e-8
Warmup Steps	0
Train Batch Size	6
Evaluation Batch Size	6
Number of Epochs	2
Gradient Accumulation Steps	16
Number of GPU	1
Early Stop Callback	False

FP_16	False
Opt Level	01
Maximum Gradient Norm	1.0
Seed	42

Figure 3.4.2 Fine-Tuning Hyperparameters Values

3.5 Generation Configuration

With reference to Figure 3.5.1 below, the left column contains the names of the arguments for the generation while the right column contains the default values of these arguments. Users may choose to pass in their customised values of these arguments, which will alter the generation results accordingly.

Argument Name	**Value**
top_k	120
top_p	0.98
max_len	256
num_return	40

Figure 3.5.1 Generation Arguments' Default Values

3.6 Pre-processing of Input Sentence

Every question of the FAQ is seen as an input sentence and pre-processed as described below. In this study, we carry out pre-processing of the input sentence in the form of abbreviations handling where abbreviation expansion + (abbreviation) is replaced with its abbreviation only. An example of the pre-processing can be observed in Figure 3.6.1.

Before pre-processing	I entered the Unique Entity Number (UEN) using 'Join as an Approved Institution (AI)' service, but your system does not match records of my Unique Entity Number (UEN). Can I still submit my application?
Abbreviation pairs present	{"UEN": "Unique Entity Number", "AI": "Approved Institution"} where the key and the value are the abbreviation and its expansion respectively.
After pre-processing	I entered the UEN using 'Join as an AI' service, but your system does not match records of my UEN. Can I still submit my application?

Figure 3.6.1 Pre-processing Example

3.7 Generation of Question Paraphrases

The model will start generating paraphrases for a given conditional input as guided by the key arguments highlighted in Section 3.5. Each conditional input is formatted as follow:

“paraphrase: *pre-processed question* </s>”

The “paraphrase:” is the same task prefix used in the fine-tuning stage, and used as a signal to the model to perform paraphrase generation.

Suppose that no customised key arguments are passed in, then the default values of the arguments listed in Figure 3.5.1 above will be used.

3.8 Post-processing of Paraphrases

Question paraphrases generated are only accepted if they are unique from both the original question and from each other. These paraphrases will then have their abbreviation repopulated with the abbreviation and its expansion from their respective original question. An example is shown in Figure 3.8.1.

Original Question	I have entered the Unique Entity Number (UEN) using 'Join as an Approved Institution (AI)' service, but your system does not have matching records of my Unique Entity Number (UEN). Can I still submit my application?
Paraphrase **before** post-processing	My application to join as an AI doesn’t match my UEN application, can I still apply?
Paraphrase **after** post-processing	My application to join as an Approved Institution (AI) doesn't match my Unique Entity Number (UEN) application, can I still apply?

Figure 3.8.1 Post-processing Example

3.9 Candidate Paraphrase Selection

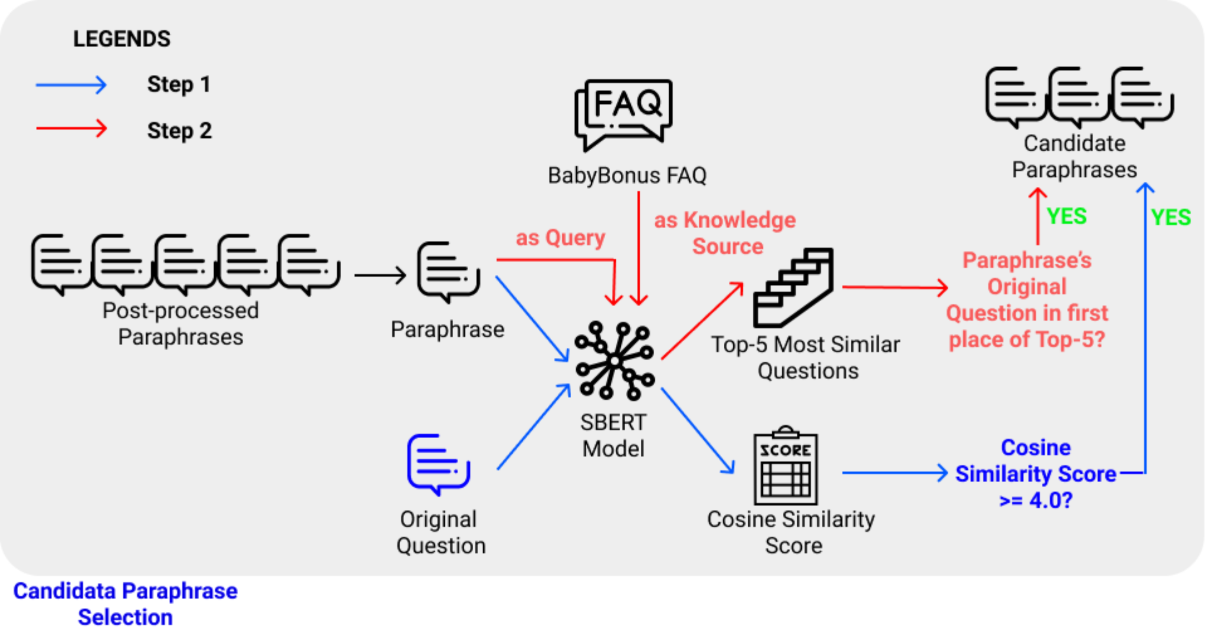

LEGENDS
Step 1
Step 2
BabyBonus FAQ
Candidate Paraphrases
Post-processed Paraphrases
Paraphrase
as Query
as Knowledge Source
YES
YES
Paraphrase's Original Question in first place of Top-5?
Top-5 Most Similar Questions
SBERT Model
SCORE
Cosine Similarity Score
Cosine Similarity Score >= 4.0?
Original Question
Candidata Paraphrase Selection

Figure 3.9.1 Two-Step Selection Process of Candidate Paraphrases

After performing post-processing of the paraphrases, we loaded the Sentence Transformer [42]model, specifically "paraphrase-distilroberta-base-v1", to shortlist high-quality paraphrases in a two-step selection process shown in Figure 3.9.1. The first selection step ensures that shortlisted paraphrases are semantically similar to their original questions and the second selection step retains paraphrases from the previous step that are most semantically similar to their original questions as compared to all other original questions. Shortlisted paraphrases are henceforth referred to as candidate paraphrases.

In the first selection step, the model will evaluate the semantic similarity between the paraphrase and its original question, and allocate a score that ranges from 0 to 5.0 inclusive. Only paraphrases with a semantic similarity score of above 4.0 are accepted and passed to the second selection step. Examples of paraphrases with a semantic score of above 4.0 are shown in Figure 3.9.2.

Original:Paraphrases	Score
Original: When do I need to change the Child Development Account (CDA) trustee?	
Paraphrase:	
1. Will I need to change the trustee of Child Development Account (CDA)?	**4.85**
2. How do I change the trustee on my Child Development Account (CDA)?	**4.43**
3. When is the time to change my Child Development Account (CDA) trustee?	**4.41**
4. Should the Child Development Account (CDA) be changed for a child?	**4.06**
Original: How can I save in the Child Development Account (CDA)?	
Paraphrases:	
1. What should I save for Child Development Accounts (CDA)?	**4.72**
2. How can we save money in the Child Development Account (CDA)?	**4.61**
3. What is the best way to save money in Child Development Account (CDA)?	**4.46**
4. How do I save to child development account (CDA) from the govt?	**4.40**
Original: Where can I use the Child Development Account (CDA) funds?	
Paraphrases:	
1. How should I use Child Development Account (CDA) funds?	**4.88**
2. Where can I use child development account (CDA) funds for youth?	**4.71**
3. What are the purposes of using Child Development Account (CDA) money?	**4.52**
4. Where can I use the Child Development Account (CDA) funds for my school?	**4.21**
Original: I have entered the Unique Entity Number (UEN) using the 'Join as Approved Institution (AI)' service, but your system indicated that my Unique Entity Number (UEN) is invalid, what should I do?	
Paraphrases:	
1. What do I do when I enter the Unique Entity Number (UEN) using the 'Join as Approved Institution (AI)' service, but your system indicated that my Unique Entity Number (UEN) is invalid, what should I do?	**4.77**
	4.50

2. If I have entered a unique entity number with the 'Join as Approved Institution (AI)' and the Unique Entity Number (UEN) is invalid, what should I do? 3. I need to enter my Unique Entity Number (UEN) using your 'Join As Approved Institution (AI)' service, but your system informed that my Unique Entity Number (UEN) is invalid. What should I do?	**4.25**

Figure 3.9.2 Original-Paraphrase with Semantic Score of above 4.0

In the second selection step, the model will use the paraphrase as a query to the corpus of the 290 original questions to retrieve the top-5 original questions most similar to the paraphrase. If the given paraphrase's original question is in the top-1, then the paraphrase will be passed to the subsequent selection step. This selection step is important because in a FAQ dataset, there can be highly similar FAQ question mapping to very different answers, therefore this selection helps to filter out paraphrases that are more similar to other original questions than the original question it was generated from. Furthermore, paraphrase should retain the same class label as the question it was generated from. Examples of similar original questions in the case of the FAQ dataset used are shown in Figure 3.9.3.

Question 1: Question 2	Score
Question 1: When do I need to change the Child Development Account (CDA) trustee? **Question 2**: How can I change the Child Development Account (CDA) trustee?	**4.80**
Question 1: How can I save in the Child Development Account (CDA)? **Question 2**: How much can I save in the Child Development Account (CDA)?	**4.80**
Question 1: Where can I use the Child Development Account (CDA) funds? **Question 2**: How can I make payment using my Child Development Account (CDA) funds?	**4.62**
Question 1: I have entered the Unique Entity Number (UEN) using the 'Join as Approved Institution (AI)' service, but your system indicated that my Unique Entity Number (UEN) is invalid, what should I do? **Question 2**: I have entered the Unique Entity Number (UEN) using 'Join as an Approved Institution (AI)' service, but your system does not have matching records of my Unique Entity Number (UEN). Can I still submit my application?	**4.53**

Figure 3.9.3 Similar Original Questions in the FAQ dataset used

Chapter 4: Experiment and Results

4.1 Paraphrase Generation with Abbreviation Handling

4.1.1 Abbreviation Handling

In our project, we are performing abbreviation handling as our pre-processing. Without abbreviation handling, we noticed that the T5 model often fails to generate for certain paraphrases. It should be noted that the generation of paraphrases of T5 model is random, in the sense that the same paraphrase may appear in the first generation run but missing in the second generation run. This is because the generation works like a black box, where we simply supply the sentence that we wish to generate paraphrases for.

Subsequent experiments revealed that by performing abbreviations handling, the T5 model was able to consistently generate paraphrases for all 290 questions of the FAQ dataset used. Furthermore, performing abbreviation handling makes T5 model less likely in replacing the characters of the abbreviation. This enables us to make use of the retained abbreviation to replace its abbreviation expression, thereby ensuring no words of the abbreviation expression are replaced. Mostly importantly, doing so enabled us to retain domain-specific terms, without the risk of specific words of the domain-specific terms being replaced by their synonyms as in the case of other paraphrase generation approaches.

4.1.2 Discussion of Results

For an example, we have chosen an original question “How can I save in the Child Development Account (CDA)?” from the FAQ dataset used. This question was selected as an example as it is similar to another original question “How much can I save in the Child Development Account (CDA)?” with a semantic similarity score of 4.80 as shown in Figure 3.9.3. The selected question also contains domain-specific terms in the abbreviation format described in Figure 2.3.1 where “Child Development Account” is the abbreviation expansion and “CDA” is the abbreviation. we have manually selected 10 paraphrases generated from the selected question as shown in Figure 4.1.3.1 below.

Original			
How can I save in the Child Development Account (CDA)?			
S/N	**Paraphrases**	**Score**	**Position**
1	What can I save in the Child Development Account (CDA)?	**4.91**	**1**
2	What are the ways to save in the Child Development Account (CDA)?	**4.84**	**1**
3	Can I save to the Child Development Account (CDA)?	**4.79**	**1**
4	What is the best way to save in Child Development Account (CDA)?	**4.77**	**1**
5	How can we save money in the Child Development Account (CDA)?	**4.61**	**2**
6	How do I save money using Child Development Account (CDA)?	**4.51**	**1**
7	What is the best way to save money in your child development account (CDA)?	**4.46**	**2**
8	How can I get Savings in Child Development Account (CDA)?	**4.41**	**1**

9	I have a Child Development Account (CDA). How do I save it?	**4.29**	**1**
10	How can I save money through Child Development Account (CDA)?	**4.22**	**2**

Figure 4.1.3.1 10 Paraphrases of the Selected Question

These 10 paraphrases are evaluated for their semantic similarity score with the selected question, and for the position of the selected question in the top-5 most similar questions of the original FAQ corpus when the paraphrase is used as a query.

From the 10 paraphrases above, we can observe that the paraphrases of above 4.0 score remained on the topic of saving in the Child Development Account. Some of these paraphrases are asking about the said topic in a different way compared to the original question. For example, paraphrases #1, #2 and #4 are the what-type questions in comparison to the how-type of the original question. Paraphrase #3 is another notably different type of question as it starts with “can”, instead of “how” or “what”. The structure of the paraphrase #9 is the same structure described in Figure 2.3.2 where a question follows after a context. This shows that paraphrases of above 4.0 score remains on the topic, while having a different grammatical structure from the original question.

We can also observe that the domain-specific term “Child Development Account” is retained where no parts of the term were being replaced by an alternative word. For example, “Growth” could have been a suitable replacement for “Development”, which would have altered the term to become “Child Growth Account”, which is undesirable. Therefore, we have successfully generated paraphrases that retains their domain-specific terms with the use of the fine-tuned T5 model.

From these paraphrases, we can also observe that the semantic score of the paraphrase with the original question is insufficient on its own to quantify whether the paraphrase should be retained as a candidate paraphrase. Paraphrases #5, #7 and #10 all have semantic score of above 4.0, which meant that they are considered semantically similar to the original question. However, when these paraphrases were submitted as a query to the original FAQ corpus, it was found that the paraphrases are more semantically similar to “How much can I save in the Child Development Account (CDA)?” than “How can I save in the Child Development Account

(CDA)?” that they were generated from. As generated paraphrases should retain the class label of the question it was generated from, paraphrases #5, #7 and #10 will be filtered out in the “Candidate Paraphrase Selection” component of the system as described in Section 3.9. Therefore, we have achieved our project objective of filtering and validating the generated paraphrases to shortlist candidate paraphrases effectively.

Chapter 5: Conclusion and Future Works

5.1 Conclusion

In this thesis, we demonstrated a system designed to automatically generate in-domain paraphrases of the questions in a FAQ dataset through a pipeline of pre-processing, post-processing and candidate paraphrase selection. We investigated the grammatical structure of the FAQ questions, highlighted that the FAQ questions typically contains domain-specific terms and are occasionally semantically similar to each other. We also highlighted that FAQ corpuses are usually small, thereby causing neural network models used for end-to-end FAQ retrieval to overfit. With attention paid to these three highlighted characteristics of the FAQ dataset, the system augments the existing FAQ corpus with validated domain-specific paraphrases of the questions. Through this end-to-end approach of generating parallel domain-specific data, we hope to minimize the cost associated with the otherwise manual generation of paraphrases. We also hope that the generated data can complement the state-of-the-art method of setting up FAQ retrieval end-to-end with neural network models by helping these models generalize better and reduce overfitting with the augmented dataset.

The system involves the use of a T5 model fine-tuned on an off-domain but labelled paraphrase dataset known as Quora Question Pairs to generate the paraphrases, and incorporated the use of abbreviations handling in the pre-processing and post-processing. The generated paraphrases were then filtered through two consecutive selection steps, so that the resulting candidate paraphrases are semantically similar to their original question and hold the same class label as their original question. From the experiments done to inspect the results of the system, we conclude that the paraphrases generated are domain specific, can have a grammatical structure different from their original question, and still be grammatically sound. Furthermore, we also conclude that checking the semantic similarity of the paraphrase with the original question should be complemented with ensuring the integrity of the paraphrase's class label, as the isolated use of the former was shown in some of the earlier works.

5.2 Future Works

The system proposed in this thesis is not perfect and have their own sets of limitation. In this section, we would be making suggestions of future works and further research that researchers interested in the field of Text Augmentation and FAQ Retrieval could explore further into. The following sub-sections of this section will discuss several possible directions that can improve on the existing system.

5.2.1 Large Language Models

In this thesis, we have used a fine-tuned T5 model to generate the paraphrases and a pre-trained Sentence Transformer model to evaluate and filter the paraphrases. The large language models used can have a direct correlation to the quality of the generated paraphrases and the reliability of the candidate paraphrases selection. GPT-2 was known for its outstanding text generation capabilities and was also used in the earlier works mentioned in Section 2.2.3. This means that we can research on whether using GPT-2 in place of T5 will give us higher quality paraphrases. Furthermore, GPT-3 [46], a successor of GPT-2, is also released recently. GPT-3 is currently one of the largest language models with a total of 175 billion parameters, compared to GPT-2's 1.5 billion parameters. As suggested by many sources, the performance of GPT-3 is likely to outperform that of GPT-2, and that of T5. Therefore, with the release of more capable and larger language models in the future, this work can be further improved on with the use of the newer language models such as GPT-3 and further researched on with the use of the other existing language models such as GPT-2.

5.2.2 Named Entity Recognition

Named Entity Recognition (NER) refers to the task of tagging entities in text with their corresponding type. Similar to our abbreviations handling, NER can be used to extract the important entities of the question. By extracting the entities, we can use our understanding of these entities to ensure that no parts of the entity text are being replaced. For example, for the question "How can Approved Institution apply for tax relief from 24 March 2020?", we can identify two entities, namely organizational entity "Approved Institution" and date entity "24 March 2020". No parts of these two entities should be replaced. The organizational entity "Approved Institution" can also be seen as a domain-specific term. Therefore, by recognizing the important entities in the question, we can generate paraphrases that retain more domain-specific terms expressed in a greater variety of formats.

5.2.3 Question Type Recognition

Question Type Recognition can also be introduced so that we can build on the existing candidate paraphrase selection process. We can use question type recognition to identify the question type of the original question. Suppose the original question is a “what” type of question and we wish to retain the 10 best paraphrases, then we can retain a certain number of paraphrases belonging to the “what” type and the remaining number of paraphrases belonging to the other question types. In this way, we can have a greater diversity of the grammatical structure of the paraphrases, which may in turn generalize the augmented FAQs better. Therefore, by incorporating Question Type Recognition, we can improve on our selection of candidate paraphrase to have paraphrases of different grammatical structure.

5.2.4 In-Domain Queries as Dataset

The dataset used to fine-tune the large language model for the task of paraphrase generation can also have a direct influence on the quality and bias of the paraphrase generated. In our thesis, we used the Quora Question Pair dataset. While the Quora Question Pair dataset consists of paraphrased question pairs, these questions are not in the same domain of the FAQ dataset used and are instead gathered from a wide range of topics on Quora. Suppose that actual user queries are stored and mapped to their corresponding target question of the FAQ dataset used, it can improve the fine-tuning of the language model used, which in turns improve on the quality of the paraphrases generated. In 2019, a paper proposed a chatbot that is able to learn from the dialogues it engaged in after deployment [47]. By evaluating the users’ satisfaction and the users’ feedback when they are unsatisfied, the chatbot is able to extract meaningful training examples from the dialogues and improve on its dialogue abilities. With this paper as an example, it suggests that it can be possible that with a little help of the user, chatbots can extract positive training examples from their past dialogues. Extending this paper to ours, we can look into how we can form the mapping of user queries to the users’ intended questions and use this in-domain dataset, in place of the off-domain Quora Question Pairs dataset, to better fine-tune our language model for generating in-domain paraphrases.

Bibliography

[1] Y. Mass, B. Carmeli, H. Roitman, and D. Konopnicki, “Unsupervised FAQ Retrieval with Question Generation and BERT,” 2020, doi: 10.18653/v1/2020.acl-main.74.

[2] M. Karan and J. Šnajder, “FAQIR – a frequently asked questions retrieval test collection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9924 LNCS, 2016.

[3] R. Wallace, “The Elements of AIML Style,” *Alice AI Foundation, Inc*, 2003.

[4] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, “Chatbot for university related FAQs,” in *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, 2017, vol. 2017-January, doi: 10.1109/ICACCI.2017.8126057.

[5] N. T. Thomas, “An e-business chatbot using AIML and LSA,” 2016, doi: 10.1109/ICACCI.2016.7732476.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Oct. 2018, Accessed: Mar. 17, 2021. [Online]. Available: http://arxiv.org/abs/1810.04805.

[7] W. Sakata, R. Tanaka, T. Shibata, and S. Kurohashi, “FAQ retrieval using query-question similarity and BERT-based query-answer relevance,” 2019, doi: 10.1145/3331184.3331326.

[8] M. Karan and J. Šnajder, “Paraphrase-focused learning to rank for domain-specific frequently asked questions retrieval,” *Expert Systems with Applications*, vol. 91, 2018, doi: 10.1016/j.eswa.2017.09.031.

[9] R. Bhagat and E. Hovy, “What is a paraphrase?,” *Computational Linguistics*, vol. 39, no. 3, 2013, doi: 10.1162/COLI_a_00166.

[10] Y. Marton, C. Callison-Burch, and P. Resnik, “Improved Statistical Machine Translation using monolingually-derived paraphrases,” 2009, doi: 10.3115/1699510.1699560.

[11] C. Callison-Burch, P. Koehn, and M. Osborne, “Improved statistical machine translation using paraphrases,” 2006, doi: 10.3115/1220835.1220838.

[12] L. Dong, J. Mallinson, S. Reddy, and M. Lapata, “Learning to paraphrase for question answering,” 2017, doi: 10.18653/v1/d17-1091.

[13] A. Fader, L. Zettlemoyer, and O. Etzioni, “Paraphrase-driven learning for open question answering,” in *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2013, vol. 1.

[14] “Paraphrasing questions using given and new information,” *Paraphrasing questions using given and new information*, vol. 9, no. 1, 1983, doi: 10.1145/965105.807463.

[15] C. Quirk, C. Brockett, and W. B. Dolan, “Monolingual Machine Translation for Paraphrase Generation,” *Emnlp-2004*, no. 2001, 2004.

[16] S. Hassan, A. Csomai, C. Banea, R. Sinha, and R. Mihalcea, “UNT: SubFinder: Combining knowledge sources for automatic lexical substitution,” 2007.

[17] D. Yuret, “KU: Word sense disambiguation by substitution,” 2007.

[18] J. Wei and K. Zou, “EDA: Easy data augmentation techniques for boosting performance on text classification tasks,” 2020, doi: 10.18653/v1/d19-1670.

[19] A. W. Yu *et al.*, “QaNet: Combining local convolution with global self-attention for reading comprehension,” 2018.

[20] S. Kobayashi, “Contextual augmentation: Data augmentation bywords with paradigmatic relations,” in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies - Proceedings of the Conference, 2018, vol. 2, doi: 10.18653/v1/n18-2072.

[21] X. Wu and M. Lode, "Language Models are Unsupervised Multitask Learners (Summarization)," *OpenAI Blog*, vol. 1, no. May, 2020.

[22] S. Witteveen and M. Andrews, "Paraphrasing with Large Language Models," *arXiv*. 2019, doi: 10.18653/v1/d19-5623.

[23] C. Hegde and S. Patil, "UNSUPERVISED PARAPHRASE GENERATION USING PRE-TRAINED LANGUAGE MODELS," *arXiv*. 2020.

[24] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Proceedings of the workshop on text summarization branches out (WAS 2004)*, no. 1, 2004.

[25] A. Lavie and A. Agarwal, "METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, vol. 0, no. June.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," *ACL*, 2001, doi: 10.3115/1073083.1073135.

[27] K. Shinzato, T. Shibata, D. Kawahara, and S. Kurohashi, "TSUBAKI: An open search engine infrastructure for developing information access methodology," *Journal of Information Processing*, vol. 20, no. 1, 2012, doi: 10.2197/ipsjjip.20.216.

[28] F. Och, C. Tillmann, and H. Ney, "Improved Alignment Models for Statistical Machine Translation," 1999.

[29] F. J. Och and H. Ney, "Improved statistical alignment models," 2000, pp. 440–447, doi: 10.3115/1075218.1075274.

[30] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," 2003, doi: 10.3115/1073445.1073462.

[31] M. Galley *et al.*, "Scalable inference and training of context-rich syntactic translation models," in *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2006, vol. 1, doi: 10.3115/1220175.1220296.

[32] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, 2007, doi: 10.1162/coli.2007.33.2.201.

[33] C. Bannard and C. Callison-Burch, "Paraphrasing with bilingual parallel corpora," 2005, doi: 10.3115/1219840.1219914.

[34] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," 2013.

[35] Y. Wu *et al.*, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," Sep. 2016, Accessed: Mar. 15, 2021. [Online]. Available: https://arxiv.org/abs/1609.08144.

[36] P. Koehn and R. Knowles, "Six challenges for neural machine translation," *arXiv*. 2017, doi: 10.18653/v1/w17-3204.

[37] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," 2014, doi: 10.3115/v1/d14-1162.

[38] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," Oct. 2013, Accessed: Mar. 15, 2021. [Online]. Available: https://arxiv.org/abs/1310.4546.

[39] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Jul. 2016, Accessed: Mar. 15, 2021. [Online]. Available: http://arxiv.org/abs/1607.04606.

[40] Edward Ma, "NLP Augmentation," *https://github.com/makcedward/nlpaug*, 2019. .

[41] D. Cer *et al.*, "Universal Sentence Encoder," Mar. 2018, Accessed: Mar. 15, 2021. [Online]. Available: https://arxiv.org/abs/1803.11175.

[42] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," 2020, doi: 10.18653/v1/d19-1410.

[43] "Quora Question Pairs," *https://www.kaggle.com/c/quora-question-pairs/data*, 2017. .

[44] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html.

[45] Ramsri Goutham, "Paraphrase any question with T5 (Text-To-Text Transfer Transformer)," *https://towardsdatascience.com/paraphrase-any-question-with-t5-text-to-text-transfer-transformer-pretrained-model-and-cbb9e35f1555*, May 31, 2020. .

[46] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," May 2020, Accessed: Mar. 17, 2021. [Online]. Available: http://arxiv.org/abs/2005.14165.

[47] B. Hancock, A. Bordes, P.-E. Mazaré, and J. Weston, "Learning from Dialogue after Deployment: Feed Yourself, Chatbot!," Jan. 2019, Accessed: Mar. 17, 2021. [Online]. Available: https://arxiv.org/abs/1901.05415.