Out-of-Domain, Out-of-Sight: Enhancing Query-Based Meeting Summarization Across Domains

Maxime Chemenda

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2023

Abstract

In recent years, the amount of meeting transcripts has been rapidly growing due to the increase in remote meetings and advancements in transcription technology. This increased the need for query-based meeting summarisation models, where the goal is to generate a summary of a meeting based on a query. This allows users to quickly extract specific information from the meeting transcript by asking a query. Despite the increasing use of such models, existing solutions have limited efficacy in generalising to new domains, and require a large amount of domain-specific data to perform well, which can be challenging to obtain due to the time-consuming and expensive process of data collection and manual annotation.

This creates an crucial need for solutions to improve the out-of-domain performance of existing models to perform well on new unseen domains. This thesis proposes a novel approach by using data augmentation techniques to solve the issue of out-of-domain performance in query-based meeting summarisation. By increasing the diversity and size of the training data, data augmentation can improve the ability of the models to generalize to new domains. Achieving this objective has practical implications in improving the accuracy and efficiency of automatically generated meeting summaries, which can save time and resources for individuals and organizations.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Maxime Chemenda*)

Acknowledgements

Firstly, I would like to express my heartfelt gratitude to my supervisor, Alexandra Birch-Mayne, for her invaluable guidance and support throughout this research journey. Her expertise and insights have been instrumental in shaping this thesis.

I would also like to extend my appreciation to the authors who created the datasets and models for providing access to these resources, which were critical to the success of this study.

I am deeply grateful to my family who have constantly supported me throughout my whole life, providing love and support, and to my friends, who brought even more joy throughout my academic journey.

-

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Meeting summarization is a task that has received significant attention in recent years (Wang and Cardie [2013]; Kulkarni et al. [2020], Li et al. [2019]) with the constant growth of produced meeting transcripts. Meetings play a crucial role in many organisations and are critical for decision-making and information sharing (Cascio [2000]). However, meetings can also be time-consuming as they usually contain a lot of information, which can be overwhelming, making it challenging to extract the essential points. With the recent growth of remote work, it is even more important to be able to extract information from meetings for people who did not attend those (Spataro [2020]).

This led to an exponential growth of meeting transcripts, which are often lengthy, each one of them often addressing many topics (Cascio [2000]). Hence, using common summarisation models that provide high-level summaries does not satisfy the needs of individuals who require detailed information based on a specific query, such as the speakers' opinions, actions and decisions (Wang and Cardie [2013]). This has led to a growing focus in query-based meeting summarization, where the objective is to provide a concise summary in response to a specific query (Mohamed and Rajasekaran [2006], Zhong et al. [2021], Murray et al. [2005]). A dataset has been created by (Zhong et al. [2021]) specifically for this task and has been established to be a benchmark dataset that we will use in our study.

The existing query-based meeting summarisation models mainly focus on single-domain settings, which limits their generalizability. This leads to a failure of understanding the nuances of meetings across different domains, resulting in poor out-of-domain performance. Hence, these models require domain-specific datasets in order to perform well on a particular domain. However, the creation of such datasets for a specific domain requires a lot of resources (Zhong et al. [2021]), which lays down the need for techniques to improve out-of-domain performance of existing models that were trained on other domains than the target one. Indeed, with a query-based meeting summarisation model that generalises well to various domains, performing this task on unseen domains would not require the creation of additional data, thus could be used for a wide range of applications.

Poor out-of-domain performance can be due to the lack of data, and with the lack of data for query-based meeting summarisation, finding ways to augment the existing data could help improve existing models' performance in out-of-domain settings. This became even more feasible with the outstanding recent advancements in state-of-the-art language models, such as GPT (Maynez et al. [2020]), which can be used to paraphrase existing data, thus resulting in meeting transcripts that contain the same information, but with a significant change in their structure.

In this context, the motivation for this thesis is to make a significant contribution to the field of query-based meeting summarization by addressing the issue of poor out-of-domain performance. By leveraging data augmentation techniques such as back-translation, and novel state-of-the-art models like GPT, we aim to improve the generalizability of models across different domains, making it a more versatile tool for organizations.

1.2 Objectives

The primary objective of this dissertation is to improve the out-of-domain performance of query-based meeting summarization models, hence the motivation of this thesis is to answer the following research question:

Can data augmentation techniques improve the performance of query-based meeting summarisation models on out-of-domain data?

To answer this research question, we will use a query-based meeting summarisation model introduced by (Mao et al. [2022]) as our baseline implementation, and apply various data augmentation techniques to its training data, such as back-translation and paraphrasing using GPT. We will explore whether these techniques can add enough diversity to existing data and allow the model to perform well in out-of-domain settings.

Overall, this research question aims to address the challenges of meeting summarization in the real world, where there is wide range of meeting domains. By answering this question, this paper aims to contribute to the development of meeting summarization models that are robust and can generalize to new domains.

1.3 Contributions

This dissertation presents several contributions to the field of query-based meeting summarization:

1. We thoroughly evaluate the ability of existing query-based meeting summarisation models to adapt to new domains. Our findings indicate that existing models have limited efficacy in handling domain-specific variations.
2. We explore data augmentation approaches for improving model performance for in-domain and out-of-domain settings using back-translation and paraphrasing with GPT. Our findings indicate that back-translation is not suitable for the query-based meeting summarisation task due to its inability to introduce variety,

while paraphrasing with GPT turns out to be a promising approach which gives great results. Hence, we provide information regarding the careful consideration required when selecting data augmentation techniques.

3. We present a novel data augmentation technique using GPT language models, which enables us to improve the quality of our model's summaries across all domains. This approach not only improves our model's performance on out-of-domain data, but also significantly improves its in-domain performance.
4. We provide a thorough analysis of the variations in the performance in function of the characteristics of the domains. By highlighting the unique challenges and characteristics of each domain, we explain that data augmentation techniques that are applied to structured and formal data can significantly improve the models' performance on both formal and informal meeting transcripts. Hence, we give guidance regarding the types of data that should be chosen for data augmentation. Specifically, we identify key requirements that a dataset should meet for paraphrasing to be a successful data augmentation technique.

1.4 Overview of the thesis

This thesis aims to address the issue of out-of-domain performance in query-based meeting summarization. The main objective is to improve the performance of existing models by exploring data augmentation techniques and evaluating their effectiveness.

Chapter 2 provides a literature review of existing work in the field, highlighting their strengths and limitations, while helping the reader understand the knowledge required to appreciate the content addressed in our paper. Chapter 3 describes the methodology and experimental design for this study, including the dataset introduced by (Zhong et al. [2021]), our baseline implementation using the model introduced by (Mao et al. [2022]), and the data augmentation techniques used to perform our research, namely back-translation and paraphrasing using GPT.

Chapter 4 presents the experimental results, where we analyse the performance of the proposed solutions and compare the differences in performance for both in-domain and out-of-domain settings. This chapter also includes an analysis of the differences between the augmented data and the original data, which help understand the variations in performance introduced by both data augmentation techniques.

Finally, Chapter 5 provides a conclusion and discusses the implications of the research findings, including their potential for future research in the field of query-based meeting summarization. The chapter also discusses the limitations of this study and provides recommendations for future work.

Chapter 2

Background

This chapter gives the readers the background required in order to understand the research done in this thesis. In the following sections, we will review the different types of meeting summarization, the NLP techniques used in summarization, data augmentation techniques, and evaluation metrics used to measure the performance of summarization systems.

2.1 Text Summarization

Summarization is the task of generating a shorter version of a document while keeping the key information contained in it. It is an important problem in natural language processing (NLP) that has many real-world applications, such as news article summarization (Eyal et al. [2019]), legal document summarization (Parikh et al. [2021]), email summarization (Zhang et al. [2021]), and has applications across various fields, such as journalism (Handler and O'Connor [2017]), business (La Quatra and Cagliero [2020]), education (Miller [2019]), and research (Haruna et al. [2022]).

Summarization can be defined mathematically as follows: given a document $D = s1, s2, ..., sn$ consisting of n sentences, the goal of text summarization is to find a summary $S = s1', s2', ..., sm'$ consisting of $m' \leq n$ sentences, that maximizes the informativeness of the summary while minimizing its length. This can be represented as an optimization problem:

minimize $S = s1', s2', ..., sm'$ subject to:

- the length of the summary m' is less than or equal to some predefined length L.
- the informativeness of the summary is maximized, where the informativeness can be measured using some metric such as ROUGE or BLEU.

The objective function can be defined as a linear combination of various factors, including sentence importance, coherence, diversity, and redundancy, each of which can be represented mathematically. The optimization problem can be solved using various techniques, including neural network-based methods, but also integer linear programming, or greedy algorithms.

In this thesis, we will focus on a distinct genre of summarization known as meeting summarization. Meetings are often rich in information and can be time-consuming and difficult to follow. Hence, meeting summarization can help participants to quickly review the key points of a meeting. Previous work on meeting summarization has explored various approaches, including sentence extraction (McKeown et al. [2005]) and keyphrase extraction (Gillick et al. [2009]). However, meeting summarization is a challenging task due to the informal nature of meetings, the presence of multiple speakers, and the lack of standardized meeting structures (Feng et al. [2021b]). Additionally, evaluation of meeting summarization is challenging due to the subjective nature of summarization and the lack of annotated data, as discussed in Section 2.5.

Meeting summarization can be further classified into two sub-categories: non-query-based and query-based. Non-query-based summarization aims to summarize the entire meeting, while query-based summarization focuses on generating summaries that answer specific queries.

2.2 Query-based meeting summarization

Query-based summarization is a type of summarization that focuses on answering specific queries, unlike traditional summarization, which aims to provide a summary of the entire text. This task can be applied to various domains, such as news articles (Annisa and Khodra [2017]), scientific papers and meetings (Zhong et al. [2021]). Meeting transcripts can indeed be lengthy and contain a large amount of information, hence query-based meeting summarization models are an important tool for efficiently extracting relevant information from them.

Query-based meeting summarization can be mathematically defined by first introducing a query $Q = (w_1, \cdots, w_{|Q|})$ and treating the task as a sequence-to-sequence problem. Specifically, each meeting transcript $X = (x_1, x_2, \ldots, x_n)$ consists of n turns, and each turn x_i represents the utterance u_i and its speaker s_i, that is, $x_i = (u_i, s_i)$. Additionally, each utterance contains l_i words $u_i = (w_1, \ldots, w_{l_i})$. The objective is to generate a summary $Y = (y_1, y_2, \ldots, y_m)$ by modelling the conditional distribution

$$p(y_1, y_2, \ldots, y_m | Q, (u_1, s_1), \ldots, (u_n, s_n)) \tag{2.1}$$

Query-based meeting summarization is a challenging task due to the unique nature of meetings. Informal language, such as jargon, idiomatic expressions, and colloquialisms, is commonly used in meetings, which can make it difficult to extract relevant information (Feng et al. [2021b]). Additionally, meetings often do not have a clear structure, which can result in conversations going from one topic to another without clear transitions, making it difficult to summarise pertinent information.

Finally, data scarcity is a significant obstacle to effective query-based meeting summarization. Indeed, the creation of high-quality annotated data is costly and time-consuming, as it requires a lot of effort from human annotators. Furthermore, obtaining meeting transcripts is challenging due to privacy concerns within organisations.

Confidential information in meeting transcripts may require anonymization, which complicates the creation of large amounts of data.

Despite all these challenges, a recently introduced dataset named QMSum (Zhong et al. [2021]) has emerged as a benchmark for the query-based multi-domain meeting summarization task. This dataset contains meetings from three domains, namely, AMI (Carletta et al. [2006]), ICSI (Janin et al. [2003]), and committee meetings of the Welsh Parliament and Parliament of Canada. However, this paper presents certain limitations, as their proposed model performs poorly for out-of-domain data. The motivation behind this thesis is hence to address this issue and explore whether the out-of-domain performance can be improved by using data augmentation techniques. Before introducing the concept of data augmentation in Section 2.4, we will present the main techniques used in natural language processing for text summarization.

2.3 NLP techniques for query-based meeting summarization

Two-step models (Lee et al. [2011]) and end-to-end models (Karn et al. [2021]) are both commonly used for query-based meeting summarization. Two-step models use a locator model to identify the relevant sentences that answer the query, and a summarizer model to generate a summary based on the locator's selected sentences. In contrast, end-to-end models generate summaries directly from the input text using a single model. Two-step models may lack coherence and fail to identify important sentences, along with a locator model that may fail to identify important sentences. On the other hand, end-to-end models may lack specificity, and require large amounts of training data, thus making it more tedious to use them.

There are two main approaches that can be used for automatic text summarisation, namely extractive summarization and abstractive summarization. Extractive summarization approaches select a subset of sentences from the original document to form the summary, which is straightforward and can produce high-quality summaries, but it may suffer from redundancy and lack of coherence (Nenkova and McKeown [2011]).

On the other hand, abstractive summarization approaches involve generating new sentences that contain the main ideas of the document. This approach is of course more complex, but can produce more coherent summaries. Extractive summarization can preserve the original wording and meaning of the text, while abstractive summarization can capture more complex relationships between sentences and generate summaries that are more informative (See et al. [2017]). However, abstractive summarization can also introduce errors and distortions, and may require more sophisticated models and training data due to its additional complexity.

Extractive summarization was the first approach to summarization and is still widely used today, but the recent advancements of transformer models, such as BERT (Devlin et al. [2019]) and GPT (Maynez et al. [2020]), led to abstractive summarization being more feasible while showing great improvements in performance.

Indeed, the progress in summarisation tasks has been increasing with the recent ad-

vancements of pre-trained language models, such as BERT and BART (Lewis et al. [2019]). These models are trained on large amounts of data and can generate summaries of great quality. Pre-trained language models have shown very promising results in various summarization tasks, including news article summarization and scientific paper summarization (Garg et al. [2021]). However, there are also limitations to these models, such as the fact that they are computationally expensive to train and use, and have limited control, as pre-trained models are not designed to follow specific rules or constraints, making it difficult to generate summaries that follow specific requirements (Xu et al. [2021]). Given that both extractive and abstractive approaches present benefits, we opted to use in our study a hybrid model that uses an extractive approach for the locator, and an abstractive approach for the summariser.

Given the limited amount of data available for query-based meeting summarization and the difficulty of the task, achieving good out-of-domain performance is essential to avoid having to create new datasets for each new domain. Data augmentation techniques can address this challenge and become a good solution to improve out-of-domain performance.

2.4 Data augmentation

Data augmentation is already a common technique in NLP for improving model performance, as it increases the size and diversity of training data (Feng et al. [2021a]) without collecting additional data. Hence, by increasing the amount of data along with its diversity, data augmentation can help to improve model generalization and improve out-of-domain performance, with the model's predictions remaining consistent despite these changes (Feng et al. [2021a]).

There are several common techniques for data augmentation in NLP, such as synonym replacement and EDA (Easy Data Augmentation) (Wei and Zou [2019]), back-translation (Beddiar et al. [2021]), and paraphrasing (Kumar et al. [2019]).

Synonym replacement consists in replacing words in the existing data with their synonyms, while EDA consists in applying various transformations to it, such as randomly deleting words. Both of these techniques can be useful in scenarios where the meaning of the sentence is still preserved even if some words are replaced with synonyms or transformed. For example, in a sentiment analysis task, replacing words with their synonyms can improve the ability of the model to generalize to new data (Fiarni et al. [2016]). Similarly, EDA is beneficial in text summarization where the objective is to generate a concise summary that captures the key information of the data, without the need to answer specific queries, in which case random transformations may still impact the quality of the summary, but in a moderate way (Somayajula et al. [2022]).

However, in query-based meeting summarization, the objective is to generate summaries that answer specific queries, hence replacing words with their synonyms or applying random transformations could produce summaries that are irrelevant to the query. In other words, the synonyms may not convey the same meaning as the original words, and this could lead to a summary that is misleading. For example, randomly deleting words could remove key information that is necessary to answer a specific query.

For these reasons, there are two other candidates to be explored for data augmentation for query-based meeting summarization for our research, namely back-translation and paraphrasing. Back-translation consists in translating sentences from a source language to another language, and then translating them back to the source language. Paraphrasing involves using a model to paraphrase the data to create new variations of the same text. This has become even more achievable with the recent advancements in state-of-the-art models such as BERT and GPT, as discussed previously.

The advantage of these two techniques is that they allow to generate new sentence structures. With new sentence structures, the augmented data can be sufficiently different from the original data to help improve the diversity of the training data. This in turn can lead to the model capturing better the nuances of the language and improve its performance. However, it is important to note that both paraphrasing and back-translation may result in low-quality output, highlighting the need for careful evaluation of the augmented data (Beddiar et al. [2021]).

2.5 Evaluation metrics

Query-based meeting summarization is a challenging task due to the informal nature of meetings, the presence of multiple speakers, and the lack of standardized meeting structures (Zhong et al. [2021]). In addition, evaluation of meeting summarization is challenging due to the subjective nature of summarization and the lack of annotated data. Both of these challenges add complexity to the query-based meeting summarisation task, leading to the need for a thorough evaluation of the generated summaries.

There are two types of evaluation metrics, namely automatic and manual. Automatic evaluation metrics are computed using algorithms that compare the generated summary with the reference summary, whereas manual evaluation metrics require humans to rate the quality of the generated summary.

Automatic evaluation metrics have the benefit that they are faster and more objective than manual evaluation metrics, but they do not always reflect the quality of the generated summary accurately, as they do not always capture the nuances of meeting summarisation, such as understanding the context, the purpose of the meeting, or the speaker's intentions (Lloret et al. [2018]). On the other hand, manual evaluation metrics can capture these nuances and provide a more accurate measure of a summary's quality, but they come with the disadvantage of being time-consuming and costly. Hence, given that manual evaluation may be subjective and resource-intensive, automatic metrics are more relevant in the context of this study.

The most common automatic metrics used in summarization are ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin [2004]), BLEU (Bilingual Evaluation Understudy) (Post [2018]), and METEOR (Metric for Evaluation of Translation with Explicit ORdering) (Banerjee and Lavie [2005]).

ROUGE is a metric based on the concept of n-gram co-occurrence and measures the overlap between the generated summary and the reference summary. BLEU is another popular metric but has been criticized for its inability to capture the diversity

of the generated summary, as it primarily focuses on n-gram precision (Post [2018]). The authors of this paper show that BLEU is sensitive to the length of the summary, penalizing longer summaries even if they contain the necessary information. On the other hand, METEOR has been criticized for being too complex and having poor correlation with human judgments, and has shown to be less reliable than ROUGE (Banerjee and Lavie [2005]). Hence, ROUGE seems to be the most appropriate metric for query-based meeting summarisation in our research.

However, ROUGE also has some limitations, such as the fact that it does not take into account the quality of the generated summary beyond surface-level similarities with the reference summary (Lloret et al. [2018]). The authors of this paper also state that ROUGE does not consider the coherence of the generated summary, which are important aspects of text summarization.

Facing these limitations, researchers attempted to develop alternative metrics, such as the Content Divergence Score (CDS) and the Pyramid Evaluation Metric (PEM) (Nenkova et al. [2007]), which take into account the quality of the generated summary beyond surface-level similarities with the reference summary, as opposed to ROUGE. Despite addressing some of the limitations of ROUGE, these newer metrics may also introduce additional complexity and may not necessarily provide a significant improvement over ROUGE that would justify their use for our study. Thus, given that ROUGE is widely used and correlates well with human judgments, it is a relevant evaluation metric to employ for query-based meeting summarization research.

Chapter 3

Methodology

In this chapter, we will first present the dataset used in our study, justify our choices regarding this dataset selection, and show that there is a need for data augmentation techniques to improve the out-of-domain performance of existing query-based meeting summarisation models. We then describe the model that we have used as a baseline to perform our experiments, which jointly trains a locator model, and a summariser model. Finally, we will dive into the two augmentation techniques used to perform our research, namely, back-translation and paraphrasing with GPT.

3.1 Benchmark dataset and data collection

In this section, we present the QMSum dataset (Zhong et al. [2021]), discuss its components, the results obtained from the paper, and its limitations in out-of-domain performance. We will end the section by explaining the modifications that we have done to the dataset, such as the preprocessing steps we applied to it.

3.1.1 Overview of the QMSum dataset

The QMSum dataset, introduced by Zhong et al. [2021], contains in total 232 meetings with 1,808 question-summary pairs, and is composed of three types of meeting data that are annotated using query-summary pairs: product meetings, academic meetings and committee meetings.

Each meeting transcript is associated with various queries, meaning that a meeting can span multiple queries. Each query can be either of these two categories:

- *General queries*: Queries related to general information, such as the contents of whole meetings.
- *Specific queries*: Queries related to relatively detailed information, such as the discussion about certain topics.

We provide below a more thorough description of each domain, along with concrete examples taken directly from these datasets (product, academic and committee) in Table

3.1, Table 3.2 and Table 3.3.

3.1.1.1 Product meetings

(Carletta et al. [2006]) introduced the AMI dataset, composed of 137 meetings on product design in an industrial context. This dataset includes transcripts of meetings that describe the process of designing a new remote control, along with their corresponding summaries.

Query	Query type
Summarize the whole meeting	General
Summarize the groupmates' self-introduction and the project introduction.	Specific
What did the group discuss about the email they received on the project announcement?	Specific

Table 3.1: Example of two specific queries and one general query, associated with a Product meeting transcript (ID ES2002a).

3.1.1.2 Academic meetings

The ICSI dataset, created by (Janin et al. [2003]), contains 59 group meetings focused on research topics among students that took place on a weekly basis at the International Computer Science Institute (ICSI) in Berkeley, along with the corresponding meeting summaries.

Query	Query type
Summarize the whole meeting	General
Summarize the discussion about the current XML format to link up different components in data	Specific
What did F think about the current XML format to link up different components in data?	Specific

Table 3.2: Example of two specific queries and one general query, associated with an Academic meeting transcript (ID Bdb001).

3.1.1.3 Committee meetings

Committee meetings contain formal discussions on many subjects, such as education system reforms and public health. It contains 25 committee meetings from the Welsh Parliament and 11 from the Parliament of Canada.

Query	Query type
Summarize the whole meeting	General
Summarize the debate about the flaws in government‘s pandemic relief program.	Specific
How did racism and long-term care related to government’s policy?	Specific

Table 3.3: Example of two specific queries and one general query, associated with a Committee meeting transcript (ID covid_0).

3.1.2 Advantages of the QMSum dataset

Amongst the available datasets available for query-based meeting summarisation, we have specifically chosen the QMSum dataset for our study as it offers several advantages over previous datasets for our task. Firstly, QMSum is the largest meeting summarization dataset available, with 232 meetings and a total of 1,808 question-summary pairs. Detailed statistics of the QMSum dataset are provided in Table 3.4, with amounts of data contained in the training, validation and testing splits, which are shown in Table 3.5. The large size of this dataset, compared to previous ones, allows exploring more deeply the architectures of various models and, most importantly, data augmentation techniques.

Datasets	# Meetings	# Turns	Len. of Meet.	Len. of Sum.	# Speakers	# Queries
Product	137	535.6	6007.7	70.5	4.0	7.2
Academic	59	819.0	13317.3	53.7	6.3	6.3
Committee	36	207.7	13761.9	80.5	34.1	12.6
All	232	556.8	9069.8	69.6	9.2	7.8

Table 3.4: Figure provided by QMSum (Zhong et al. [2021]), showing the statistics of the QMSum dataset.

Datasets	Train	Valid	Test
Product	690	145	151
Academic	259	54	56
Committee	308	73	72
All	1,257	272	279

Table 3.5: Figure provided by QMSum (Zhong et al. [2021]), showing how many query-summary pairs are contained for each split across all domains.

Additionally, the QMSum dataset is designed with a multi-domain focus, including Product, Academic, and Committee meetings (Zhong et al. [2021]). This brings an improvement compared to previous datasets, which focused on a single domain. Thus, it was a natural choice to use this dataset when exploring techniques to improve out-of-domain performance of models.

Furthermore, the QMSum dataset brings a focus on capturing specific contents of

meetings by having shorter summary lengths, with an average length of 69.6 words. This contrasts with the previous datasets such as AMI and ICSI, which contain longer summaries. The brevity of summaries in QMSum challenges models to accurately capture relevant information and compress it into concise summaries, thus pushing the boundaries of current meeting summarization methods.

Lastly, the QMSum dataset was created using a thorough annotation process. Professional annotators were provided with meeting transcripts and specific queries, and they were asked to summarize the relevant parts of the meeting based on those queries. This process resulted in a rich set of query-summary pairs that can be used in our research to explore data augmentation approaches to improve out-of-domain performance.

3.1.3 QMSum's Limitations in Out-of-Domain Performance

Table 3.6 depicts the performance of QMSum's BART (Lewis et al. [2019]) summarization model, when using gold spans instead of a locator model. The results indicate that out-of-domain performance is significantly lower than in-domain performance, with the best scores achieved when training and testing on the same domain, except for the Academic domain. Indeed, the highest R-2 and R-L scores for when tested on the Academic domain are achieved when the model is trained on three domains. Further analysis of these results are provided in Chapter 4.

We observe that training the model on Product and testing it on the Product domain results in R-1, R-2, and R-L scores of 35.43, 10.99, and 31.37, respectively. However, when trained on the Academic or Committee domain and evaluated on Product, there is a significant decrease in the R-1 score by 23%, 56%, and 23%, respectively. Therefore, the results suggest the need for techniques to help the model perform better on out-of-domain data, which is the objective of our study.

Before describing the data augmentation techniques we used to address this problem in Section 3.3, we will now provide an overview of the preprocessing pipeline we applied to the QMSum dataset, followed by a presentation of the baseline implementation used for our research in Section 3.2.

Datasets	Product			Academic			Committee			All		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Pro.	**35.43**	**10.99**	**31.37**	22.59	3.41	19.82	24.48	3.84	21.94	30.02	7.58	26.62
Aca.	27.19	4.86	24.09	**26.69**	4.32	22.58	27.84	4.29	25.10	27.22	4.59	24.02
Com.	25.56	3.48	22.17	23.91	2.99	20.23	**32.52**	**6.98**	**27.71**	27.07	4.28	23.21
All	34.93	10.78	31.21	26.47	**5.05**	**23.01**	31.16	6.47	27.52	**32.18**	**8.48**	**28.56**

Table 3.6: Figure provided by QMSum (Zhong et al. [2021]), showing the multi-domain and cross-domain summarization experiments. Each row represents the training set, and each column represents the test set. The cells with text written in **bold** denote the best result on the dataset in this column. Standard Rouge F-1 score is used to evaluate the model's performance.

3.1.4 Preprocessing

To preprocess our data, we first extracted input data from JSON files provided by QMSum, and applied various methods from the Natural Language Toolkit [1] (nltk) module, and tokenized them using RobertaTokenizer [2] (for the locator model) and BartTokenzier [3] (for the summariser model) from the HuggingFace Transformers library.

3.1.4.1 Tokenization

Tokenization is the process of breaking down the input text into smaller units, such as words or subwords. Using the nltk module, we tokenize the input into words, convert them to lowercase (normalisation), remove punctuation, and join them back together into a single string separated by spaces.

Many pre-processing techniques in NLP include stop words removal and stemming (Gharatkar et al. [2017]). However, we have not used these techniques, as they might remove or modify words that are actually important for our task. Indeed, in some cases, stop words carry important contextual information, while in other cases, stemming can modify a word in a way that changes its meaning, which can lead to a poorer performance of our model (Kathiravan and Haridoss [2018]). Additionally, we use the pre-trained language models RoBERTa and BART, which have already been trained on a large amount of data that includes a wide range of language patterns, including stop words. This means that the tokenizers associated with these models have already learned how to handle these stop words, and attempting to remove them could negatively impact the performance of the model (Gerretzen et al. [2015]).

3.1.4.2 Cleaning

We also perform cleaning, which is the process of removing unwanted characters and symbols from the input text. We remove specific markers, abbreviations, and pauses from the input text, by performing various string replacements on the input text to remove tags and markers used in the QMSum dataset. These replacements include removing the vocalsound , disfmarker , pause , nonvocalsound , and gap tags, as well as replacing various abbreviations like a_m_i_ and t_v_ with their full forms (ami and tv, respectively).

Figure 3.1 explains the process of normalisation, tokenisation and cleaning with a sample sentence from the QMSum dataset. We then use the output from this sequence of operations and convert it to input IDs.

3.1.4.3 Converting to input IDs

Given that we use a model which contains a retriever employing RoBERTa, and a summariser employing BART (as later explained in Section 3.2), different tokenization

[1] https://www.nltk.org/

[2] https://huggingface.co/docs/transformers/model_doc/roberta

[3] https://huggingface.co/docs/transformers/model_doc/bart

Figure 3.1: Visualisation of the steps performed in manual processing. This figure shows, for each step, the output of the relevant operation for a sentence taken from a Product meeting (ID ES2005a).

methods are required for the input data. To tokenize the input data for the retriever task, the RoBERTa model is fine-tuned and therefore, the RoBERTaTokenizer is employed as it is specifically designed for pre-training and fine-tuning RoBERTa models. On the other hand, the BART model, which is a denoising autoencoder well-suited for text generation tasks, is used for the generator task. Thus, the BartTokenizer is used for this purpose, as it is designed for pre-training and fine-tuning BART models.

After normalising the text data, tokenizing and cleaning it using the steps described in Section 3.1.4.1 and Section 3.1.4.2, we convert the tokens into input IDs. These input IDs are numerical representations of the tokens that can be fed into the corresponding pre-trained model. We created input IDs by using the RobertaTokenizer for the retriever, and BartTokenizer for the summariser. By using a different tokenizer for the locator and the summarizer, we ensured that our input IDs were compatible with the respective pre-trained models.

3.1.4.4 Truncation

The input IDs are then truncated to a specified maximum length, which is a necessary step, as transformer models, such as RoBERTa and BART, have fixed input lengths. If the input length exceeds the model's limit, then the model will not be able to process it.

The maximum input length for RoBERTa refers to the maximum number of tokens that can be fed to the RoBERTa model at one time, and it is set to 512. This limit is necessary as the model can only process inputs up to a certain length due to memory constraints. Hence, with truncation, if an input exceeds the maximum length, it will be split into smaller chunks to be processed separately.

The maximum source length and maximum target length for BART refer to the maximum number of tokens that can be used for the input and output sequences, respectively.

The maximum source length is set to 64, and it determines the maximum number of tokens in the source document that can be fed into the BART generator. If the source document exceeds this length, it must be split into smaller chunks, and each chunk is then separately fed into the BART generator. The maximum target length is set to 900, which refers to the maximum number of tokens that can be used to produce the output. This parameter guarantees that the summary is neither too long, nor too short, as if the summary exceeds the maximum target length, it will be truncated.

3.1.4.5 Padding

After truncation, padding is added to ensure that all inputs have the same length. Given that our pre-trained models require inputs of fixed length, padding guarantees that the model can process batches of inputs efficiently. In our context, padding is added to the right side of the input sequence using the padding token specified in the tokenizer.

After tokenising the input and converting it to input IDs, we created input sequences for the models. These input sequences contained the query associated with the relevant meeting transcript, with special tokens such as $< s >$ (start) and $< /s >$ (end) to indicate the beginning and end of different segments. For instance, we used the following input format for the retriever model: "$< s >$ Query $< /s >$ Relevant Text Spans $< /s >$". Preparing the inputs this way enabled the models to process and understand the relationship between the query and the corresponding meeting transcripts. Figure 3.2 illustrates the tokenization process performed by the RoBERTa tokenizer, including the conversion from the original text to input IDs and padding.

3.2 Baseline Implementation

Extract-then-generate approaches are commonly used for long-input summarisation (Zhong et al. [2021]), a type of summarisation that query-based meeting summarisation falls into, due to the recurrent large length of meeting transcripts. These approaches use a locator (also known as extractor or retriever) to locate the relevant text spans in the meetings that answer the query, and a generator (also known as summariser) to summarise these extracted text snippets. However, extract-then-generate approaches usually train the extractor and the generator separately, which can limit their performance as they suffer from cascaded errors from the extractor to the generator (Mao et al. [2022]).

Hence, (Mao et al. [2022]) proposed a variation of the extract-then-generate approach by introducing a new model specifically designed for long-input summarisation: Dynamic Latent Extraction for Abstractive Summarization (DYLE). DYLE jointly trains the extractor and the generator in an end-to-end manner, and keeps the extracted text snippets latent.

By keeping the extracted text snippets latent, DYLE can learn in a more flexible way the representation of the input document and generate more accurate summaries, as it can capture the most relevant information from the input document. This is proven by the results they obtain, which largely outperform existing models using the QMSum dataset.

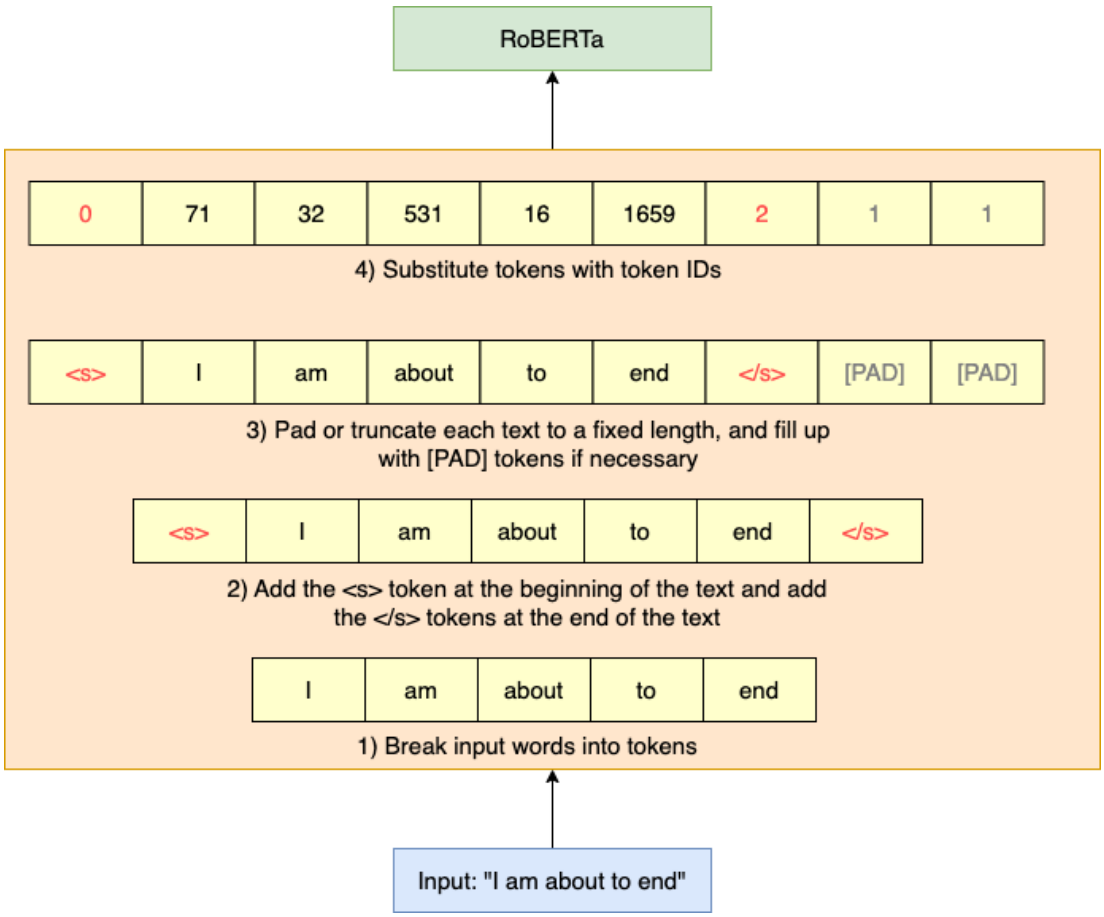

Figure 3.2: RoBERTa input formatting schema that shows the steps required to convert a sentence to token IDs to be used by RoBERTa. We use an example of sentence taken from a Product meeting (ID ES2003b).

In the following sections, we present the model architecture details of DYLE, which we used as our baseline implementation, and explain how it leverages the power of pre-trained language models such as RoBERTa and BART (both available in the HuggingFace [4] library) for a state-of-the-art model for long-input summarisation. We will also present the hyperparameters used to train the model and our training process, including the optimization methods and regularizers employed. Finally, we will discuss the performance metrics used to evaluate the retriever and the generator.

3.2.1 Model Architecture

In the long-input summarization task, the input consists of L text snippets, $X = (x_1, \ldots, x_L)$, and an optional query q if a query is paired with a summary. The output is a summary y of length T. Using dialogue utterances by each speaker as snippets, the goal is to learn a model that generates a sequence of summary tokens y given the input snippets X and the previously generated tokens $y < t$:

[4] https://huggingface.co/docs/transformers/model_doc/bart

$$P_\theta(y \mid q, X) = \prod_{t=1}^{T} P_\theta(y_t \mid q, X, y_{<t}) \tag{3.1}$$

DYLE offers a variation of extract-then-generate approaches to solve this task by incorporating a dynamic weighting mechanism for the extracted snippets. An overview of DYLE's model is presented in Figure 3.3. In the following sections, we will dive deeper into each component observed in this figure, with a particular focus on the extractor, the generator and the various losses used to compute the training objective.

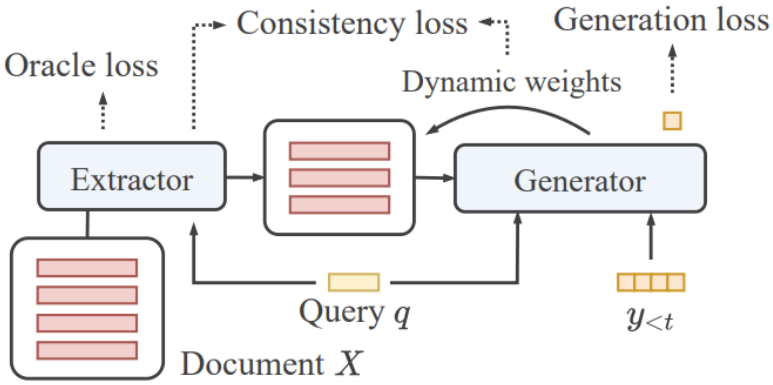

Figure 3.3: Figure provided from Mao et al. [2022]. Overview of DYLE's approach, where the input is a document X (each $x \in X$ is a sentence) and an optional query q, and the output is a summary y.

3.2.1.1 Extractor

RoBERTa is a pre-trained transformer-based language model designed for various NLP tasks, including text classification, question answering, and natural language inference (Liu et al. [2019]). In the context of the locator, it is used to encode the input data and produce embeddings that can be used for similarity matching with the query, making it a suitable choice for a locator model.

The extractor module takes in the input document snippets $X = (x_1, ..., x_L)$ and the query q, and outputs a score $s_i = E_\eta(q, x_i)$ for each snippet x_i, where η represents the parameters of the extractor. The K most relevant snippets are extracted and used as the input for the generator:

$$X_K = top-K(E_\eta(q, x_i), x_i \in X) \tag{3.2}$$

For our implementation, we chose $K = 20$. Figure 3.4 illustrates the process by which the top K snippets are extracted.

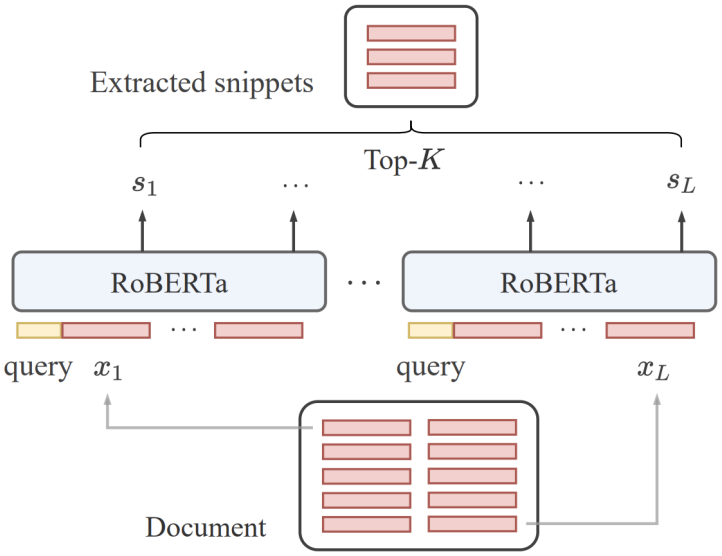

Figure 3.4: Figure provided by Mao et al. [2022]. Long-input extractor. Each document is divided into chunks, each containing consecutive snippets. A shared RoBERTa encodes each chunk independently.

3.2.1.2 Generator

BART is a pre-trained sequence-to-sequence model used in generation tasks, such as text summarization, machine translation, and dialogue generation (Lewis et al. [2019]). In the context of the generator, it is used to generate the summary based on the encoded document and query representations produced by the retriever.

Thus, BART is better suited for generation tasks such as summarization, which is the task of the generator in this case.

The generator module takes in the extracted snippets $X_K = (x_{k_1}, ..., x_{k_K})$, the query q, and the previously generated tokens $y_{<t}$, and generates the next token y_t using the conditional probability $P_\theta(y_t|q, X_K, y_{<t})$. As mentioned earlier, in contrast to previous work, DYLE's generator incorporates a dynamic weighting mechanism, which assigns a weight $w_i = P_\theta(x_i|q, X_K, y_{<t})$ to each extracted snippet x_i at each decoding time step t. The dynamic weighting mechanism allows the model to understand how it uses the extracted snippets and helps improve the extraction process by down-weighting irrelevant snippets.

3.2.1.3 Loss functions

DYLE uses various loss functions both for the extractor and the generator, such as the oracle loss, the consistency loss and the generation loss, as seen in Figure 3.3:

- **Oracle loss**: DYLE uses extractive oracles, denoted as X_o, to supervise the extraction component of the model. The extractive oracle loss $L_{\eta_{oracle}}$ is calculated based on the cross-entropy loss between the text snippets selected by the extractor

and the extractive oracle.

- **Consistency loss**: DYLE uses dynamic weights to train the extractor, where the averaged dynamic weights represent the overall importance of the snippet. The consistency loss measures the distance between the averaged dynamic weights distribution and the extractor distribution.
- **Generation loss**: The generation loss is defined as the negative log-likelihood of the gold summary: $L_{\theta_{gen}} = -\log P_\theta(y|q, X_K)$

The overall training objective of the model is a combination of these three losses:

$$L_{\theta,\eta} = \lambda_g L_\theta^{gen} + \lambda_c L_\eta^{consist} + \lambda_o L_\eta^{oracle}, \tag{3.3}$$

where λ_g, λ_c, and λ_o are hyperparameters that control the weights of each loss component. The generator is optimized using the generation loss, while the extractor is optimized using both the consistency loss and the oracle loss.

In the next section, we will present the specific hyperparameters and training process used in our experiments.

3.2.2 Implementation details

We trained our model on a single NVIDIA A100 GPU using gradient checkpointing to save memory, with an effective batch size of 8.

We used the Adam optimizer for both the locator and generator components, with a learning rate set to 5×10^{-5} for the locator, and a learning rate of 5×10^{-6} for the generator.

As explained in Section 3.2.1.3, DYLE uses three loss functions during training, each with its own hyperparameter: generation loss (λ_g), oracle loss (λ_o), and consistency loss (λ_c).

Here are the following hyperparameters that we used:

- $\lambda_g = 1$: The coefficient for the generation loss in the training objective (Equation 3.3). DYLE performed a 2-step binary search between 0 and 2 to find the optimal value.
- $\lambda_o = 1$: The coefficient for the oracle loss in the training objective. DYLE performed a 2-step binary search between 0 and 2 to find the optimal value.
- $\lambda_c = 1$: The coefficient for the consistency loss in the training objective. DYLE performed a 3-step binary search between 0 and 10 to find the optimal value.

3.2.3 Evaluation Metrics

We use the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric family, a common automatic metric used for summarisation tasks (Lin [2004]) that we employ to evaluate the locator and the summariser. For the locator, ROUGE measures the

overlap between the extracted text spans and the gold relevant text spans, whereas for the summariser, it measures the overlap between the generated summary and the reference summary at the n-gram level. Specifically, we use:

- **ROUGE-1 (R-1)**, which measures the overlap between unigrams.
- **ROUGE-2 (R-2)**, which measures the overlap between bigrams.
- **ROUGE-L (R-L)**, which measures the longest common subsequence (LCS) between the summary and the reference.

The ROUGE score is calculated as the ratio between the number of overlapping n-grams and the total number of n-grams in the reference summary (for the summariser) or gold spans (for the locator). The ROUGE scores are computed for each summary-meeting pair and then averaged over the entire test set. When presenting our results in Chapter 4, we use the standard ROUGE-1, ROUGE-2, and ROUGE-L F1 scores.

3.3 Data augmentation

In this chapter, we present and justify our choices regarding the data augmentation techniques we used to attempt improving DYLE's out-of-domain performance for query-based meeting summarization using the QMSum dataset. We will then provide a description of the implementation details of the selected techniques, along with a discussion regarding the benefits and limitations of these techniques.

3.3.1 Back-Translation

Back-translation is a commonly used technique in NLP for augmenting training data with the aim to generate new text and add diversity to the training set by introducing sentences with new phrasing and structure while conveying the same meaning (Liu et al. [2022]). This method consists in translating the original text into a different language, and then translating it back to the original language using machine translation models. Figure 3.5 illustrates how back-translation works with an example sentence that is translated from English to French, and then back from French to English.

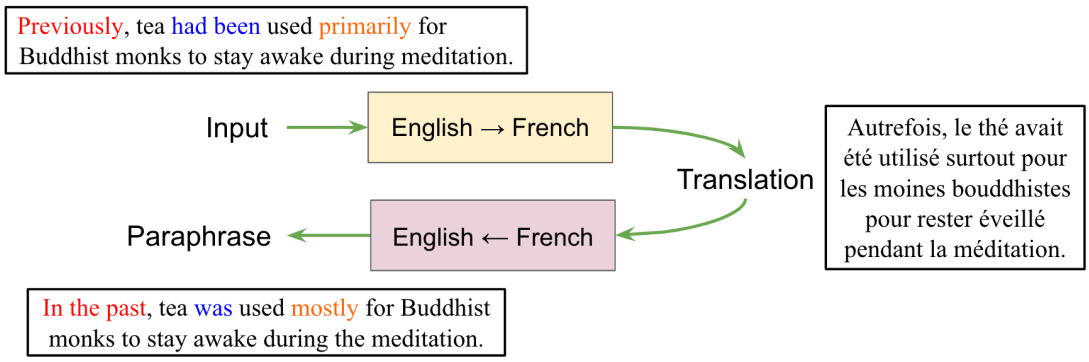

Figure 3.5: Back-Translation example of converting a sentence from English to French, and then back from French to English.

3.3.1.1 Implementation Details

To implement the back-translation technique, we used the Google Translate API [5] to translate the original English meeting transcripts and query-answer pairs into French, and then back to English. We used the following hyperparameters for the Google Translate API:

- Source language: English
- Target language: French
- Translation model: Neural Machine Translation (NMT)
- Batch size: 32

Regarding the other hyperparameters, we used the default values provided from the Neural Machine Translation (NMT).

After translating the original English text into French, we then translated the foreign language version back into English. An example of such augmented query-answer pair from QMSum is shown in Table 3.7.

To ensure the quality of the generated data, we manually reviewed a subset of the translated text to ensure that it accurately reflected the meaning of the original English text.

	Original Input	Augmented Output
Query	Why did Marketing recommend to specify the target market when discussing details of button design and location function?	Why did marketing recommend specifying the target market when considering button design details and placement function?
Answer	Project Manager thought that the interface design was still not intuitive and useful enough for now. Marketing agreed and pointed out that the present target group might be too large. Marketing suggested that the team should figure out specifically for whom they intended to design the interface in case the customers were confused about the remote control and got dissatisfied.	The project manager felt that the interface design was not intuitive and useful enough yet. Marketing agreed and pointed out that the current target group might be too large. Marketing suggested that the team should determine specifically who they intended to design the interface for in case customers were confused about the remote and dissatisfied.

Table 3.7: Example of an augmented query-answer pair with back-translation with French related to a Product meeting (ID ES2003b). Substitutions are marked in orange, deletions are marked in red and additions are marked in blue.

[5] https://cloud.google.com/translate

3.3.2 Data augmentation with GPT

In this section, we present the other data augmentation technique used in our study, namely paraphrasing with GPT. We first discuss the various approaches that could have been used and justify the choices we made. We then proceed to describe the implementation details for this technique we chose.

3.3.2.1 Various approaches

Paraphrasing is a useful technique for data augmentation in the context of query-based meeting summarization. In our case, we used GPT, a state-of-the-art language model (Floridi and Chiriatti [2020]), which can be used in various ways. One approach is to use GPT to paraphrase the original input, whereas another approach is to use GPT for text generation, which would generate new output based on the original input.

Using GPT to paraphrase the original data involves inputting the text into the GPT model and generating new text that is similar in meaning but contains different words and phrasing. However, the generated text may not be diverse enough, especially if the original text is complex or contains specialized vocabulary. Using the text generation approach could instead provide more variety, as it is creating something new, rather than modifying an existing input. Table 3.8 shows an example of an augmented query-answer pair from the QMSum dataset using paraphrasing with GPT, and Table 3.9 shows the same query-answer pair augmented twice using the text generation approach.

	Original Input	Augmented Output
Query	Why did Marketing recommend to specify the target market when discussing details of button design and location function?	What was the reason behind Marketing's suggestion to identify the target audience when discussing the specifics of button design and location function?
Answer	Project Manager thought that the interface design was still not intuitive and useful enough for now. Marketing agreed and pointed out that the present target group might be too large. Marketing suggested that the team should figure out specifically for whom they intented to design the interface in case the customers were confused about the remote control and got dissatisfied.	The Project Manager believed that the interface design was not currently user-friendly and effective. Marketing agreed and mentioned that the current target audience might be too broad. Marketing suggested that the team should determine the specific target audience for whom the interface is intended to avoid confusion and dissatisfaction among customers who might have trouble using the remote control.

Table 3.8: Example of an augmented query-answer pair using paraphrasing with GPT related to a Product meeting (ID ES2003b). Substitutions are marked in orange, deletions are marked in red and additions are marked in blue.

	Query	Answer
Input	Why did Marketing recommend to specify the target market when discussing details of button design and location function?	Project Manager thought that the interface design was still not intuitive and useful enough for now. Marketing agreed and pointed out that the present target group might be too large. Marketing suggested that the team should figure out specifically for whom they intented to design the interface in case the customers were confused about the remote control and got dissatisfied.
Invalid augmented query-answer pair	How did Project Manager describe the voice recognition service embedded on the computer during the discussion of button design and location function?	Project Manager exemplified the robust voice recognition service embedded on the computer, indicating that it would drag and dictate but also require a lot of memory.
Valid augmented query-answer pair	Were there any concerns expressed during the discussion of button design and location function regarding the incorporation of voice recognition?	Yes, Project Manager expressed concerns about the power required and the cost.

Table 3.9: Example of two generated query-answer pairs by GPT using as input a query-answer pair from a Product meeting (ID ES2003b). The first row corresponds to the original query-answer pair present in the dataset. The second row corresponds to an augmented query-answer pair that is irrelevant, thus incorrect, and has to be removed from the augmented data. The third row corresponds to a valid query-answer pair generated that is relevant to the meeting.

To obtain augmented data in Table 3.9, we used GPT to generate a new query-answer pair based on an existing one. However, we encountered an issue with the ”Invalid augmented data” row, where the generated query and answer were completely unrelated to the meeting transcript. GPT generated a fictional query that theoretically could have been asked, but the answer to the query did not exist in the meeting transcript. This resulted in a fictional answer being created, rendering the data irrelevant, which is a problem that occurred recurrently when exploring this technique. However, the ”Valid augmented data” shows that this technique can sometimes be effective, as the generated query was similar to the original one, and the corresponding answer was correct. To address the issue of irrelevant data caused by text generation, we would need to monitor and filter out the produced data which is irrelevant. However, this would require human evaluation, which was not feasible for our research due to time and resource constraints. Hence, we opted not to use the text generation approach and instead employed the paraphrasing method, which ensured that the generated queries and answers were relevant to the meeting transcripts.

3.3.2.2 Implementation details

In this section, we describe the specific implementation details and hyperparameters used for paraphrasing using GPT. We chose to use OpenAI's GPT, specifically the Davinci model [6], due to its exceptional performance in natural language understanding and generation.

To ensure the quality of the paraphrased text, we have set various hyperparameters that control the generation process:

- **Model:** We used the text-davinci-003 model, which is one of the larger models in the GPT family, providing higher accuracy and fluency in the generated text.
- **Temperature:** We set the temperature parameter to 0.7, creating a balance between diversity and coherence in the model's output. As the temperature increases, the model produces more diverse and creative output.
- **Max tokens:** We limited the maximum number of tokens in the generated output to be equal to the input length to ensure that the paraphrased text and the input had similar lengths.
- **Prompt:** To ensure that the model understands our task, we experimented with various prompts and decided to append "Paraphrase the following text and provide the output in 1 line: " before the input text to provide clear instructions to the model.

3.3.3 Data Augmentation Pipeline

We employed back-translation and paraphrasing to all thee types of data of the dataset, namely meeting transcripts, queries, and answers. In this section, we provide a description of the data augmentation pipeline that we implemented to augment this data, along with a description of the review process we used to ensure that the augmented data was of good quality.

3.3.3.1 Procedure for augmenting data

Given that GPT has a limit of 2048 tokens, and the Google Translate API has a limit of 5000 characters for their input, we had to divide the meeting transcripts in chunks, as they exceeded these limits. We thus opted to take each speaker utterance to be a chunk. We noticed that the models produced better variations of the data with smaller chunks, which is why we didn't choose to divide the entire meeting transcripts into chunks with lengths that are equal to the models' limit of input tokens. Furthermore, to avoid overloading the GPT and back-translation models, we iterated through each turn of the meeting transcripts, paraphrasing the text only if it exceeded 15 characters, and refrained from sending requests for short text. After various experiments, we concluded that the text had to contain at least 15 characters for the models to produce a variation of the input.

[6] https://platform.openai.com/docs/models/overview

We applied data augmentation before the processing steps outlined in Section 3.1.4 to allow the models to make use of the entire input to generate new data, to make it more interpretable for the back-translation and GPT modules. After augmenting the data, we followed the normal preprocessing steps as described in Section 3.1.4, which included cleaning the data, tokenization, and encoding.

Using this process for both data augmentation techniques, we doubled the amount of query-answer pairs, and increased by 95% the number of characters in the meeting transcripts.

3.3.3.2 Review process

Once the augmented data was generated, we had to ensure it was of good quality and that it conveyed the same meaning as the original data, without introducing inconsistencies. Hence, we implemented a manual review process, where we randomly sampled 100 augmented sentences for each domain and manually reviewed it to ensure its quality. However, this approach doesn't completely guarantee the quality of the augmented data due to the small size of the selected sentences we had to choose for our review process, due to limited human resources.

3.3.4 Benefits and Limitations

As discussed in this section, two data augmentation techniques were employed, which both have their advantages and disadvantages. Both techniques offer the advantage of increasing the amount of training data. However, using back-translation can result in the generation of text that does not offer meaningful changes to the input's structure. On the other hand, the versatility of GPT increases the risk of generating data that is too different from the original input, as explained previously.

A comparison between the example of query-answer pair augmented using French, as shown in Table 3.7, and the example augmented using paraphrasing with GPT, depicted in Table 3.8, reveals that the back-translation method does not add much variety for this particular example. The modifications mainly consist of synonym replacements, which does not alter the sentence structures much. In contrast, the paraphrasing with GPT technique introduces new sentence structures and word choices that are not present in the original text, potentially improving the model's performance. However, this approach comes at a cost, as it requires more resources, as it is more time-consuming and costly.

In the following chapter, we will examine the results of our study and investigate how these differences between the data augmentation techniques affected our model's performance.

Chapter 4

Results And Discussion

After having described the dataset, the model and the data augmentation techniques we used along with our implementation details, we will now report the experimental results that we obtained with a thorough analysis. We begin by presenting the results we achieved by using our baseline implementation, while comparing these results with QMSum. We then present the results obtained after performing back-translation and paraphrasing using GPT, while examining the similarities and differences observed. Finally, we end this chapter with an analysis of the variety of augmented data introduced by both augmentation techniques, which contributes to explaining the differences observed in both approaches.

4.1 Baseline Experiments

In this section, we present our baseline experiments using the DYLE (Mao et al. [2022]) model on in-domain and out-of-domain settings. We compare these results with those obtained by QMSum (Zhong et al. [2021]) and show that both results show a real lack of performance in out-of-domain settings. We then proceed to a brief analysis of the unexpected results obtained in our baseline experiments, which we will explain in later sections.

Datasets	Product			Academic			Committee		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Pro.	**35.43**	**10.99**	**31.37**	22.59	3.41	19.82	24.48	3.84	21.94
Aca.	27.19	4.86	24.09	**26.69**	**4.32**	**22.58**	27.84	4.29	25.10
Com.	25.56	3.48	22.17	23.91	2.99	20.23	**32.52**	**6.98**	**27.71**

Table 4.1: QMSUM results from their paper (Zhong et al. [2021]): Performance of QMSum on the various domains using R-1, R-2, and R-L metrics. Each row represents the training set, and each column represents the test set

	Product			Academic			Committee		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Pro. (S)	**34.63**	**10.33**	**30.89**	25.59	5.25	22.48	31.77	7.22	27.87
Aca. (S)	30.93	7.67	27.51	**26.24**	**5.34**	**23.52**	31.52	6.77	27.33
Com. (S)	27.94	6.71	25.12	23.71	4.25	21.33	**32.51**	**7.73**	**28.65**
Pro. (R)	20.30	**5.34**	18.41	13.39	2.16	12.05	6.55	**3.24**	6.36
Aca. (R)	**23.44**	3.63	**21.33**	**20.25**	**2.21**	**17.88**	**8.72**	2.31	**8.1**
Com. (R)	15.05	1.93	13.65	10.00	1.04	9.11	5.58	1.93	5.21

Table 4.2: **Baseline Experiments:** In-domain and out-of-domain ROUGE scores for the Summariser (S) and Retriever (R). Each row represents the training set, and each column represents the test set.

4.1.1 Comparison with QMSum

We evaluated our baseline model across the three domains available in the QMSum dataset (Zhong et al. [2021]), namely Product, Committee, and Academic. The results obtained are shown in Table 4.2, where we used the same evaluation metrics as QMSum, namely ROUGE-1, ROUGE-2 and ROUGE-L, to ensure a thorough evaluation and comparison with QMSUM, whose results are presented in Table 4.1.

As QMSum did not provide results for their retriever for in-domain and out-of-domain settings, we were only able to compare the performance of the generated summaries. Our in-domain performance is comparable with QMSum's results, while our model largely surpasses QMSum in out-of-domain settings when tested on the Product and Committee domains, where it beats QMSum on all the ROUGE metrics. For instance, when trained on Product and tested on Committee, our model achieved R-1, R-2, and R-L scores of 31.77, 7.2, and 27.87, respectively, while QMSUM only achieved scores of 24.48, 3.84, and 21.94 for the same metrics. This represents an improvement of 30% for R-1, 88% for R-2, and 27% for R-L.

We believe this is due to the fact that QMSum trains the locator and summariser separately, which can lead to cascading errors from the locator to the summariser. In contrast, our model uses dynamic weights and jointly trains the locator and summariser, which allows optimising both the locator and the summarizer together, instead of treating them as two separate components. This change is one of the novel contributions of DYLE (Mao et al. [2022]) in their paper, where they presented the model that we use as our baseline.

4.1.2 Baseline results overview

4.1.2.1 Out-of-domain performance

Despite the improvements seen by DYLE for in-domain settings, we observe for both models significantly lower results for out-of-domain settings, as seen in Table 4.2. For instance, when trained on Committee and tested on Product, the summariser leads to R-1, R-2 and R-L scores of 27.94, 6.7 and 25.12, which are significantly lower than when tested in-domain with Committee, with scores of 32.51, 7.7 and 28.65.

These differences are not due to the variations between the Product and Committee test sets, which could be a potential reason, as proven by the fact that training on Product and testing on Product results in scores of 34.63, 10.3 and 30.89, proving that the drop in performance is due to the model's lack of domain adaptation capability. We also observe the same phenomena when evaluating the retriever with other test sets. Indeed, when training the model on Academic, the retriever obtains R-1, R-2 and R-L scores of 20.25, 2.2 and 17.88 when tested on Academic, respectively, but obtains much poorer scores when tested out-of-domain, such as with the Committee test set, obtaining scores of 8.7, 2.3 and 8.1.

4.1.2.2 Unexpected results

We observe surprising results when training our baseline model on the Academic domain, as the in-domain performance is significantly lower for the summariser than when tested out-of-domain. For instance, our model obtains an R-1 score of 26.24 for in-domain, but performs better in out-of-domain settings, achieving an R-1 score of 30.93 when tested on Product, and 31.52 when tested on Committee.

We will see in the following sections that training our model on the Academic domain, with or without data augmentation, often leads to unexpected results that don't follow the same trend as the results we obtain when training our model on Product and Committee. After performing a thorough analysis of academic, product and committee meetings, we concluded that these results are due to the overall complexity of the Academic domain. Firstly, the tone of the Academic meetings is more relaxed and conversational, while the Committee meetings are more formal and structured. Product meetings also have a more structured and goal-oriented approach, with a clear and defined goal in the dialogue utterances, with the participants working collaboratively towards achieving it. In contrast, the academic transcripts are unstructured and focus on more general and abstract discussions without a clear or defined goal.

Further analysis on the structure of meetings shows a clear difference between the Academic and Committee meetings. Table 4.3 shows the average character and word count for each speaker's utterance. On average, an utterance in an Academic meeting only has 74.58 characters, compared to 374.32 characters for Committee meetings. Hence, the Academic meetings contain many short utterances that do not convey much information, which results in the model struggling to form summaries of great quality as it fails to identify the relevant text spans that answer a specific query. However, the structure of the meetings is not the only factor causing a poor performance when testing on the Academic domain, as we see that the utterances for the Product meetings are even shorter than those in Academic meetings.

Our analysis on the character and word count would lead to suppose that the results obtained when testing on the Product domain should be poor, yet our model achieves great results in this setting. As explained previously, this is due to the fact that despite Product meetings being more informal, they are more goal-oriented, thus most of the occurrences can contain key information to form a valid summary.

Datasets	Average Character Count	Average Word Count
Product	50.79	10.52
Academic	74.58	15.89
Committee	375.34	64.49

Table 4.3: Average character and word count contained in a speaker's utterance for each domain.

4.2 Data Augmentation Experiments

In this section, we will discuss the results obtained after performing our data augmentation techniques on the datasets, namely back-translation and paraphrasing with GPT. We will begin by presenting an overview of our results. We will then proceed to discuss the differences that both of these techniques bring in our model's performance. Finally, we will further justify these differences by analysing the differences between the augmented data generated by both techniques across all the domains.

4.2.1 Results Overview

	Product			Academic			Committee		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Pro. (S)	**34.92**	**10.52**	**31.03**	26.31	**5.64**	23.04	**32.25**	**7.67**	27.27
Aca. (S)	30.87	7.61	27.40	**26.43**	5.43	**23.85**	30.81	6.74	26.75
Com. (S)	28.45	7.08	25.59	23.47	4.27	20.68	30.63	7.06	**27.44**
Pro. (R)	20.87	**5.09**	19.01	13.7	2.18	12.48	6.94	**3.33**	6.7
Aca. (R)	**22.69**	3.64	**20.30**	**18.36**	**2.21**	**16.00**	**7.64**	2.15	**7.03**
Com. (R)	15.67	2.62	14.22	10.13	1.38	9.27	4.61	1.64	4.40

Table 4.4: **Back-translation Experiments:** In-domain and out-of-domain ROUGE scores for the Summariser (S) and Retriever (R). Each row represents the training set, and each column represents the test set.

	Product			Academic			Committee		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Pro. (S)	**35.30**	**10.61**	**31.11**	26.84	5.53	23.66	30.89	6.84	27.15
Aca. (S)	30.59	7.33	27.34	**26.86**	**5.66**	**23.98**	30.12	6.47	26.52
Com. (S)	30.36	7.46	26.79	24.55	4.52	21.67	**32.69**	**8.44**	**29.29**
Pro. (R)	21.02	**4.99**	19.12	14.77	2.08	13.23	6.64	3.25	6.40
Aca. (R)	**23.21**	3.62	**20.99**	**19.50**	**2.14**	**17.37**	8.85	2.41	8.18
Com. (R)	20.33	4.20	18.42	15.6	1.95	14.14	**12.37**	**5.27**	**11.73**

Table 4.5: **Paraphrasing with GPT Experiments:** In-domain and out-of-domain ROUGE scores for the Summariser (S) and Retriever (R). Each row represents the training set, and each column represents the test set.

We applied back-translation to the training data of all three domains, namely Product, Academic and Committee, and evaluated our model on the test sets with three metrics,

namely R-1, R-2, and R-L. We reported the results achieved by the retriever and the summaries in Table 4.4 and Table 4.5 for back-translation and paraphrasing, respectively.

A major distinction we observe between the two techniques is that after paraphrasing with GPT, the best results are always achieved in the context of in-domain settings, except for one exception, for the retriever when tested on Product. However, Table 4.5 shows that this is not the case for back-translation, as for instance the best R-1 score for the Committee test set is achieved by the model trained on Product, with an R-1 score of 32.25, whereas the model trained on Committee only achieved 30.63.

We will dive into an explanation for these differences in the following section. As we progress, we will discuss various other differences in performance that require a deeper analysis.

4.2.2 Comparison between data augmentation techniques

The results obtained from paraphrasing with GPT and back-translation vary, and a closer examination of the specific improvements and changes in data they bring can explain the results obtained.

4.2.2.1 Overview

Figure 4.1 and Figure 4.2 bring us more insights regarding the variations of the ROUGE scores when applying back-translation and paraphrasing. We observe that back-translating did not bring any significant improvement on any test set, although some small improvements were made. These slight improvements do not follow a clear pattern and do not exceed a high magnitude, with the best improvement being an increase of 0.72 for the summariser when trained on Product and tested on Academic, as seen in Figure 4.1. Back-translation can lead to occasional drops in performance, which are also sometimes present with paraphrasing. In contrast, GPT shows a wider range of variations ranging from -1.4 to +6.87, showing that when paraphrasing improves the scores, it has the potential to do so in a substantial manner.

4.2.2.2 Performance variation spikes

Figure 4.2 depicts significant improvements when paraphrasing the Committee data with GPT. All three ROUGE metrics are significantly improved for the retriever, with R-1 and R-L scores increased by at least 4.5 units for all test sets, even reaching an improvement of 6.87 for the R-1 score when tested in-domain. The summariser scores also improved for all test sets, with particularly good results when tested on Product meetings, with an R-1 score improving by 2.42 units.

Hence, compared to augmenting other domains, augmenting the Committee domain had a significant impact, especially for the retriever. The key difference between the Committee meetings and the other meetings is that Committee meetings are much more lengthy, well-structured and formal. This could suggest that for the paraphrasing approach to be effective, the data aimed to be augmented needs to be well-structured and

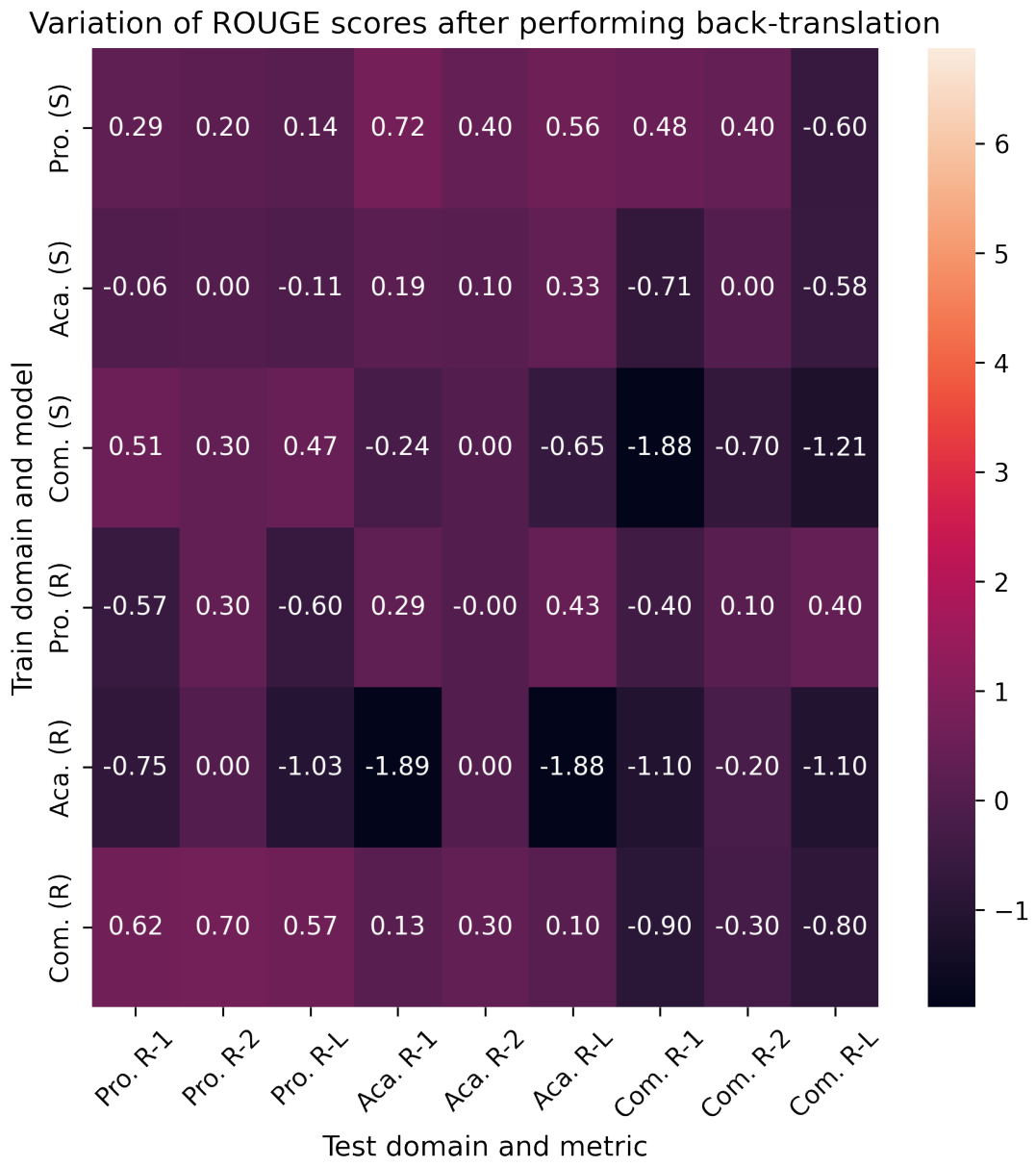

Figure 4.1: Variations for the R-1, R-2 and R-L scores between the baseline results and the results obtained after augmenting the data with **back-translation**. Each row corresponds to the train domain and the model (S for Summariser, R for Retriever). Each column corresponds to the metric (R-1, R-2, R-L) for a test domain. For instance, the top row corresponds to the variations of ROUGE scores for the Summariser when trained on Product.

more formal, rather than informal, as it is the case for Product and Academic meetings, which haven't seen a significant improvement in their scores.

The results indicate the both in-domain and out-of-domain improved using the paraphrasing approach, especially for the Committee data. However, this does not necessarily mean that this technique reduced the gap in scores between in-domain and out-of-domain evaluation. Indeed, in order to further analyse the variation of the gap between

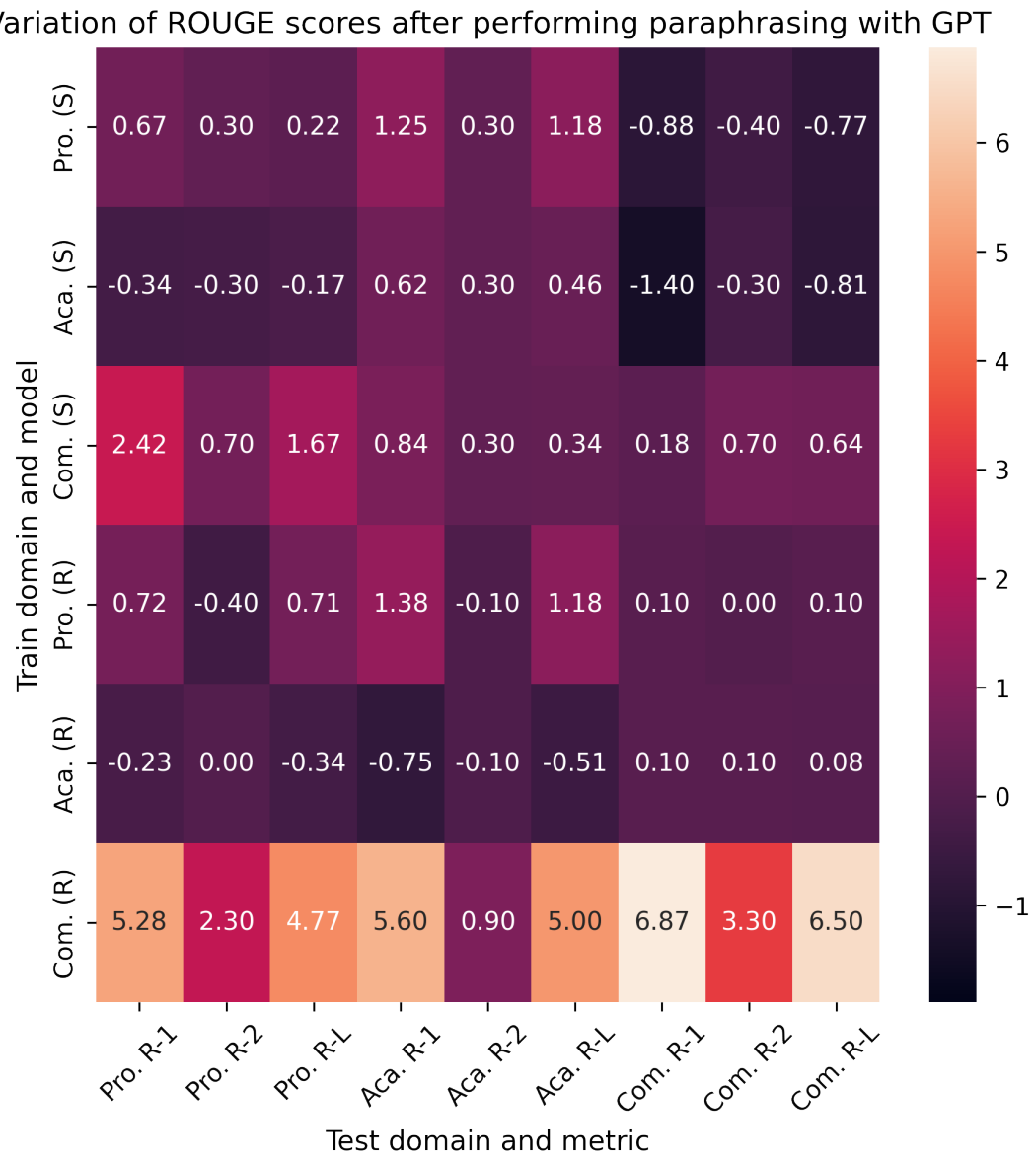

Figure 4.2: Variations for the R-1, R-2 and R-L scores between the baseline results and the results obtained after augmenting the data using **paraphrasing with GPT**. Each row corresponds to the train domain and the model (S for Summariser, R for Retriever). Each column corresponds to the metric (R-1, R-2, R-L) for a test domain. For instance, the top row corresponds to the variations of ROUGE scores for the Summariser when trained on Product.

in-domain and out-of-domain performance, we direct our attention to Figure 4.3, which shows for example that the gap between in-domain and out-of-domain performance decreased by 50% when the summariser is evaluated on Product. These results indicate that data augmentation using paraphrasing can improve both in-domain and out-of-domain performance, while reducing the gap between them.

It is worth noting that the same visualisations illustrating the variations in the out-of-domain performance gap for the Product and Academic domains can be found

in the Appendix A.1. We do not include these results here as they did not provide any meaningful information, considering that as concluded now, only augmenting the Committee data resulted in a significant improvement in out-of-domain performance.

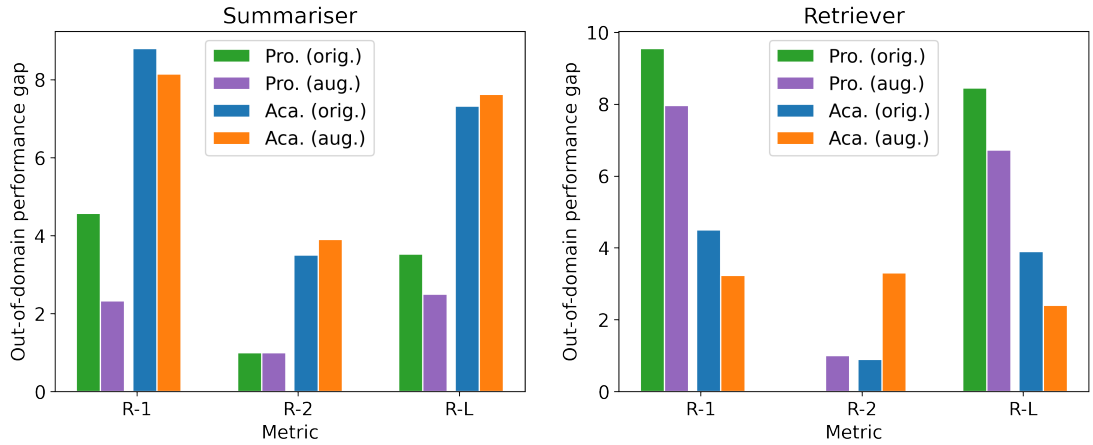

Figure 4.3: Comparison of the out-of-domain gap before and after performing **paraphrasing with GPT** when trained on **Committee** for the summariser (left) and the retriever (right). Each bar represents the magnitude of the difference between the score achieved by Committee (in-domain test set) and the corresponding out-of-domain test sets (Product and Academic).

Looking back at Figure 4.2, we also notice that when there is a significant improvement, the spikes are mostly limited to a single test set, except for the Committee retriever, which increased for all test sets and metrics, as discussed previously. For instance, augmenting the *Product* data particularly improved its performance when tested on the *Academic* set. Similarly, augmenting the *Committee* domain had a significant impact on the *Product* test set for the summarizer.

These observations show that augmenting the Product data made its summariser perform better on the Academic test set, and vice versa. Augmenting both of these train sets also led to a significant decrease in the performance when evaluated on the Committee domain. As explained earlier, Academic and Product share the characteristic that they are informal and contain more relaxed conversations with short dialogue utterances, which is not the case for Committee. Hence, this shows that augmenting a domain that contains an informal structure leads to an improvement of out-of-domain scores for domains that also contain informal meetings. They however perform worse on domains that contain formal discussions, probably because the model has not seen this kind of meeting during training and has overfitted to informal meetings.

However, augmenting Committee (which contains formal meetings) does not lead to a drop in performance when evaluated on Product and Academic. It even considerably improves the ROUGE scores for the Product set. This shows that augmenting a domain with formal meetings containing long utterances still improves out-of-domain performance, even for informal meetings. However, the opposite is not true. This shows a crucial conclusion in our study, which is that the data that is chosen to be augmented needs to be carefully selected in order for the paraphrasing augmentation technique to

be successful.

4.2.3 Variety of augmented data

Our results indicate that back-translation did not bring any noticeable improvement at all. In contrast, paraphrasing our data with GPT has proved to improve both in-domain and out-of-domain performance. The reason for these observations lies in the diversity introduced by the augmentation techniques, which can be visualised by plotting the UMAP projections onto a 2D space through dimensionality reduction of the original text from the training set, and compare these projections with the augmented text.

These UMAP projections are presented in Figure 4.6, which shows the projections of the training text from the Committee domain, as well as in Figure 4.4 and Figure 4.5 for the Product train set, and Academic train set, respectively. Analysing these figures allows us to visualise the degree of similarity between the augmented data and the original data.

These figures indicate that for all three domains, the augmented text using back-translation is very similar to the original text, as all the data points are really close to each other. In contrast, when visualising the similarity between the original data and the augmented data using paraphrasing, we observe two clusters for Figure 4.4 (Product) and Figure 4.5 (Academic). We also see in Figure 4.6 (Committee) that the augmented text produced by GPT provides more diversity than when augmented with back-translation, although the clusters are less distinct than for the Product and Academic train sets.

These observations show that GPT introduced new variations to the training data, as it is capable to generate new sentences that convey the same meaning but with different structures. The addition of these variations to the training data made the model more capable to handle new variations of data, hence being able to generate more accurate summaries on meetings from other domains.

These figures also bring an explanation towards some results observed in Figure 4.1, where we see that the in-domain performance is the one that decreased the most for all domains when applying back-translation. Indeed, this is the case for the summariser trained on Committee, and the retrievers when trained on Product, Academic and Committee. After analysing the UMAP projections for these domains, we can explain these drops for in-domain performance: because the augmented data with back-translation is very similar to the original text, it causes the corresponding summariser or retriever to overfit on the train set as we are essentially doubling the train set size with almost the exact same data.

Regarding back-translation, we can already conclude from these results that it is not a suitable technique to improve the in-domain or out-of-domain performance for query-based meeting summarisation models, if used within the same conditions as in our study. However, a limitation can immediately be defined regarding our work on back-translation: we have only augmented the data once, using French. The lack of diversity introduced by our usage of back-translation could be explained by the fact that English and French have a really close linguistic distance. Back-translating our data using

Japanese, for instance, could result in a greater diversity of training data, as Japanese has a more distinct grammar structure, vocabulary, and syntax compared to English.

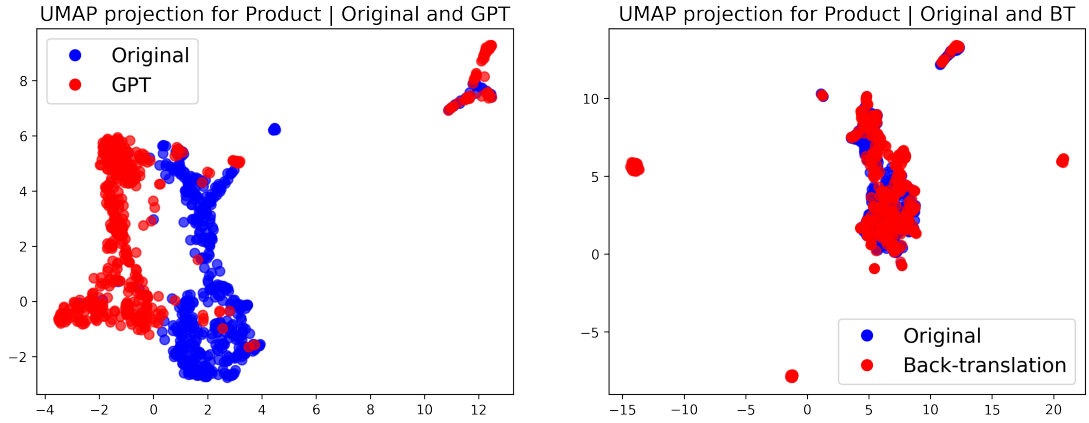

Figure 4.4: UMAP projection on a 2D space of original and augmented sentences contained with the **Product** train set using paraphrasing with GPT (left) and back-translation (right) encoded with RobertaTokeniser (used by our locator).

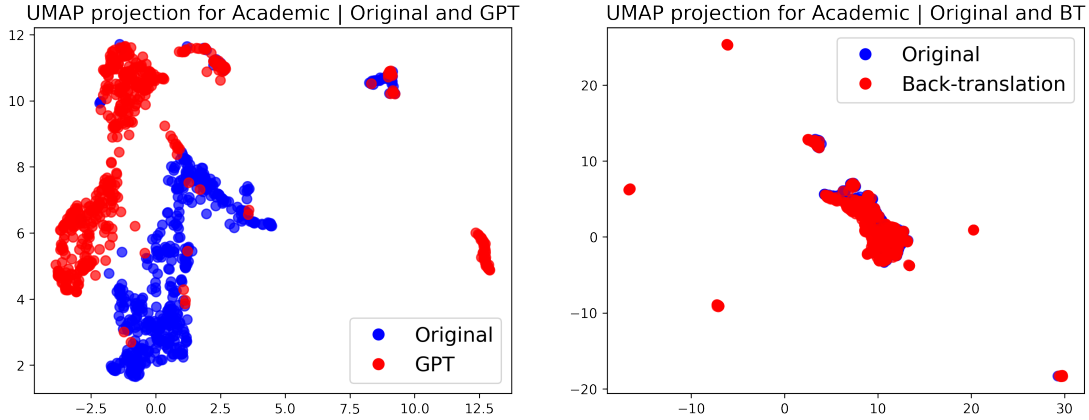

Figure 4.5: UMAP projection on a 2D space of original and augmented sentences contained with the **Academic** train set using paraphrasing with GPT (left) and back-translation (right) encoded with RobertaTokeniser (used by our locator).

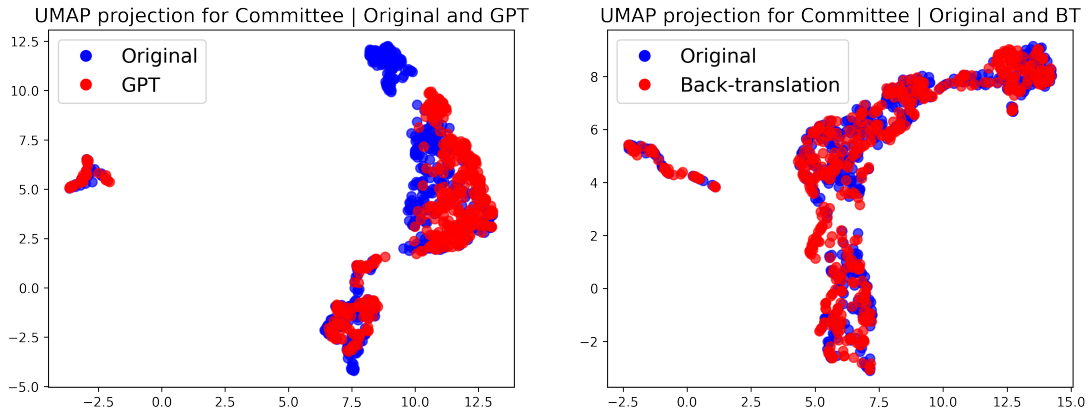

Figure 4.6: UMAP projection on a 2D space of original and augmented sentences contained with the **Committee** train set using paraphrasing with GPT (left) and back-translation (right) encoded with RobertaTokeniser (used by our locator).

Chapter 5

Conclusions

5.1 Summary

In this thesis, we have explored the limitations in out-of-domain performance encountered by existing models for the query-based meeting summarisation task. We conducted our research using QMSum (Zhong et al. [2021]), a benchmark dataset that contains meetings associated with query-answer pairs for three domains, namely Product, Academic and Committee. Using DYLE (Mao et al. [2022]) as our baseline model, which already showed outstanding results on the QMSum dataset, we investigated whether applying data augmentation techniques to the existing data could improve the in-domain and out-of-domain performance of the model.

We chose two data augmentation techniques which were the most relevant for our task according to our analysis, namely back-translation with French, and paraphrasing with GPT (Floridi and Chiriatti [2020]).

Our findings indicate that performing back-translation with French did not introduce enough variety to training data to allow our model to show any improvements. We indeed showed that the data augmented using back-translation was very similar to the original data, thus rendering the data augmentation process inefficient. We however suspect that this can be due to the choice of the foreign language used, namely French. Using a foreign language that differs vastly from English could allow the augmented text to be sufficiently altered to have an impact on the performance of the model. Further work would have to be performed in order to definitively conclude whether back-translation is indeed an inefficient technique for query-based meeting summarisation.

On the other hand, our experiments showed that using a pre-trained GPT model for generating paraphrased versions of the data led to significant improvements for both in-domain and out-of-domain settings. We showed that the resulting augmented data contained new sentence structures and a new vocabulary, which resulted in the model being able to handle a wider variety of data, leading to better performance when being tested on unseen data from other domains that differ from the domain's data it was trained on.

Furthermore, our results indicate that augmenting the data using the GPT paraphras-

ing approach was particularly successful on a specific domain, namely Committee. Committee meetings are much more structured, more formal and contain much longer dialogue utterances than meetings from other domains. This led us to understand the key characteristics in a domain-specific dataset to take into consideration when performing data augmentation on it. Indeed, if the data is formal and structured with lengthy content, then augmenting it gives better performance when evaluating on both informal and formal meetings. The opposite is however not true, as augmenting unstructured and informal data didn't lead to an improvement in neither formal nor informal data.

We also observed that paraphrasing the data using GPT affected the performance of the retriever more than the summariser. This can suggest that the problem with out-of-domain performance is mainly related to the retriever. A limitation to our approach is that to validate this hypothesis, we would need to test the summariser on gold spans.

In conclusion, our paper shows that back-translation did not bring any improvements, compared to GPT which has shown to be a powerful tool that can improve both in-domain and out-of-domain performance when applying it to carefully selected data. The improvements we observed with GPT can lay ground to further work using this model.

5.2 Future Work

For future work, we suggest further exploring the capabilities of GPT for query-based meeting summarisation. Using the text generation approach instead of paraphrasing could bring even more variety to the augmented data, but this would require a heavy manual review process of the augmented data, as explained in our paper. Furthermore, one could consider using GPT for the retriever and summariser, instead of RoBERTa and BART. The performance of the resulting model would then need to be evaluated and used to assess the trade-offs of this technique. Indeed, this would result in a much larger model that would require large resources to train and use, whereas the approach we offered can be used on relatively small hardware. In addition, our approach is more interpretable and precise components can be changed for different needs, which would not be as feasible with a model using GPT for the locator and summariser. Finally, there would also be some privacy and confidentiality concerns when using GPT for predictions, as organisations tend to be protective with their meeting transcripts.

Another avenue for future work is to investigate alternative evaluation metrics that better capture the nuances of meeting summarization. While automatic evaluation metrics such as ROUGE have been widely used, they may not always accurately reflect the quality of a summary. Human evaluation can provide a more comprehensive assessment of the summary quality (Lloret et al. [2018]), but is more time-consuming and resource-intensive.

We also suggest exploring the use of domain adaptation techniques to further improve out-of-domain performance. One approach could be to use transfer learning to leverage knowledge from a pre-trained model for a specific domain to improve the performance of a model in another domain (Kouw and Loog [2018]). Another approach would be to use domain adversarial training, which involves training the model to learn features

that are domain-invariant, potentially leading to a better generalisation across domains (Ganin et al. [2016]). Additionally, unsupervised domain adaptation techniques such as domain adaptive pre-training can be used, where a model is pre-trained on a large amount of data from different domains and then fine-tuned on the target domain data (Wu et al. [2021]). Finally, we could explore the use of ensemble methods to combine models trained on different domains to improve overall performance (Nozza et al. [2016]), with the training data containing much more variety than if it was trained on a single domain.

Bibliography

Dininta Annisa and Masayu Leylia Khodra. Query-based summarization for indonesian news articles. In *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, pages 1–6, 2017. doi: 10.1109/ICAICTA.2017.8090959.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Djamila Romaissa Beddiar, Md Saroar Jahan, and Mourad Oussalah. Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*, 24:100153, 2021. ISSN 2468-6964. doi: https://doi.org/10.1016/j.osnem.2021.100153. URL https://www.sciencedirect.com/science/article/pii/S2468696421000355.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. The ami meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005, Edinburgh, UK, July 11-13, 2005, Revised Selected Papers 2*, pages 28–39. Springer, 2006.

Wayne F Cascio. Managing a virtual workplace. *Academy of Management Perspectives*, 14(3):81–90, 2000.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Matan Eyal, Tal Baumel, and Michael Elhadad. Question answering as an automatic evaluation metric for news article summarization, 2019. URL https://arxiv.org/abs/1906.00318.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp, 2021a.

Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng. Dialogue discourse-aware graph model and data augmentation for meeting summarization, 2021b.

Cut Fiarni, Herastia Maharani, and Rino Pratama. Sentiment analysis system for

indonesia online retail shop review using hierarchy naive bayes technique. In *2016 4th International Conference on Information and Communication Technology (ICoICT)*, pages 1–6, 2016. doi: 10.1109/ICoICT.2016.7571912.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Apar Garg, Saiteja Adusumilli, Shanmukha Yenneti, Tapas Badal, Deepak Garg, Vivek Pandey, Abhishek Nigam, Yashu Kant Gupta, Gyan Mittal, and Rahul Agarwal. News article summarization with pretrained transformer. In Deepak Garg, Kit Wong, Jagannathan Sarangapani, and Suneet Kumar Gupta, editors, *Advanced Computing*, pages 203–211, Singapore, 2021. Springer Singapore. ISBN 978-981-16-0401-0.

Jan Gerretzen, Ewa Szymanska, Jeroen J Jansen, Jacob Bart, Henk-Jan van Manen, Edwin R van den Heuvel, and Lutgarde MC Buydens. Simple and effective way for data preprocessing selection based on design of experiments. *Analytical chemistry*, 87(24):12096–12103, 2015.

Sandesh Gharatkar, Aakash Ingle, Tanmay Naik, and Ashwini Save. Review preprocessing using data cleaning and stemming technique. In *2017 international conference on innovations in information, embedded and communication systems (iciiecs)*, pages 1–4. IEEE, 2017.

Dan Gillick, Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tur. A global optimization framework for meeting summarization. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4769–4772, 2009. doi: 10.1109/ICASSP.2009.4960697.

Abram Handler and Brendan O'Connor. Rookie: A unique approach for exploring news archives, 2017.

Ralivat Haruna, Afolayan Obiniyi, Muhammed Abdulkarim, and A.A. Afolorunsho. Automatic summarization of scientific documents using transformer architectures: A review. In *2022 5th Information Technology for Education and Development (ITED)*, pages 1–6, 2022. doi: 10.1109/ITED56637.2022.10051602.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE, 2003.

Sanjeev Kumar Karn, Francine Chen, Yan-Ying Chen, Ulli Waltinger, and Hinrich Schuetze. Few-shot learning of an interleaved text summarization model by pretraining with synthetic data, 2021.

P Kathiravan and N Haridoss. Preprocessing for mining the textual data-a review. *vol*, 7:5–8, 2018.

Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. Aquamuse: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint arXiv:2010.12694*, 2020.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3609–3619, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1363. URL https://aclanthology.org/N19-1363.

Moreno La Quatra and Luca Cagliero. End-to-end training for financial report summarization. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 118–123, Barcelona, Spain (Online), December 2020. COLING. URL https://aclanthology.org/2020.fnp-1.20.

Jae-Kul Lee, Hyun-Je Song, and Seong-Bae Park. Two-step sentence extraction for summarization of meeting minutes. In *2011 Eighth International Conference on Information Technology: New Generations*, pages 614–619, 2011. doi: 10.1109/ITNG.2011.210.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J Radke. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, 2019.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

Jian Liu, Yufeng Chen, and Jinan Xu. Document-level event argument linking as machine reading comprehension. *Neurocomputing*, 488:414–423, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2022.03.016. URL https://www.sciencedirect.com/science/article/pii/S0925231222002867.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Elena Lloret, Laura Plaza, and Ahmet Aker. The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, 52:101–148, 2018.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed H Awadallah, and Dragomir Radev. Dyle: Dynamic latent extraction for abstractive long-input summarization. In *ACL 2022*, 2022.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization, 2020.

K. McKeown, J. Hirschberg, M. Galley, and S. Maskey. From text to speech summarization. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v/997–v1000 Vol. 5, 2005. doi: 10.1109/ICASSP.2005.1416474.

Derek Miller. Leveraging bert for extractive text summarization on lectures, 2019.

Ahmed A Mohamed and Sanguthevar Rajasekaran. Improving query-based summarization using document graphs. In *2006 IEEE international symposium on signal processing and information technology*, pages 408–410. IEEE, 2006.

Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. 2005.

Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233, 2011. ISSN 1554-0669. doi: 10.1561/1500000015. URL `http://dx.doi.org/10.1561/1500000015`.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2):4–es, 2007.

Debora Nozza, Elisabetta Fersini, and Enza Messina. Deep learning and ensemble methods for domain adaptation. In *2016 IEEE 28th International conference on tools with artificial intelligence (ICTAI)*, pages 184–189. IEEE, 2016.

Vedant Parikh, Vidit Mathur, Parth Mehta, Namita Mittal, and Prasenjit Majumder. Lawsum: A weakly supervised approach for indian legal document summarization, 2021.

Matt Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks, 2017.

Sai Ashish Somayajula, Linfeng Song, and Pengtao Xie. A Multi-Level Optimization Framework for End-to-End Text Augmentation. *Transactions of the Association for Computational Linguistics*, 10:343–358, 04 2022. ISSN 2307-387X. doi: 10.1162/tacl_a_00464. URL `https://doi.org/10.1162/tacl\_a\_00464`.

Jared Spataro. The future of work—the good, the challenging & the unknown. *https://www. microsoft. com/en-us/microsoft-365/blog/2020/07/08/future-work-good-challenging-unknown*, 2020.

Lu Wang and Claire Cardie. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, 2013.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019.

Han Wu, Kun Xu, Linfeng Song, Lifeng Jin, Haisong Zhang, and Linqi Song. Domain-adaptive pretraining methods for dialogue understanding. *arXiv preprint arXiv:2105.13665*, 2021.

Liwen Xu, Yan Zhang, Lei Hong, Yi Cai, and Szui Sung. ChicHealth @ MEDIQA 2021: Exploring the limits of pre-trained seq2seq models for medical summarization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 263–267, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.bionlp-1.29. URL https://aclanthology.org/2021.bionlp-1.29.

Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. Emailsum: Abstractive email thread summarization, 2021.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. Qmsum: A new benchmark for query-based multi-domain meeting summarization, 2021.

Appendix A

Additional result visualisations

A.1 Out-of-domain performance gap variations

This section contains figures illustrating the out-of-domain performance gaps before and after applying paraphrasing with GPT and back-translation. These visualisations can however be misleading. For instance, Figure A.3 shows that the out-of-domain performance gap decreased for the summariser. This in indeed the case, but it is not due to back-translation improving out-of-domain performance, as it is due to back-translation worsening the in-domain performance, with no significant improvement for out-of-domain. Hence, by definition, the out-of-domain performance gap decreased.

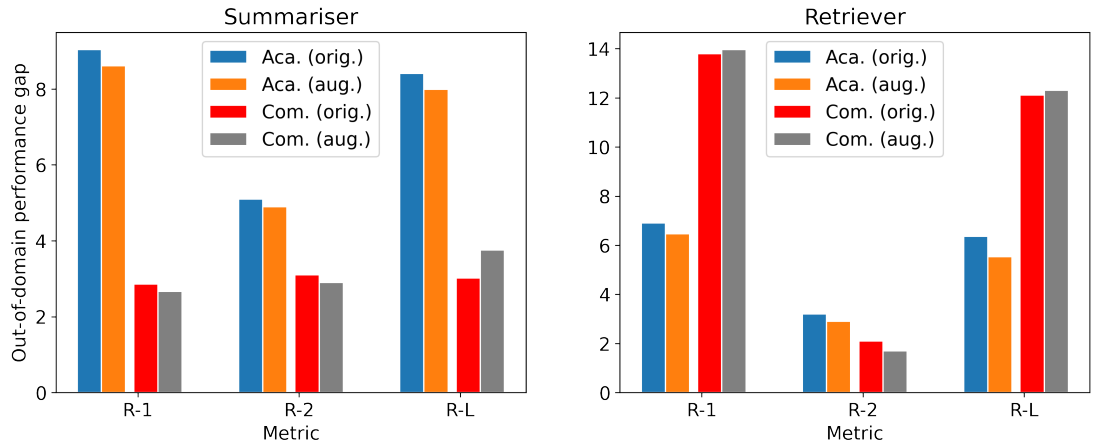

Figure A.1: Comparison of the out-of-domain gap before and after performing **back-translation** when trained on **Product** for the summariser (left) and the retriever (right). Each bar represents the magnitude of the difference between the score achieved by Product (in-domain test set) and the corresponding out-of-domain test sets (Academic and Committee).

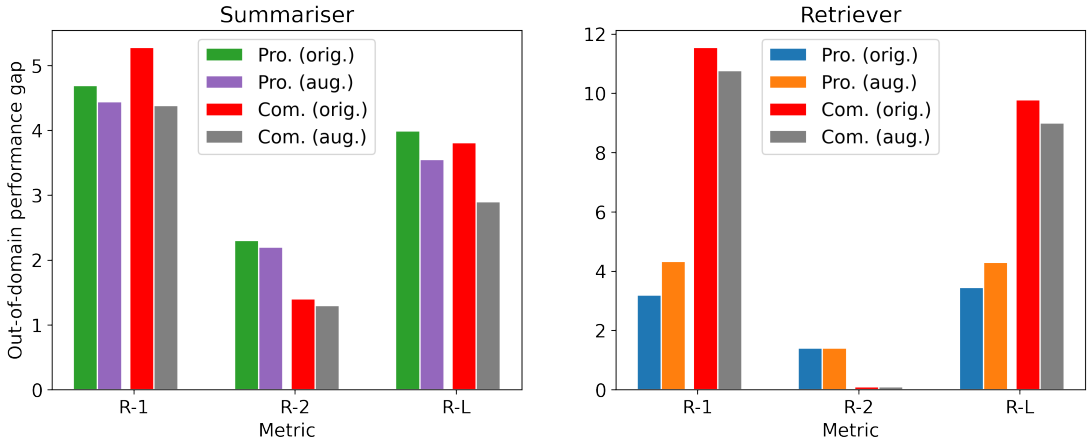

Figure A.2: Comparison of the out-of-domain gap before and after performing **back-translation** when trained on **Academic** for the summariser (left) and the retriever (right). Each bar represents the magnitude of the difference between the score achieved by Academic (in-domain test set) and the corresponding out-of-domain test sets (Product and Committee).

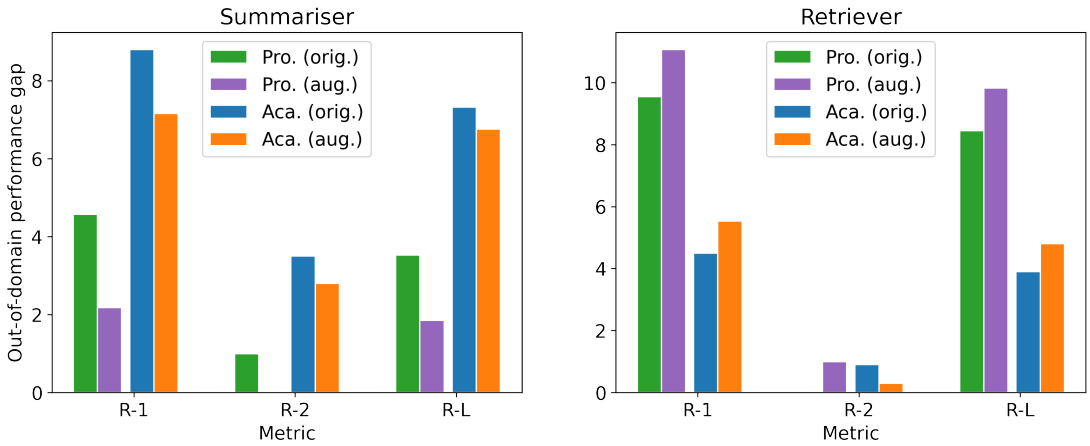

Figure A.3: Comparison of the out-of-domain gap before and after performing **back-translation** when trained on **Committee** for the summariser (left) and the retriever (right). Each bar represents the magnitude of the difference between the score achieved by Committee (in-domain test set) and the corresponding out-of-domain test sets (Product and Academic).

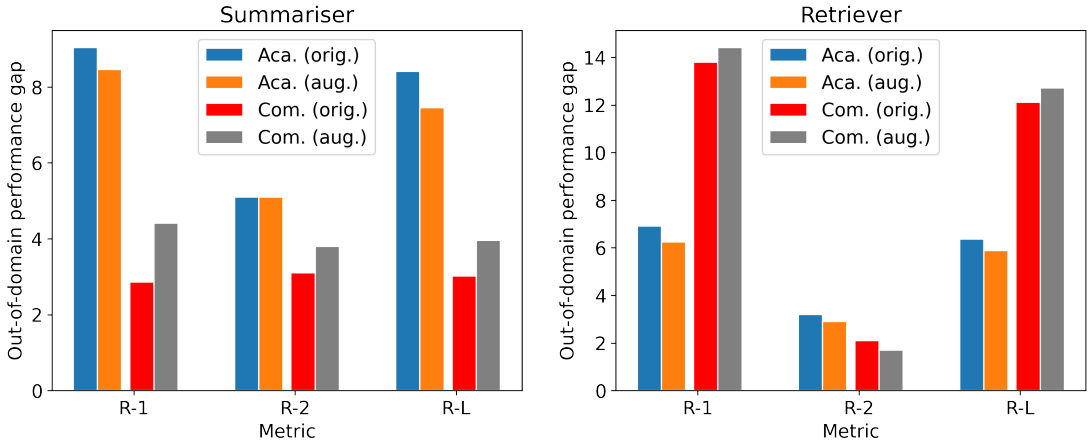

Figure A.4: Comparison of the out-of-domain gap before and after performing **paraphrasing with GPT** when trained on **Product** for the summariser (left) and the retriever (right). Each bar represents the magnitude of the difference between the score achieved by Product (in-domain test set) and the corresponding out-of-domain test sets (Academic and Committee).

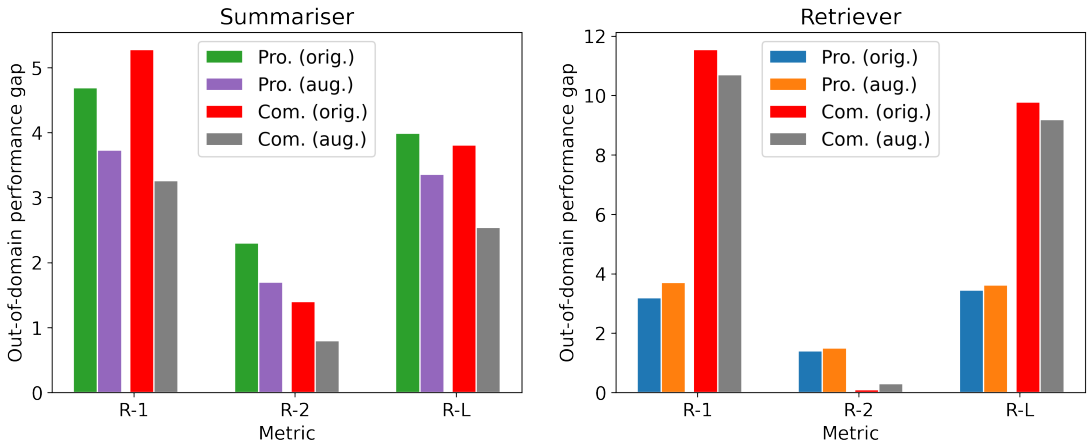

Figure A.5: Comparison of the out-of-domain gap before and after performing **paraphrasing with GPT** when trained on **Academic** for the summariser (left) and the retriever (right). Each bar represents the magnitude of the difference between the score achieved by Academic (in-domain test set) and the corresponding out-of-domain test sets (Product and Committee).