Author:
Xu, Zhaozhen

Title:
Processing questions with multi-task sentence embedding

Processing Questions with Multi-task Sentence Embedding

By

ZHAOZHEN XU

Department of Computer Science

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

DECEMBER 2022

Word count: 35595

Abstract

Deep neural networks were widely applied to understanding human language in the past decade. In order to let the machine build a better comprehension of text, researchers created various tasks to train the algorithms and language models. The machine was trained to understand articles, emails, and even shorter text like Tweets. With the development of online questioning communities and voice assistants, we notice a new type of text data coming into natural language processing research: Questions. Analysing questions can provide a new perspective on understanding communities and people's interests. Furthermore, while raising different kinds of questions, people also seek similar questions asked by others and answers. Thousands of questions are asked on a daily basis. With this large amount of data collected, it is time-consuming to analyse all the questions by a human. It will be helpful if a model can automatically process the questions.

In this thesis, we focus on training and fine-tuning Bidirectional Encoder Representations from Transformers (BERT), one of the transformer-based neural networks that created leading performance in many language comprehension tasks as well as generating sentence representation in a high dimensional vector space. To understand how BERT projects the sentences into vectors, we investigate the composition of the sentence representations generated by BERT with a simple sentence corpus: each sentence comprises only a subject, verb, and object. We decompose the sentence embedding into its attributes: words. The attribute representations can be used to predict the representation of an unseen sentence.

We also train a question-specialised version of BERT (QBERT) to become a "generalist" for analysing the questions in three different tasks: question topic classification, equivalent question detection, and question answering. QBERT is trained as a "generalist" based on multi-task learning techniques. The idea of a "generalist" is that QBERT should deliver good performance on a range of tasks involving questions in natural language, trading off some accuracy for robust performance across the various tasks. An important challenge is if labelled data from one task can be beneficial in fine-tuning the internal representations of QBERT, also when used on a different task.

With one version of QBERT we trained, QBERT-RR, we propose a pipeline for processing questions and apply it to analyse a new question corpus collected by "We The Curious", a science centre located in Bristol. There are more than 10,000 questions collected. Researching these questions has the potential to help the museum understand the visitors' curiosity and create exhibitions or educational content that are more relevant to their interests and lives. On top of that, QBERT-RR also answers some of the questions with one-sentence answers using a large-scale external knowledge source.

Dedication and acknowledgements

First of all, I would like to thank my supervisor Professor Nello Cristianini for his guidance. Nello has always been supportive, encouraging, and passionate about my research. I highly value his insight and help throughout my PhD journey. Moreover, I would like to thank my reviewer Dr Raul Santos-Rodriguez for his precious advice since my Master's studies in 2017. My appreciation is also due to the rest of the team working with me in the Intelligent Systems Lab at the University of Bristol, Nouf Bindris, Adam Sutton, Sheng Wang, and Zhijin Guo.

I was fortunate to access the question dataset collected by "We The Curious" science centre. The questions they shared with me motivated this research in the first place, which is processing the questions. Thank you to Dr Amelia Howarth and Nicole Briggs for helping while analysing the questions and writing the research papers with me. I also acknowledge Anna Starkey for her central role in "Project What If" and for being part of the team who initiated this research, and Professor David May for bringing us all together and providing his insight along the way.

A special thank goes to my best friends, Bingjie Yu and Kyriakos Georgiou, who are also my peer mentors. Thank you for being so patient while listening to my struggles and confusion, as well as sharing your experience with me for both research and life. I appreciate all the fruitful conversations and memorable times we had together.

Furthermore, I am very grateful for having all my research buddies: Hongbo Bo, Yuhang Ming, Yuexiao Liu, Zeynel Samak, Burak Akyol, and Benjamin Arana Sanchez, at the University of Bristol. We started our PhD journey together back in 2018 and went through all the ups and downs together in the past four years. I appreciate them for being there and letting me know I was never alone on this journey.

I would like to thank all my friends in Bristol. Bingyu Zhao, Stella Karagyaurova, Petros Maliotis, Yi Li, Andrej Turk, Yiguo Qiao, and many others, thank you for enlightening my life in a city that is 5,790 miles away from my home. Also, I am grateful to my friends back home, Yifan Li, Yucheng Tang, Yihe Zhang, Ziwei Zhou, and Tuzi. Even though we no longer live on the same continent and have a massive time difference, you still send me your support as always.

I want to send my appreciation to George Paschalis, who gives me lots of joy and support during my PhD. He inspires me to aim higher and face challenges all the time. This path would be much more arduous without him.

Most importantly, my deepest gratitude goes to my family, especially my parents, who have given me unconditional love and unlimited support from the first day I came to this world. Without them, I would not have had a chance to pursue my study in the UK, and complete all this work. It has been really difficult for my family and me not to visit each other in the past three years due to the pandemic. I hope I will be able to go home soon after finishing my studies.

Last but not least, I would like to dedicate this thesis to my grandpa, the best mathematician in my mind. He introduced me to the fantastic world of maths. And the fun fact is that some of the maths my grandpa taught me when I was a child are still being used in this thesis.

LIST OF PUBLICATIONS

The contribution of the work presented in this thesis has been recognised through peer-reviewed publications:

1. Xu, Zhaozhen, Zhijin Guo, and Nello Cristianini. "On Compositionality in Data Embedding." In Advances in Intelligent Data Analysis XXI: 21st International Symposium on Intelligent Data Analysis, IDA 2023, Louvain-la-Neuve, Belgium, April 12,Äì14, 2023, Proceedings, pp. 484-496. Cham: Springer Nature Switzerland, 2023.

2. Xu, Zhaozhen, and Nello Cristianini. "QBERT: Generalist Model for Processing Questions." In Advances in Intelligent Data Analysis XXI: 21st International Symposium on Intelligent Data Analysis, IDA 2023, Louvain-la-Neuve, Belgium, April 12,Äì14, 2023, Proceedings, pp. 472-483. Cham: Springer Nature Switzerland, 2023.

3. Xu, Zhaozhen, Amelia Howarth, Nicole Briggs, and Nello Cristianini. "What Makes us Curious? Analysis of a Corpus of Open-Domain Questions." In CS & IT Conference Proceedings, vol. 11, no. 20. CS & IT Conference Proceedings, 2021.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

Table of Contents

LIST OF TABLES

List of Figures

Chapter 1

Introduction

Text is one of the most important mediums for humans to learn and communicate. We record most our knowledge with text and build relationships with others by expressing our feelings and exchanging our thoughts using language. For a computer to understand humans, naturally, we expect it to comprehend language as we speak and write it. This can enable more in-depth human-machine interaction especially in situations where information is difficult to process by a human. Teaching a computer how to understand our language and process text, is where natural language processing starts; by taking real-world input and processing it. With natural language processing, computers understand articles, books, and even shorter text like social media posts in different languages. Beyond the fact described in the text, computers can identify sentiments, concerns, activities, values, etc. The ability to understand text and language also results in many well-known applications such as machine translation, text classification, information retrieval, and chatbot/virtual assistants.

The development of the Internet and online communities has contributed to the information explosion. This has created the pressing need to process this ever-increasing information with an efficient and powerful tool. Deep learning neural networks have been widely applied in the past decade to extract and process information from large amounts of text. Some state-of-the-art deep learning models like Transformer [148] are widely used in building language models and have resulted in leading performance for various language understanding tasks [43, 118, 168].

While investigating all these applications and natural language processing models, we noticed a growing type of text data: the Questions. With the rise of commercial voice assistants such as Siri and Alexa and communities such as Quora and Stack Overflow, numerous questions are being asked every day. People seek answers or information by asking questions. However, we came to understand that the value of the “Question” is way beyond the limit of getting an answer. Processing questions can provide a new perspective on understanding communities and people's interests. In the meanwhile, how to process this expanding number of questions has become a critical challenge. Training a deep neural network can become the solution to processing questions generated every day without human intervention.

The first step for processing questions is converting the text into a format the computer can handle. As a result, we need to model the text in a numeric way. Ideally, the numeric representation should contain some information within the text, such as semantic and syntactic information. Mathematical modelling is a process of defining and solving real-world problems with precise mathematical terms. It creates a bridge between mathematics and the world and has been widely used in different fields, such as physics, engineering, and social science. With modelling, people can represent items with mathematical notations and solve problems in real scenarios with mathematical tools. For example, representing words for a computational model to process the human language. A good representation should capture enough information to solve the problem.

One approach to model a set of items with their relations, is to represent those items as points in a vector space. By representing items as vectors, we can present the relations of interest between those items by geometric or algebraic relations between those vectors of coordinates. This process of modelling the items is also known as "Embedding" in machine learning. Supervised information tailors the learned embedding for the desired task. For example, we may decide that a set of documents should be embedded in a space in such a way that the distance between vectors reflects their content similarities. Since the questions are mostly formed in a sentence, we are looking for a sentence embedding method that can represent the meaning of the question as well as the relations with other questions and contents.

The development of neural networks results in a better way to representing sentences and creates leading performance in terms of mapping sentences with a similar meaning closer to each other in vector space. With this approach, the values of each coordinate are not considered to be meaningful, so they cannot be used to interpret what information is being used by the artificial intelligence (AI) system. This considerably limits the possibility of explaining the decisions of a system. Understanding the properties within the embedding has the potential to help with various challenges of modern AI, such as explaining AI decisions based on these embeddings, and the possibility of performing analogical reasoning or counterfactual question answering.

One particularly interesting phenomenon that we observe in the case of word embeddings is that after enforcing certain specific relations, we can observe that other relations can also be recovered from that representation. When we request that the embedding reflects "co-occurrence" relations between words, we also observe that certain semantic and even syntactic relations are represented in a predictable manner, typically in an additive form. In one sense, this may allow performing analogical reasoning (Berlin is to Germany as Paris is to France). In another sense, this may be regarded as a form of "compositionality" (Berlin = Germany + capital; Paris = France +

capital). This allows for certain types of inference to be performed (for example, the classic: queen - king = waitress - waiter [102], and the consequent possibility to perform analogical inference). The presence of biases, such as gender bias, has also been observed in the same representation. Both of those can be explained through the "distributional" assumption, which is the idea that the meaning of a word depends on the statistical distribution of words related to it.

The discussion of the distributional structure and meaning can be dated back to the 1950s when Harris [62] discovered a “parallel meaning structure”. He proposed that the correlation between language distribution and its meaning is much greater with respect to the context of discourse. In other words, words in a similar context tend to have similar semantic meanings. The word embedding space learns to represent a word with its distributional context [126], and certain properties of the distributional context can be preserved, such as semantic meaning, syntactic information, topic information, etc. The challenge arises: After projecting the sentence into a vector space, do sentences with similar meaning tend to be close to each other in the vector space? Moreover, is any kind of distributional context preserved in a well-learned sentence embedding?

In this thesis, we try to decompose sentence embedding with a linear system. Decomposing the embedding helps us investigate the properties that build sentence embeddings.

We focus on investigating the representation generated by one of the state-of-the-art pre-trained language models: BERT (stands for Bidirectional Encoder Representations from Transformers) [43]. A language model is a model that calculates the probability distributions for sequences of words [73]. It usually requires a large scale of data and massive computational power during the training process. As a result, many state-of-the-art models are pre-trained with general text corpora, such as Wikipedia and books, and they show the capability of learning and understanding language. The model that learns the language can then be saved and fine-tuned for other specific tasks.

Despite BERT being pre-trained on large-scale unsupervised tasks, it still requires lots of training data to achieve optimal performance on a specific task. This is due to the fact that language models based on deep neural networks usually contain millions or even billions of parameters. Moreover, the labelled data required for fine-tuning the language model are always limited and hard to obtain. Thus, it is usually expensive to train and use a separate network for each task. The challenge is if one trained network can perform multiple tasks. To overcome this, we use multi-task learning, a learning approach that improves generalisation by adding related tasks and domain information during the training [25].

By applying multi-task training, we intend to build a “generalist” who can solve multiple tasks rather than a “specialist” who is only trained to maximise the performance on one specific task. While training a single-task model, more data result in better performance. However, one of the main challenges in machine learning is that there is never enough data. Multi-task learning enables the machine to accomplish better performance by using the (limited) data from different tasks.

Natural language understanding related tasks come in a wide variety. For example, single sentence classification tasks like sentiment analysis, pairwise classification tasks like natural language inference, and regression tasks like sentence similarity. Research shows that using one model to learn all these tasks can improve model generalisation and performance [92]. As a result, we intend to apply multi-task learning to train one model for question processing in this thesis.

A typical question processing module includes query formulation and answer type detection [74]. The query formulation can include part-of-speech tagging and stop words removing. Answer type detection categorises the questions based on the answer type, such as numeric, location, entity, and so on. The typical question processing module aims to enhance the performance of a question answering system. However, there is more potential in using this vast amount of data to extract other valuable information from the question itself.

Therefore, we report here on a multi-task processing question network we called QBERT (Q stands for question) which is built with a leading deep neural network BERT [43]. QBERT is built to solve three tasks we defined in the question domain: detecting the topic of questions, detecting equivalent questions with similar meaning but different wording, and locating potential answers to these same questions. These three tasks are helpful in discovering the information in the questions beyond facts.

Our approach is based on a fine-tuned language model sentence-BERT (SBERT) [125], a Siamese BERT that projects the sentences into high-dimensional vector space. The embeddings of the sentences with similar semantic meanings are close to each other in the high-dimensional space. Notice that our intention is not to design a new model but to fine-tune SBERT in a multi-tasking way for processing questions. After fine-tuning SBERT with different tasks and loss functions, the embeddings generated from input data can be used for both classification and retrieval tasks. Moreover, our approach is time-efficient in finding the related sequence in a large corpus such as Wikipedia while combining with approximate nearest neighbour search [70].

After training QBERT, we apply it to analyse a real-world dataset. In 2017, “Project What If” was started at the “We The Curious” science centre of Bristol (UK), with the stated intention of being the first exhibition all about “the curiosity of a city”. Its aim was no less than to capture the curiosity of Bristolians (and visitors) by collecting all their questions. It was focused on the questions “of real people”, and through these is aimed at understanding what Bristolians were curious about. In other words, it was not so much about the answers to individual questions as it was about understanding a Community from the questions it asks.

Despite the clear identity of “We The Curious” as a science centre, the organisers of this project were trying to gauge a broader set of interests, about culture and society, in a time of rapid change. A collection of the spontaneous questions of thousands of people was expected to tell us a lot about the people who asked them. It was expected that through this project “We The

Curious" could learn more about the changing the role a science centre might play in exploring these questions together with its community. Over the following three years, the project gathered over 10,000 questions, both in their "museum" venue and in initiatives around the city. That list taken together contained many questions, worries, doubts, and ambitions of thousands of citizens.

A "We The Curious" corpus (WTC corpus) is built up with all these questions. By analysing the queries from the WTC corpus, we present QBERT in a practical way and show that it can be used for analysing questions like data from other sources. Besides, by analysing the results produced by QBERT, we provide the first overview of the WTC corpus, including identifying the visitor's interest, the level of understanding, and the questions that a computer can answer with reference from Wikipedia.

1.1 Research Scope

This thesis focuses on applying the multi-task learning approach to processing and analysing questions. To help better understand the remaining chapters and narrow down the research area presented in this thesis, we review the three most important concepts in this section. These concepts include question, question processing, and multi-task learning.

1.1.1 Question in English

The Cambridge grammar of the English language [67] defined "Question" at two levels in linguistics. At the semantic level, a question is capable of acquiring a set of logically possible answers. At the pragmatic level, a question is a speech act designed to obtain information from the addressee. In real scenarios, these two requirements can be an unnecessary and sufficient condition for defining a question.

People might pay attention to different kinds of questions while researching and designing natural language processing models. Questions include but are not limited to factoid questions (that can be answered with a precise answer with a short phrase or sentence), non-factoid questions (open-ended questions that require more complex explanations), factual questions (that require fact-based answers), counterfactual questions (that ask for what would happen if something was not the case like "what if"), and domain-specific questions (that focus on a certain domain like medicine, biology, etc.). These categories might not be completely independent of each other.

In our research, we do not limit the domain of the questions. When we analyse the questions collected by "We the Curious" science centre, both factual and counterfactual questions are considered. When training the model to answer the questions, the datasets are confined to factoid questions. However, we do not filter out the non-factoid questions collected by "We the Curious".

1.1.2 Processing Questions

Question Processing is recognised as an important stage for question answering systems. Typically, the question processing stage includes question parse, classifying the questions based on expected answer type, and keyword extraction [61]. By processing questions in such a way, they try to perform better question answering. But in spite of that, we think that processing questions has the potential to produce more information than act as an early processing stage for question answering systems.

As a result, we define *processing questions* as a sub-field of natural language processing that targets analysing the semantic and syntactic information in the question. More specifically, three question-related tasks are included to process the questions: building question taxonomy, identifying similar questions, and question answering. More details will be explained in section 5.1.

1.1.3 Multi-task Learning

Multi-task learning (MTL) is a learning approach that improves generalisation by adding inductive bias, such as training signals in related tasks and domain information [25]. There are two main strategies for multi-task learning. One standard approach is adding additional tasks, which are also referred to as auxiliary tasks, to improve the performance of the target task. Another [99] is learning all the tasks jointly without identifying the primary task. Empirically, adding additional tasks to improve primary tasks is more similar to transfer learning, which fine-tunes a pre-trained network trained on auxiliary tasks with the primary task.

In transfer learning, the learning methods are categorised by the domain and task share between target and additional tasks [123]. Following the terminologies in transfer learning, Worsham and Kalita [160] formulated a similar taxonomy for MTL. If the target and auxiliary tasks share the same tasks but different domains, the learning method is considered transductive MTL. For MTL that shares the same domain but distinct tasks, it is called inductive bias MTL. Moreover, if the network does not share either domain or task during training, it is referred to as multi-task feature learning.

There are two approaches for training an MTL model: hard parameter sharing and soft parameter sharing [128]. Hard parameter sharing shares the parameters of all shared network layers between all the tasks. On the other hand, soft parameter sharing adds a constraint during training to create separate but similar parameters for each task.

In this thesis, our model learns all the tasks jointly using data in the same domain without identifying any primary task. The aim of applying MTL is to have a "generalist" that can achieve a good performance on all tasks. The question processing model is trained with three distinct question-related tasks. Thus, it is an inductive bias MTL trained with hard parameter sharing.

1.2 Research Questions

The research in this thesis is guided by a few research questions (RQs). These research questions came up during exploring sentence embedding and its application in processing questions. Each study in this thesis contributes to addressing these RQs. Therefore, we use these research questions as a road map for going through all the studies. By answering all these research questions listed in this section, this thesis will be able to provide insight into multi-task sentence embedding and question processing.

Our research is built on top of the success of deep learning in sentence embedding, especially the success of the pre-trained language model BERT. Thus, our first RQ is: **RQ1: How does BERT learn a sentence representation?** RQ1 is addressed in chapter 4 by clarifying BERT's structure and training strategies. Understanding how BERT works helps build a good foundation for training and fine-tuning it for specific tasks.

In word embedding, research [101] revealed semantic and syntactic relationships within embeddings that have not been seen during the learning process. When looking into the sentence embedding generated by BERT, we are interested in **RQ1a: What properties do BERT sentence embedding contain?** RQ1a enables us to investigate whether the sentence embedding can be decomposed and if the BERT embedding holds some semantic or syntactic information.

In this thesis, we are particularly interested in using the sentence embedding model for processing and analysing questions. The second RQ is asked to comprehend the questions: **RQ2: What information can we learn from Questions?** RQ2 promotes scoping the potential tasks that can be defined and utilised for the model to learn the questions. To be more specific, **RQ2a: What tasks can be included in processing questions?** Chapter 5 and chapter 6 try to answer RQ2 and RQ2a by proposing a model for processing questions and an analysis of a new dataset.

When designing the model for processing questions, one of the challenges is that it is expensive to train individual models for every single task. Thus, we introduced multi-task learning, which leverages one model for all question-related tasks. While training the models, the RQ is **RQ3: How does multi-task learning affect the performance of processing questions?** With regards to RQ3, and to further investigate different training strategies and the benefits of using multi-task learning, we design a series of experiments to understand **RQ3a: Can one model perform multiple tasks for processing questions, as well as create a balanced performance between tasks?** and **RQ3b: Does adding more question related data improve the performance on specific task?**

The final RQ for this thesis is **RQ4: How can processing questions help in education, such as in teaching and educational avenues like science centres and museums?** In chapter 6, we address RQ4 by describing how this study can be used in the education setup based on the result of analysing the WTC corpus. While analysing the WTC question corpus, we also ask **RQ4a: What information does the WTC reveal?** and **RQ4b: Can we understand questioner's interests by analysing these questions?** With RQ4, we are aiming to connect this research with a real-world scenario and open a discussion on the advantages of analysing questions.

1.3 Main Contributions

This section lists the key findings based on the experiments and results in this thesis.

- Defining three tasks for understanding and processing questions: question taxonomy, equivalent questions detection, and question answering.
- Decomposing the BERT sentence embedding into word representations with a linear system.
- Using the learned word representations from the linear system to predict the actual BERT sentence embedding for a sentence that has never been seen.
- Fine-tuning BERT to embed the sentence for each question processing task.
- Enhancing the performance and generalising question processing models by introducing QBERT, which fine-tunes BERT with multi-task learning techniques. With multi-task learning, we find that the performance of one of the QBERTs we train achieves similar performance compared to its corresponding single-task models.
- Proposing a pipeline for analysing a new question corpus.
- Applying QBERT to WTC corpus and providing the first analysis of the datasets according to the processing results.
- Discussing how processing questions can be helpful in teaching and educational avenue like science centres.

1.4 Thesis Outline

This section presents a detailed overview of the thesis, as well as a summary of each chapter. The chapters cover the deep learning models applied in our research. We mainly focus on understanding and decomposing one of the large pre-trained language models, BERT. Furthermore, we introduce a multi-task model we train for processing questions. In the end, an application of our model to a real-world dataset collected by “We the Curious” science centre is presented.

Chapter 1 - Introduction

This chapter has introduced the thesis and discussed the research background of the study. The study is motivated by the increasing number of questions collected from online communities and the rise of smart devices. On top of that, with respect to the development of deep learning algorithms in natural language processing, these numerous questions can be processed automatically. This chapter gives a scope on what exactly part of "questions" and methods we study in this thesis. Four research questions are clarified in this chapter, which indicate the aims of our studies. Furthermore, the main contributions of the study are identified in this chapter.

Chapter 2 - Preliminaries: Deep Learning in Natural Language Processing

This chapter presents all the preliminary concepts and notations that are required for a better understanding of this thesis. Firstly, it describes the concept of embedding and compositionality. Followed by that, we provide a basic explanation of the main deep learning networks that have been leveraged in natural language processing research. It starts from the recurrent neural network and ends up in transformer networks. Alongside the evolution of the neural networks, we also explain some crucial structures and mechanisms for the networks and natural language processing, such as attention mechanisms and pre-trained language model. Lastly, this chapter includes the evaluation matrices used in this thesis.

Chapter 3 - Literature Review

The purpose of this chapter is to establish a link between the contribution of our study and the previous work by elaborating on the primary research that has been done in the related areas clarified in the previous chapters. The chapter begins with an extensive review of existing pre-trained language models. Then it presents the research on combining multi-task learning with natural language understanding. Moreover, this chapter introduces all the techniques used for sentence embeddings, followed by an in-depth inspection of compositionality studies in sentence embedding. Finally, we investigate at question processing that includes the studies of all the question-related tasks.

Chapter 4 - Compositionality in BERT Sentence Embedding

This chapter thoroughly explains and illustrates how the sentences are projected into a vector space using SBERT. This model has created leading performance in multiple natural language processing tasks and can be used to generate meaningful representations for sentences. We also explain the structure and learning process of BERT in this chapter.

In order to understand the information captured by BERT sentence embedding, we introduce a method to interpret BERT embedding by decomposing it into its properties. Furthermore, a series of statistical hypothesis tests were introduced to measure compositionality in sentence embeddings, including a linear system that decomposes the sentence embedding into phrase representation and three experiments to predict a sentence embedding without seeing the actual sentence during the training process. We demonstrate these tests by using SBERT to embed the sentence from a simple sentence corpus, which is generated for investigating the attributes contained in the sentence embedding. By decomposing the sentence embedding, we obtain a set of attribute representations that can be utilised to predict the embedding of a sentence. The predicted embeddings achieve a high similarity compared to the original BERT sentence embeddings.

Chapter 5 - Training QBERT for Processing Questions

In this chapter, we define three question-related tasks, including question topic classification, equivalent question recognition, and question answering. After that, we introduce QBERT, a generalist model that trains SBERT with a multi-task learning approach in the context of processing questions. The architecture of the networks is demonstrated. Moreover, we include the details of training and evaluating QBERT. The methods of training QBERT are illustrated with graphs, equations, and algorithms. On top of that, the chapter demonstrates the datasets used to train the network and the methods to evaluate the multi-task network on all question-related tasks. QBERT is trained with various curricula. One of the QBERT models we trained is able to generate a general representation for multiple tasks and obtain a balanced performance among all Question Processing related tasks and has achieved a similar performance compared to its corresponding single-task models. Finally, a discussion of the results is given based on the performance of QBERT.

Chapter 6 - Analysing a New Question Corpus: WTC Corpus

This chapter presents a pipeline for processing a question corpus. Based on the method described in the previous chapter, in this chapter, we apply one version of QBERT to a new corpus collected from “Project What If” held by “We The Curious”, a science centre located in Bristol, UK. The project encourages their visitors and Bristolians to write down their questions and explore their curiosity. The methods of collecting questions are presented exhaustively. The corpus collected by the science centre contains more than 10,000 questions and has been moderated beforehand. With this large amount of data collected, it was time-consuming to process and analyse all the questions by a human. In addition, the questions collected cover various types and topics and contain similar questions with different wordings. Therefore, using QBERT to process the questions can help comprehend Bristolians’ curiosity without human intervention and reduce the amount of data for further potential analysis. By analysing the processing results given by QBERT, we find out what topic the visitors are interested in and what are the most popular questions. Furthermore, QBERT tries to answer the question with a one-sentence answer extracted from Wikipedia Summary. At the end of this chapter, we also discuss the contribution of processing questions and this study in the field of education.

Chapter 7 - Conclusions

This chapter concludes the thesis by summarising the work done in the studies. The key findings from the studies have constructed an understanding of sentence embedding in terms of compositionality and how generalist sentence embedding can help process and comprehend questions in multiple aspects. In this chapter, we try to answer the research questions raised for this thesis. At the end of this chapter, some key limitations in our work are stated, along with possible future research directions to understand sentence embedding and discover more information from processing questions.

CHAPTER 2

PRELIMINARIES: DEEP LEARNING IN NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is the task of understanding human (natural) language using a computer. For decades, NLP tasks have been based on shallow machine learning models such as support vector machine (SVM) and logistic regression trained on high dimensional and sparse features. Thus, by using deep learning methods, it aims to learn the patterns that are hard to be learned efficiently by the traditional machine learning models.

Deep learning architectures have achieved state-of-the-art results in natural language understanding. With the multiple processing layers, the artificial neural network can learn hierarchical representations of the text. Compared to traditional machine learning based NLP systems that rely on hand-crafted features, the deep neural network enables the computer to learn a dense feature representation without human intervention. Research [35] showed that a simple deep learning method could outperform the state-of-the-art results in some NLP tasks.

In this chapter, we will introduce language processing and deep learning concepts related to this thesis. Section 2.1 defines the general mathematical notations used in the thesis. In section 2.2 we will explain embedding, which generates representation for text analysis. This section will also clarify the idea of embedding compositionality which is critical for the research presented in the thesis. Followed by the embedding, we will introduce some signature deep neural networks and techniques used in NLP research in section 2.3. In the end, section 2.4 describes the concept of the pre-trained language model that has been widely used in language understanding research and created a leading performance in various tasks.

We assume the reader has prior knowledge of linear algebra, machine learning, and natural language processing.

2.1 Notations

In this thesis, a set with n elements/attributes is denoted as equation 2.1, where the uppercase letter X represents the set, and the lowercase x represents the elements.

$$X = \{x_0, x_1, \ldots, x_{n-1}\} \tag{2.1}$$

Note that the index starts from 0.

Instance/item is represented by non-bold character, for example a set with n instances are notated as $I = \{i_0, i_2, \ldots, i_{n-1}\}$. Vector is represented by bold characters such as $\mathbf{V}$ as follows.

$$\mathbf{V} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} \text{ or } \mathbf{V}^T = [v_0, v_1, \ldots, v_{n-1}] \tag{2.2}$$

The matrix is represented with upper-case bold characters, and the matrix dimensions are denoted as subscripts. For example, $A_{m,n}$ is an m by n matrix.

$$\mathbf{A}_{m,n} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-1,0} & a_{m-1,1} & \cdots & a_{m-1,n-1} \end{bmatrix} \tag{2.3}$$

We denote the ground truth label as Y and the predicted result as $\hat{Y}$.

2.2 Embedding and Compositionality

To process text, we need to encode it into a form the computer can understand. This process is known as embedding in machine learning. This section will explain the idea of embedding and compositionality.

2.2.1 Embedding

One way to model a set of items using their relations is to represent them as points in a vector space. By representing items as vectors, we can present the relations of interest between those items by geometric or algebraic relations between those vectors of coordinates. This process of modelling the items is also known as "Embedding".

Given a set of relations that we want to enforce, there may be multiple solutions to this problem, even after we have specified which geometric relations we will use to represent them. For example, we may decide that a set of documents should be embedded in space in such a way that the distance between vectors reflects their content similarities.

In machine learning, embedding also involves mapping discrete variables to a learned continuous vector representation with a task-specific mapping function. When saying $\mathbf{B}$ is the embedding of I, it can be written as

$$\mathbf{B} = \Phi(I)$$

Where Φ is the mapping function that transfers the discrete vector to a continuous dimensional space, the embedding function can be learned from data. Supervised information tailors the learned embedding for the desired task. For example, in word embedding, co-occurrence information of words is used as the supervised information.

In this thesis, we apply embedding to represent word tokens. Furthermore, we focus on training multi-task sentence embeddings and understanding sentence embeddings by decomposing them.

2.2.2 Compositionality

Compositionality is a crucial factor for human understanding. For instance, when we learn the word "unbearable", we are able to learn and remember the meaning of the word by its root "bear", prefix "un", and affix "able". This also happens in understanding a sentence. The meaning of the sentence is often deduced from the meaning of its parts [79]. Considering a nonsensical sentence, such as "A pink zebra writes a paper that is a self-portrait.", a human can easily extract the sentence's meaning without seeing the sentence before. However, such a sentence might never occur in any corpus for training an algorithm. Therefore, for artificial intelligence to really "understand", it should not just memorise all the possible combinations of the sentences. Instead, the algorithm should learn the compositionality even if the compositional structure is not demonstrated in the training process [49].

Compositionality can mean many things under different circumstances. In linguistics, a representation is called "compositional" if the meaning of an expression depends only on its structure and the meaning of its elementary parts. If this was taken literally, we should see that the embedding of "big cat" should result from the combination of two vectors, one for "big" and one for "cat". We see this in singular/plural or male/female inflexion in words.

A property of certain embeddings that has the potential to help with the above concerns (as well as others) is that of "compositionality". Introduced in the domain of traditional linguistics, this property has been extended to also cover vector representations. Traditionally it refers to the way in which the meaning of a linguistic expression results from its components. For example, single words can often be decomposed into parts that modify the meaning of the initial word stem, as in the following example.

- Com+pose
- De+com+pose

- De+com+pos+ition
- Com+pos+ition+al
- Com+pos+ition+al+ity

In the case of vector embeddings, we substitute the "string concatenation" operation with the "vector addition" operation, so that a vector representation is compositional if it can be regarded as the sum of a small set of components (which can hopefully be interpreted and even manipulated). As an artificial example of this idea, we could imagine an embedding Φ that maps from items (tokens) to vectors in such a way that

$$\Phi(compositionality) \approx \Phi(com) + \Phi(pos) + \Phi(ition) + \Phi(ality)$$

This thesis defines a learned representation as compositional when it can represent complex concepts or items by combining simple attributes [53]. We mainly look into additive compositionality as follows.

$$\mathbf{b}_I = \sum_{i=0}^{n-1} \mathbf{x}_i$$

Where I is the item that has a set of n attributes, $\mathbf{b}$ is the vector representation of the item I, and $\mathbf{x}$ is the attribute vector.

2.3 Sequence Models: From RNN to Transformer

Sequence models are machine learning models that process data in sequences, such as time-series data, video, and text. When using a sequence model, the sequence is more important than the individual data point. For example, in a word, the combination of the characters is more important than the characters themselves. Applying the sequence model means that the processed data are no longer independently and identically distributed samples. The data carry some dependency because of their sequential order [91].

Sequence models are considered to have "memory" over previous computations. It is widely used for the computer to understand and generate a representation of a sentence as its meaning is associated with combining words. In this section, we will start by explaining a neural network specifically designed for processing sequential data, followed by other networks and mechanisms that are developed on top of that.

2.3.1 Recurrent Neural Networks

Recurrent Neural Network (RNN) [129] is a neural network which can process sequential data by using a "loop" that repeats the computation over each token of the sequence for every time step. It is suitable for modelling the context dependencies in the text.

Figure 2.1 illustrates a single-layer RNN that includes input, hidden, and output layers. The input layer takes one token of the sequence as input at each time step. In the hidden layer, the current hidden state ($\mathbf{h}_t$) depends on the previous state ($\mathbf{h}_{t-1}$) and the input ($\mathbf{x}_t$) at the present step. A recurrent cell is utilised to create a fixed-sized sequence representation that contains the information of the prior time steps by feeding the token one at a time into the cell. Furthermore, the output layer projects the hidden state into a selected dimensional space.

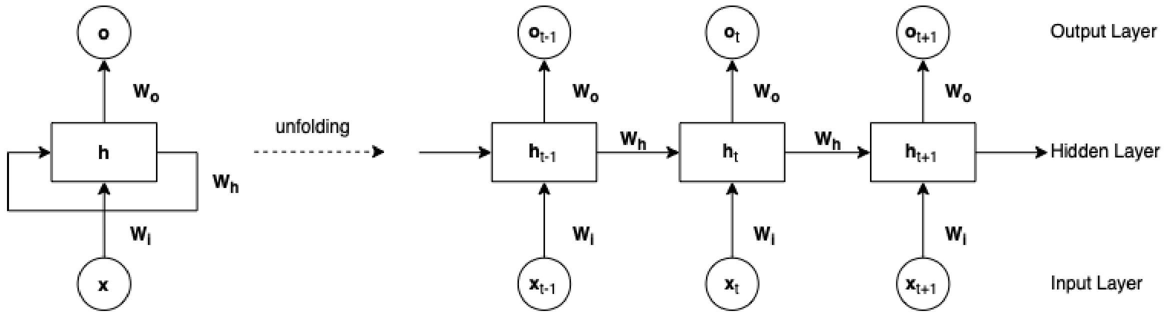


Figure 2.1: Single layer RNN. RNN cell is unfolded across time on the right.

The hidden state $\mathbf{h}_t$ and output vector $\mathbf{o}_t$ are calculated as follows.

$$\mathbf{h}_t = tanh(\mathbf{W}_i\mathbf{x}_t + \mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2.4}$$

$$\mathbf{o}_t = \mathbf{W}_o\mathbf{h}_t + \mathbf{b}_o \tag{2.5}$$

Where $\mathbf{x}_t$ is the input vector at time step t. In NLP, textual data, such as characters, words or sentences, are usually encoded into a fixed-size vector as the input for each time step. $\mathbf{o}$ is the output value produced by a dense layer. It can be a probability distribution among multiple classes. $\mathbf{W}_I$, $\mathbf{W}_h$, and $\mathbf{W}_o$ are the trainable weight matrices which share across different time steps of the model. $\mathbf{b}_h$ and $\mathbf{b}_o$ represent the bias.

In theory, RNN is considered a model with memory over all previous computations and current information. The memory makes RNN suitable for many NLP tasks. However, in practice, RNN faces the challenge of long-term dependencies [16]. Besides, not all the information in the sequence has to be stored in the memory. For example, information on stopwords like “the”/“a” might be unnecessary in a sentence. As a result, the long short-term memory cell is widely used to overcome these limitations.

Long Short-Term Memory (LSTM) [65] solves the long-term dependencies by adding multiple gates and cell state $\mathbf{c}_t$ to the basic RNN structure. With these gates, LSTM is able to modify the memory of the cell by adding or removing information.

As shown in figure 2.2, there are three gates used in the LSTM, which are the input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$, and output gate $\mathbf{o}_t$.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{2.6}$$

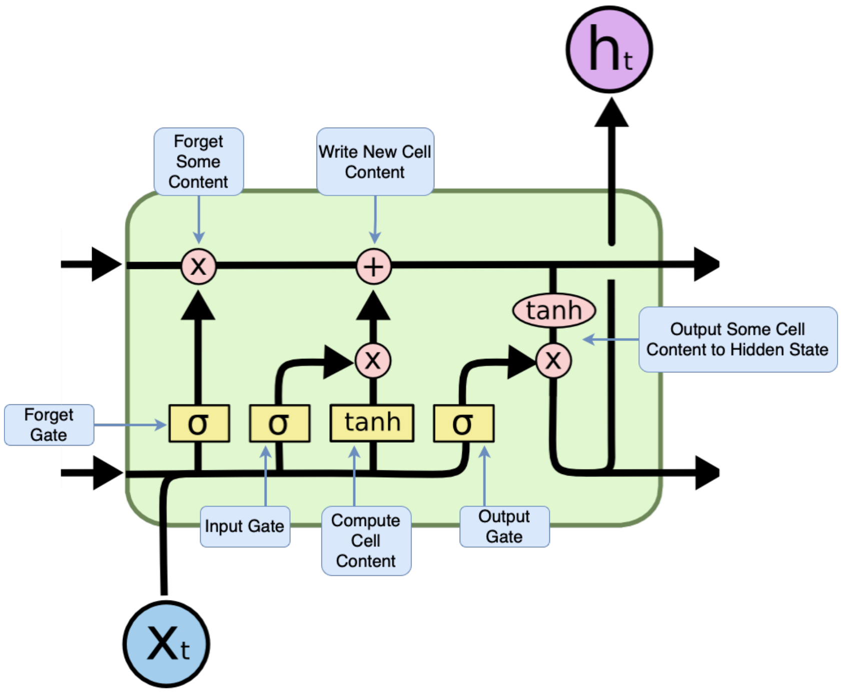


Figure 2.2: The repeating module with gates in the LSTM network. Source link: `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`

(2.7) $$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f)$$

(2.8) $$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o)$$

The cell state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$ are then updated with the gates information.

(2.9) $$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c)$$

(2.10) $$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t)$$

The sigmoid function for each gate gives a value between zero and one that decides how much information should go through. One lets everything through, and vice versa. The cell state keeps the memory from the previous time steps. The input and forget gates decide what to store in the memory (cell). The output gate controls what parts of the cell are output to the hidden state.

Another popular structure in the recurrent neural network is called GRU (Gated Recurrent Unit) [33]. It is long-established in sentence representation methods. GRU uses two gates, an update gate ($\mathbf{z}_t$) and a reset gate ($\mathbf{r}_t$), instead of three compared to LSTM. The gates are calculated as follows.

(2.11) $$\begin{aligned}\mathbf{z}_t &= \sigma(\mathbf{W}_{\mathbf{x}}^{(\mathbf{z})}\mathbf{x}_t + \mathbf{W}_{\mathbf{h}}^{(\mathbf{z})}\mathbf{h}_{t-1})\\ \mathbf{r}_t &= \sigma(\mathbf{W}_{\mathbf{x}}^{(\mathbf{r})}\mathbf{x}_t + \mathbf{W}_{\mathbf{h}}^{(\mathbf{r})}\mathbf{h}_{t-1})\end{aligned}$$

The update gate determines the information to keep from the previous hidden state. And the reset gate decides the information forgotten from the previous hidden state. The hidden state is updated as equation 2.12.

$$\begin{gathered}\mathbf{h}'_t = tanh(\mathbf{W}_{\mathbf{x}}^{(\mathbf{h}')}\mathbf{x}_t + \mathbf{r}_t \odot \mathbf{W}_{\mathbf{h}}^{(\mathbf{h}')}\mathbf{h}_{t-1}) \\ \mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1-\mathbf{z}_t) \odot \mathbf{h}'_t\end{gathered} \tag{2.12}$$

2.3.2 Encoder-Decoder and Sequence-to-Sequence

Based on the topology of the sequence model, at the last step, the state unit contains all the information from the previous time steps. Thus, it can be applied to summarise a sequence and produce a fixed-size representation with the hidden state at the last step, as shown in figure 2.3. For instance, for a sentence, it takes a word token as input for each time step. In such a way, the last hidden state is the embedding of the input sentence.

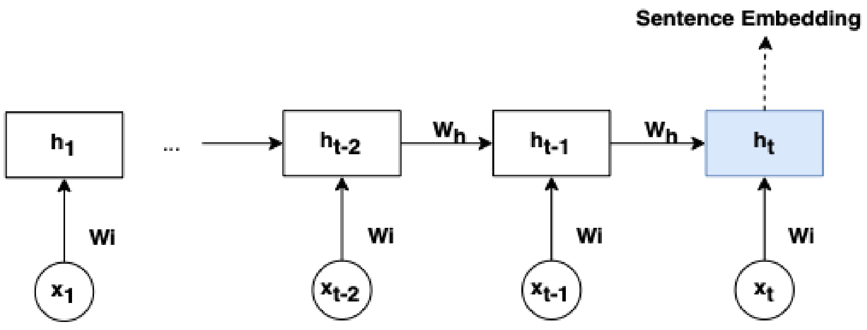


Figure 2.3: The basic sentence embedding architecture of the RNN. The last hidden state of the network is recognised as the sentence embedding.

To acquire the sentence embedding, people trained the model with downstream tasks. Two types of frameworks are widely used for learning a representation: many-to-one and many-to-many. As shown in figure 2.4, a many-to-one model takes sequential data as input and produces a fixed-length vector that can be used for tasks like sentiment analysis, topic classification, etc. Another more popular model is the many-to-many model, which contains two parts: an encoder and a decoder. The encoder takes a sequence as an input and learns a representation, while the decoder interprets the representation into another new sequence. For instance, in a machine translation system, the model learns English and outputs the target sequence in French. This model that maps the embedding back to another sentence by applying an encoder-decoder framework is also known as a sequence-to-sequence model [145].

The core idea of the encoder-decoder framework is to apply one network to summarise the input sentence into a fixed-length vector and another network to generate the target sentence from the vector. The last hidden state of the encoder, recognised as the context vector, is used as the initial state for the decoder. The encoder-decoder framework can map the input and

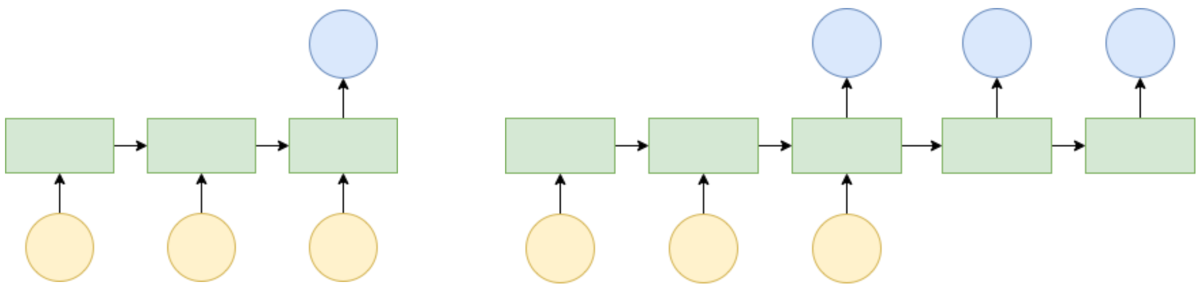

(a) Many-to-One (b) Many-to-Many

Figure 2.4: Two types of sequence models used for sequence representation learning.

output sequence in different lengths, which has been proven to significantly improve the result in machine translation [68] and conversation models [136].

Although the sequence to sequence model is an effective model to summarise the input sequence, this framework faces a problem when the input sequence is very long and information-rich. Based on the long dependencies problem of RNN mentioned before, learning the representation for capturing all the semantic information in long sentences is very challenging [54]. For some text generation tasks, not all context of the input sequence is needed at each decoding step. For example, in machine translation, while translating “boy” in the sentence “A boy is eating the banana.”, the model does not need to know the rest of the words in the sentence. In the process of decoding, a more practical approach is to allow the decoder to refer back to the part of the input sequence that is highly related to the decoding part. This inspires the attention mechanism.

2.3.3 Attention Mechanism

Attention was first presented by Bahdanau et al. [10] and used in machine translation. It allows each output in the target sequence to pay attention to the related part of the input sequence instead of the whole context. Comparing to the original sequence to sequence model, which only considered the last hidden state, the attention mechanism conditions on every hidden state of the encoder. Applying the attention mechanism results in a more targeted and better performance in many NLP tasks.

The general attention mechanism performs the following computations. Each query $\mathbf{q} = \mathbf{o}_{t-1}$ calculates a score value with a matched key $\mathbf{k}_i$ so that

$$e_{\mathbf{q},\mathbf{k}_i} = \mathbf{q} \cdot \mathbf{k}_i \tag{2.13}$$

The generalised attention is then computed as a weighted sum of the value vectors (hidden state) $\mathbf{h}_i$ as follows.

$$\begin{aligned} \alpha_{\mathbf{q},\mathbf{k}_i} &= softmax(e_{\mathbf{q},\mathbf{k}_i}) \\ attention(\mathbf{q},\mathbf{K},\mathbf{H}) &= \sum_i \alpha_{\mathbf{q},\mathbf{k}_i} \mathbf{h}_{\mathbf{k}_i} \end{aligned} \tag{2.14}$$

Where the Softmax function takes the input $e_{\mathbf{q},\mathbf{k}_i}$ and normalises it into a probability distribution as equation 2.15. After applying the softmax function, the input components are in the interval $(0,1)$ and have a sum of 1.

$$softmax(e_{\mathbf{q},\mathbf{k}_i}) = \frac{exp(e_{\mathbf{q},\mathbf{k}_i})}{\Sigma exp(e_{\mathbf{q},\mathbf{K}})} \tag{2.15}$$

With attention, LSTM has been found successful in many language generating and representation learning tasks [9, 117]. However, adapting the input, forget, and output gates will need four times of parameters compared to basic RNN. Thus, it is more challenging for computer resources and takes longer to train. Besides, due to the “recurrent” structure of the network, it takes one token input at a time, which can not make full use of the parallel computation provided by GPU.

As a result, in 2017, Vaswani et al. introduced the idea of a Transformer network that still uses the encoder-decoder structure and attention but reduces the training time [148].

2.3.4 Transformer

Transformer [148] was initially built for machine translation. It uses an encoder-decoder architecture similar to a recurrent-based network. The difference is that the input sequence can be passed in parallel. The encoder and decoder learn various tasks during training. For instance, in machine translation, the encoder learns the context; on the other hand, the decoder learns the relation between two languages. In our research, we only leverage the encoder part of the Transformer. Thus, we will focus on explaining the encoder of the Transformer network.

Transformer adds a positional encoding on top of the input token embedding to include the context information. In such a way, the same word in different content can have different embeddings that refer to distinct meanings. The positional embedding is calculated as equation 2.16.

$$\begin{gathered} Positional\_Encoding(pos, 2k) = sin(pos/10000^{2k/d_{token}}) \\ Positional\_Encoding(pos, 2k+1) = cos(pos/10000^{2k/d_{token}}) \end{gathered} \tag{2.16}$$

Where pos is the position of the token in the sequence, d_{token} is the size of the token embedding, and k refers to the individual dimensions of the token embedding.

Then the word vectors are passed to an encoding block that contains a multi-head self-attention layer and a fully connected feed-forward layer. The feed-forward layer is applied to all the attention vectors and generates fix-sized representations for the next encoder block or decoder. Since the attention vectors are independent of each other, the network can take the sequential data simultaneously by using the parallelised feed-forward networks.

In the attention layer, Transformer applies a self attention in equation 2.17 instead of the general attention explained in equation 2.14. Self attention scales the attention weight with $\frac{1}{\sqrt{d_{\mathbf{k}}}}$, where $d_{\mathbf{k}}$ is the dimension of the key.

(2.17)
$$attention(\mathbf{Q},\mathbf{K},\mathbf{H}) = softmax(\frac{\mathbf{Qk}^T}{\sqrt{d_{\mathbf{k}}}})\mathbf{H}$$

Using self-attention results in the attention vector being weighted higher to itself. Therefore, Transformer introduces a multi-head attention mechanism that takes the weighted average of eight attention vectors for each token as below.

(2.18)
$$\begin{gathered} MultiHead(\mathbf{Q},\mathbf{K},\mathbf{H}) = Concat(head_1,...,head_8)\mathbf{W^O} \\ where\ head_i = attention(\mathbf{QW}_i^{\mathbf{Q}},\mathbf{KW}_i^{\mathbf{K}},\mathbf{HW}_i^{\mathbf{H}}) \end{gathered}$$

Where $\mathbf{W^Q} \in \mathbb{R}^{d_{token} \times d_k}$, $\mathbf{W^K} \in \mathbb{R}^{d_{token} \times d_k}$, $\mathbf{W^H} \in \mathbb{R}^{d_{token} \times d_h}$, and $\mathbf{W^O} \in \mathbb{R}^{8d_h \times d_{token}}$.

The Transformer encoder generates word representation that encapsulates the meaning of the words. The decoder then takes these embeddings and the previous word in the target sequence to generate the next word. The decoder has a similar structure to the encoder but with an extra masked multi-head attention block to ensure that the sequence's future words are masked during the learning process.

Transformer has replaced LSTM in sequence to sequence and sequence to vector models. And it is used to create pre-trained language models in the NLP tasks.

2.4 Pre-trained Language Model

The language model's purpose is to help computers better understand human language. It can be used as a tool to perform any tasks that are related to language understanding, such as question answering, sentiment analysis, text summarisation, and machine translation.

Traditionally, a language model is a model that assigns a probability to the word based on the previous sequence, which is also referred to as an n-gram language model (n-gram refers to n-word sequence) [15]. For example,

$$P(Sea|The\ sharks\ live\ in\ the)$$

Given a sequence with words $w_0, w_1, \ldots, w_{n-1}$, the probability of next word w_n can be calculated as $P(w_n|w_{0:n-1})$. The probability of the entire sequence $P(w_0, w_1, \ldots, w_{n-1}, w_n)$ can be computed as equation 2.19.

(2.19)
$$P(w_0 \ldots w_n) = \prod_{k=0}^{n} P(w_k|w_{0:k-1})$$

With the increasing amount of text, the language model will have more and more vocabulary. As a result, the number of possible n-gram sequences in the training set increases exponentially, which causes a data sparsity problem. Neural network based modern language models are first introduced to improve the sparsity problem by leveraging learned continuous representations of words to make predictions [15]. Since then, many different neural networks have been used to train a language model, such as RNN, LSTM, and Transformer.

Pre-training is a kind of transfer learning where the model is trained first on auxiliary tasks before being fine-tuned for a particular task. Without pre-training, a neural network typically initialises its parameters with random states. On the other hand, with pre-training, the network can start with the parameters that have been optimised using auxiliary data.

In such a way, the language model can be “well read” before performing on a specific NLP task, and the old knowledge can help achieve better performance on the new tasks. In deep learning, pre-training intends to use self-supervised learning on large scale unannotated corpus instead of supervised training, such as Wikipedia dump and books. Applying pre-training on a neural language model achieves a big success in NLP. More details will be introduced in section 3.1.

2.5 Evaluation Metrics

In this section, we will give a brief explanation of all the evaluation metrics used in our research.

2.5.1 Cosine Similarity

Cosine similarity measures the similarity between two vectors by measuring the cosine of the angle between them. The score has a distribution between -1 to 1, where 1 means exactly the same, and -1 means exactly the opposite. For vector **x** and **y**, cosine similarity D_{cosine} is defined as follows.

$$D_{cosine} = \frac{\mathbf{x}\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\Sigma_{i=0}^{n-1} x_i y_i}{\sqrt{\Sigma_{i=0}^{n-1} x_i^2}\sqrt{\Sigma_{i=0}^{n-1} y_i^2}} \tag{2.20}$$
$$Cosine\_Distance = 1 - D_{cosine}$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the component of vector **x** and **y**, respectively.

Because it is unaffected by the magnitudes of the vectors (the lengths of the texts), cosine similarity is frequently used to measure text similarity by calculating the angle between vectors. Additionally, in NLP, texts are typically represented in a high-dimensional space. Cosine similarity is more robust in high-dimensional space than Euclidean distance.

2.5.2 Euclidean Norm

Euclidean Norm calculates the distance between the origin to a point **X** in the vector space $\mathbb{R}^n$.

$$Norm = \|\mathbf{X}\|_2 = \sqrt{\Sigma_{i=0}^{n-1} \mathbf{x}_i^2} \tag{2.21}$$

2.5.3 Accuracy and F1

Accuracy is a metric used for evaluating classification model as follows.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ Number\ of\ predictions} \tag{2.22}$$

F1 score is also used to evaluate the classifier when the dataset is unbalanced. F1 score is the combination of Precision and Recall which is calculated as equation 2.23.

$$\begin{aligned} Recall &= \frac{TP}{TP+FN} \\ Precision &= \frac{TP}{TP+FP} \\ F1 &= \frac{2*Precision*Recall}{Precision+Recall} \end{aligned} \tag{2.23}$$

Where TP, FN, and FP stand for true positive, false negative, and false positive, as shown in Table 2.1.

Table 2.1: Confusion Matrix

		Actual Class	
		Positive (P)	Negative (N)
Predicted Class	Positive (P)	TP	FP
	Negative (N)	FN	TN

For the retrieval task, we measure the accuracy at rank K (Accuracy@K), which measures how often the actual class falls in the top K predictions.

Chapter 3

Literature Review

In Chapter 2, we provided some of the preliminary knowledge in deep learning and language processing required to understand this thesis. In this chapter, we discuss the background research on deep learning networks in different NLP research and applications, aiming to further define the scope of this thesis and link our contribution to the existing research.

Firstly, we review the literature on pre-trained language models, which reveals the latest direction and approach for training and applying NLP models. Then we explore the approach we apply for learning the vector representation of the text: sentence embedding. These dive further into the contribution space.

In our research, we focus on processing questions by:

- categorising the questions
- identifying similar question pairs
- question answering

Thus, we also explore the existing question taxonomy, text similarity, and question answering research.

3.1 Pre-trained Language Model

The development of deep learning enables researchers to solve more NLP problems with neural networks. As mentioned in the previous chapter, many different types of neural networks have been used, and the architectures have become more complicated. The deep structure creates good performance and relieves people from feature engineering. However, a complex structure also means that it requires more data to train a model. Supervised data in NLP is relatively limited for some tasks and expensive to obtain. Deep models are more likely to overfit without sufficient datasets [13]. As a result, recent work focuses on leveraging pre-training, a transfer learning approach that trains the model to learn the knowledge from source tasks/domain and applies it to new tasks/domain [146]. A pre-trained language model usually applies self-supervised learning on a large annotated corpus to learn the general language representations, which can benefit the downstream tasks and avoid overfitting on small datasets [48].

Earlier pre-trained language models started with “modern” word embedding, representing words as a dense vector with the neural network language model. Mikolov et al. [102] proposed two models, continuous bag-of-words and skip-gram, and were able to learn high-quality word embeddings that could be used to enhance the performance of NLP tasks. Another word embedding method, GloVe [113], which learned the word embedding with a word co-occurrence statistic from a large corpus, also became popular for representing words for downstream tasks. Although these pre-trained word embeddings were able to capture the semantic content, they are context-independent. Furthermore, their downstream models still need to learn from scratch. This type of pre-trained language model is also referred to as feature-based strategy [43].

Another strategy is called fine-tuning strategy, which trains the downstream tasks by fine-tuning the pre-trained model. Dai and Le [40] proposed a pre-trained language model with an LSTM auto-encoder that applied an encoder-decoder structure to encode the input sequence into a vector and then decode it back to the original sequence. The learned parameter of the LSTM was used as the initial setup for a later supervised sequence learning task. Their results showed that pre-training could improve the generalisation of the LSTM and simplify the training process for the target tasks. Moreover, combining pre-training and fine-tuning improved the results of many text classification tasks.

To capture the context, the researchers applied the attention mechanism to their models. Ramachandran et al. [122] found that pre-training the LSTM sequence-to-sequence model with multi-layer attention and residual connections can improve a range of NLP tasks by fine-tuning the model on each task. McCann et al. [98] trained a deep LSTM sequence-to-sequence model with attention on machine translation datasets to learn the contextualised word vectors. Then they applied the encoder to downstream tasks, resulting in better performance in sentiment analysis, entailment, question classification, and question answering.

Another approach to capture content is by using a bi-directional model, such as bi-directional LSTM. A normal LSTM reads the input sequence in one direction (forward: from the past to the future). By contrast, a bi-directional LSTM attached the forward layer with a backward layer to read the input sequence from both directions. Peters et al. [114] proposed ELMo, the first deep contextual word embedding. ELMo first pre-trained the bi-directional LSTM to learn the word vectors. Then it fine-tuned these representations as a feature for specific tasks.

With the success of the previous pre-training on LSTM, modern pre-trained language models apply more complicated network structures, such as Transformer, and are trained on new pre-training tasks. Two Transformer-based pre-trained language models have become the mainstream approaches for NLP. One is GPT (Generative Pre-training) [24, 118, 119], which is built based on the decoder of the Transformer. The other is BERT [43], which makes use of the encoder from the Transformer.

GPT-1 [118] was first trained to target the problems in supervised learning, such as the limitation of extensive annotated data, and the limitation of one model only working for one particular task. It introduced an unsupervised pre-training process to learn the language model on the BookCorpus. As explained in section 2.4, a language model predicts the target word w_i from the previous sequence in terms of $P(w_i|w_{i-k},...,w_{i-1};\theta)$ where θ are the parameters of the neural network. After pre-training, the model was fine-tuned with supervised tasks by maximising the likelihood $P(Y|s_1,...,s_n)$ of observing label Y, and given sequence $(s_1,...,s_n)$. This showed that the model had already learned about the language during pre-training and was easier to converge for supervised tasks. GPT-1 outperformed the state-of-the-art supervised specialised models on 9 out of 12 downstream tasks.

Furthermore, Radford et al. [118] proposed a GPT-2 trained on a larger corpus (WebText) and added more parameters to the models. GPT-2 aimed at learning multiple tasks with the same pre-training model so that it calculated $P(output|input, task)$ instead. It was known as task conditioning. GPT-2 was capable of zero-shot task transfer, meaning it understood the nature of the task without giving any examples. GPT-2 results showed that training on a larger corpus and having more parameters helped achieve better performance on zero-shot tasks than state-of-the-art, especially on machine translation. However, it could have achieved better performance on text summarisation. Brown et al. [24] presented GPT-3, the latest version of GPT that intended to perform and understand the tasks without fine-tuning but only a few demonstrations. The model had a similar architecture as GPT-2, but it was trained on five different corpora, each having a certain weight. GPT-3 achieved strong performance on many NLP tasks, including translation, question answering, and cloze tasks. While GPT models rewrite the existing best performance, two major problems limit access to the models. First, when improving the performance, the model's size increased dramatically. GPT-1, 2, and 3 have 117 million, 1.5 billion (100 times more than GPT-1), and 175 billion (more than 10,000 times than GPT-1) parameters, respectively. This creates an extremely strict computation requirement for using the model. Besides, OpenAI claimed that the completed version of the model is not accessible to the public because its capabilities may lead to misuse.

Another mainstream pre-trained language model is BERT, first introduced in 2018 by Devlin et al. [43]. It leveraged the encoder of the Transformer network and outperformed state-of-the-art on various language understanding tasks. Devlin et al. introduced a new pre-training method called masked language modelling. Instead of predicting the next word in the sequence, masked language modelling predicted the masked words within the sequence. Another task BERT applied for pre-training is next sentence prediction, which classifies if two sentences are adjacent. The pre-trained model can be fine-tuned on downstream tasks.

Many other pre-trained language models are developed based on the concept of BERT. XLNet [168] improved BERT by bringing in a modified pre-training strategy called permutation language modelling. They noticed that the special token used to mask the word in masked language modelling was absent in the downstream tasks. Thus, they applied a random permutation on the order of the input sequence and predicted the target word using the permuted input. In such a way, the target words were predicted with different contexts and no longer needed to be replaced by a mask token. Nevertheless, XLNet was generally more computationally expensive and took longer to train than BERT due to the permutation language modelling. Another variant of BERT, RoBERTa [93], removed the next sentence prediction task and changed the masked tokens during each training epoch. RoBERTa outperformed both BERT and XLNet on the GLUE benchmark [151]. All these BERT-based methods have around 120 million parameters for a base model and 350 million parameters for a large one. Hence, Sanh et al. [131] and Lan et al. [82] aimed at creating a lighter version of BERT by trading off some accuracy. DistilBERT [131] applied knowledge distillation [64] that reduced the parameters of the base model to 66 million but remained 95% of the performance. ALBERT [82] claimed that the encoder layers in BERT learned similar operations on different layers. Thus, they used cross-layer parameter sharing [42] that shared all the parameters between layers. Besides, ALBERT applied factorised embedding parameterisation that decomposed the large embedding matrix into one matrix that embedded the one-hot word representation into a dense vector with dimension E and another matrix projected the vector into hidden space. By doing so, the embedding parameters were reduced from $O(V \times H)$ to $O(V \times E + E \times H)$, where V is the size of the vocabulary and H is the dimension of the hidden state. Overall, ALBERT reduced the size of BERT to 12 million parameters for the BASE model and 18 million parameters for the LARGE model. And with 70% amount of the parameters in BERT, ALBERT outperformed BERT in a range of downstream tasks.

The pre-trained language model can also be fine-tuned for a particular domain. Lee et al. [84] trained BERT for the biomedical field by pre-training BERT on biomedical-related corpora. Training BERT on domain-specific corpus enables the model to generate pre-trained word representations with a better understanding of the target field. BioBERT excelled at a variety of biomedical text mining tasks, demonstrating that BERT could be adaptable to a specific domain.

The idea of the pre-trained language model simplifies the training process for specific NLP tasks. And these models dominate the performance on various tasks, such as sentiment analysis, text classification, question answering, and machine translation. However, the complex architecture of the network requires more computational resources to train and utilise the model. Furthermore, it becomes more challenging to interpret the internal process of the network and understand the reasoning behind the network decision.

3.1.1 Multi-task Learning in Natural Language Understanding

Multi-task learning (MTL) is a collection of techniques that trains all tasks jointly. It aims at improving generalisation by adapting domain information contained in the related training tasks [25]. Pre-training can be considered as one approach in MTL. Pre-training on auxiliary task targets enhancing the performance of the primary tasks. As the examples listed in the previous part, MTL (pre-training/fine-tuning approach) has achieved significant results in NLP.

Despite the size of the deep contextual model, using a separate model for each task can be challenging for computer resources. Therefore, we are interested in finding an approach to train one model to perform different tasks. This is not a new idea. McCann et al. [99] designed an MTL model that solved all the tasks simultaneously without identifying the primary tasks. In this thesis, we call the single model that can perform multiple tasks a "generalist". In this section, we will focus on reviewing some of the existing methods to train a generalist.

There is increased attention to the problem of learning generalist agents (as opposite to specialist) in a way that the same representation can be used in a range of tasks, even if it does not excel at any specific task [124]. While a specialist should be expected to excel at its one task, a generalist is expected to be good at many problems.

Liu et al. proposed MT-DNN [92], a multi-task framework including a shared lexicon layer and Transformer encoder. To adapt to various tasks such as classification and regression, MT-DNN applied task-specific layers for different types of tasks. MT-DNN performed multi-task training based on the pre-trained BERT. All the tasks and layers were trained jointly during the process. The weights of the task-specific layers were updated according to the tasks-related training data. Fine-tuning encoder layers with multiple tasks simultaneously allowed MT-DNN to generate a shared representation between tasks. It increased the overall performance of the model on GLUE tasks [151]. However, it is arguable that MT-DNN is a generalist because various task-specific layers are still involved while operating the model.

Some other research [98, 120] re-frames the datasets for different tasks into the same format. MQAN [99] formulates all the datasets into question answering over context. For instance, for sentiment classification, the "question" was "Is this sentence positive or negative?", the "context" was the sentence, and the "answer" is the label of the sentence. However, it did not include any regression task. After re-framing datasets, MQAN leveraged two bi-directional LSTM encoders to encode "question" and "context". Then the dual coattention and compression layer generated the representations for both "question" and "context". A multi-pointer-generator decoder was used to decide whether the "answer" should be copied from the "question" or "context" or generated from a limited vocabulary.

T5 [120] created a shared Text-To-Text framework for the tasks by using both the encoder and decoder in the Transformer with some changes. The framework suggested using the same model, the same loss function, and the same hyperparameters on all the NLP tasks. The model's input was re-organised so that the model could recognise which task it belonged to. The output of the model was the text version of the desired output. For example, to translate a sentence from English to German, the input was formatted as "translate from English to German: This is a sentence.". T5 was pre-trained with Colossal Clean Crawled Corpus. During fine-tuning, multi-task T5 was trained on different tasks jointly. The multi-task T5 underperformed the T5 that was fine-tuned separately for individual tasks on the GLUE and superGLUE [150] benchmarks.

All these models focus learning and evaluating on general language understanding tasks like GLUE [151], superGLUE [150], and decaNLP [99]. In contrast, we focus on one specific domain of text: the Questions.

3.2 Sentence Embedding

Embedding is important in NLP. It converts text into a form that learning algorithms can understand: numbers. There are multiple approaches to representing text, which, have a critical impact on the performance of machine learning algorithms. A piece of text can be represented by character level, word level, or sentence level.

Word embedding and sentence embedding have become more prevalent in recent years. They can effectively identify similarities between words and sentences. Sentence embeddings, which capture the individual word meanings and their relationships within a sentence, offer a more holistic comprehension of the text than word embeddings. Sentence embeddings can be applied to tasks including machine translation, information retrieval, clustering, and semantic textual similarity.

Our research intends to capture the relation between sentences using vector representations. Thus, we apply and fine-tune sentence embedding to our question processing models.

There are various learning methods for sentence embedding. Most of them are evaluated on certain tasks. For example, sentiment/opinion classification tasks, sentence similarity tasks, and multimodal tasks. Li et al. [88] divided current sentence embedding models into five categories: word collection models, pragmatic coherence methods, semantic comparison methods, hybrid multi-task models, and pre-trained contextual models. In this section, we will give a brief overview to the history and background of sentence embedding following this category.

Word Collection Models

Word collection models were introduced in the early stage of sentence embedding research. They are constructed from averaging or concatenating pre-trained word embedding. Although averaging word embeddings is a relatively simple method, it provided a "hard to beat" baseline for many evaluation tasks [37, 151].

Wieting et al. [158] proposed two simple averaging models with LSTM. For a set of word representations $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the averaging model generated the sentence embedding by taking the average of word embeddings, such as $\frac{1}{n}\Sigma_{i=1}^{n}\mathbf{x}_i$. And another model learned a projection matrix

$\mathbf{W}_p$ and a bias vector $\mathbf{b}$ on top of the averaging model. They found that these two methods outperform other LSTM models in most semantic evaluation tasks.

Another word collection model, Weighted Removal [8], computed a weighted average of the word embeddings as the sentence representation. The weight is computed as $a/(a+P(word))$, where a is a defined parameter for adjusting the tradeoff between topic-related works and high-frequency words, and P(word) is the unigram probability of the word. Weight Removal bypassed the simple averaging models [158]. However, there is a common limitation in averaging models, they ignore the order of the words, which turns out to be critical for the performance on many tasks [97, 171].

Pragmatic Coherence Methods

Pragmatic Coherence Methods learn sentence embedding with the help of grammatical and lexical cohesion and contextual knowledge. Usually, adjacent sentences are leveraged during training. This enables the sentence embedding to capture the context information.

In 2015, Kiros et al. [77] proposed an unsupervised approach to train the sentence embedding called Skip-Thoughts. It was similar to the skip-gram model [56]. Rather than predicting the surrounding words, the model was trained to predict surrounding sentences. The model consisted of one GRU encoder and two GRU decoders to generate the previous and the following sentence of the target sentence. Skip-Thoughts created a leading performance at that moment on multiple classification tasks. However, it is slow to train using a recurrent-based network because the model takes one word at a time.

Based on Skip-Thoughts vector, Quick-Thoughts [94] utilised a GRU encoder to embed the sentence. Nonetheless, instead of using the decoders to reconstruct the adjacent sentences, Quick-Thoughts applied a classifier to predict the next sentence from a set of candidates. This could improve the speed of the training and make Quick-Thoughts more competitive in training with large-scale datasets. Quick-Thoughts obtained better performance than Skip-Thoughts on classification tasks [66, 89, 108, 109, 140, 157].

Semantic Comparison Methods

Semantic comparison methods focus on the difference between connotations in two sentences and usually contain two encoders / Siamese networks during the training process. Models are trained with annotated natural language inference datasets [22, 159].

InferSent [36] is a supervised sentence embedding method trained with Stanford Natural Language Inference Corpus (SNLI) [22]. InferSent was constructed with two identical bi-directional LSTM encoders with max pooling for mapping the sentence pairs into embedding vectors. The relation between these two embeddings was extracted by concatenation, element-wise product and absolute element-wise differences. The resulting embeddings that captured the relationship between two input sentences were used to predict the categories (neutral, contradiction, or entailment) of the sentence pair.

Multi-task Models

Unlike pragmatic coherence methods that focus on modelling the context, and semantic comparison methods that focus on semantic similarity, multi-task models are learned without task-specific objectives.

Subramanian et al. [143] introduced an MTL model: MILA/MSR, which tried to generalise different objectives in the same training scheme. MILA/MSR model was built up with a bi-directional GRU network and trained on Skip-Thoughts sentence prediction task, neural machine translation, constituency parsing, and natural language inference. The research showed that adding more tasks improved the performance of sentence embedding. Furthermore, fine-tuning pre-trained MILA/MSA with task-specific data led to better results than complex supervised approaches using attention mechanisms. The generalised model outperformed InferSent while adding more data.

Following the same approach, Cer et al. [27] used a Transformer network [148], which trained on various tasks to produce the universal sentence embedding (USE-T). The encoder was trained by next sentence prediction, conversational input-response [63], and natural language inference tasks. The results showed that USE-T had a strong performance on various evaluation tasks. Nevertheless, it traded off the compute and memory usage. Besides, the compute time for USE-T noticeably increased with the sentence length.

Pre-trained Contextual Models

Different from other categories, pre-trained contextual models arise with the idea of the pre-trained language model. Some of the models in this section were not initially trained for sentence embedding. However, some outputs from these models can be recognised as sentence representations and result in strong performances in language processing tasks.

In 2018, the development of BERT [43] refreshed the leaderboard for multiple NLP tasks [121, 151, 159]. It used the encoder part from the Transformer architecture to learn the representation of the input word tokens with two tasks: masked language model and next sentence prediction. Then the learned model could be fine-tuned for downstream tasks. There are multiple ways to get a sentence representation from BERT. For example, by taking the average of the learned word representations or using a special [CLS] designed for classification tasks and considered to contain the information of the entire input. After BERT, many models were introduced based on BERT, such as RoBERTa [93] and ALBERT [82]. And they kept improving the performance on the downstream NLP tasks.

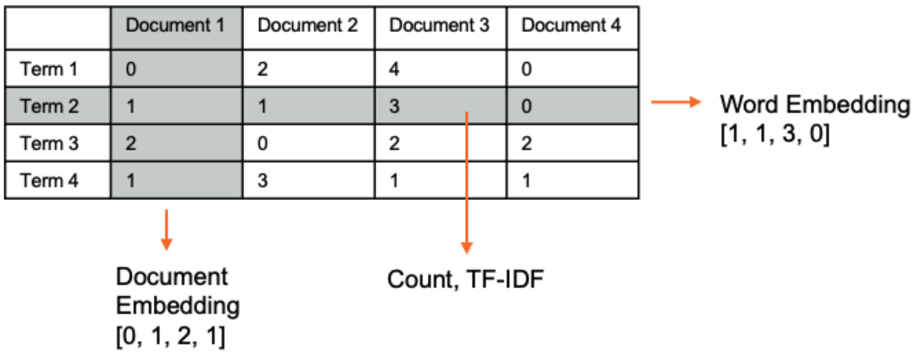
Despite the significant performance made by BERT and its variants, it was inefficient to use BERT for producing fixed-length sentence representations. Therefore, sentence-BERT [125], a

BERT-based Siamese framework, was trained to capture sentence similarity using the semantic comparison method. One version of sentence-BERT resulted in better performance among both classification and sentence similarity tasks [2–5, 40, 97] compared to existing sentence embedding methods.

The review shows that pre-trained contextual models are taking the lead in sentence embedding methods. Also, adding more tasks can potentially increase the generalisation of sentence representations. In our research, we aim to leverage sentence embedding to represent questions and capture the relation between questions and other facts, such as topics and answers. In the end, these relations can be determined by semantic comparison. Hence, we will use sentence-BERT as a foundation for training our question representation models. More details of BERT and sentence-BERT will be explained in section 4.1.

3.3 Embedding Compositionality

Text embedding has been developed in artificial intelligence research constantly. Statistical methods were widely used in the early years to represent the text. For example, as shown in Figure 3.1, a word embedding for a term was built up by counting its occurrence in each document. The counting vectors were then replaced by a TF-IDF method [71, 95], which is still a statistical measure of the importance of a term to a document from a set of documents. Another method, word co-occurrence statistics, was once popular as well. It was introduced based on the logic "words with similar meaning will appear in similar contexts".



	Document 1	Document 2	Document 3	Document 4
Term 1	0	2	4	0
Term 2	1	1	3	0
Term 3	2	0	2	2
Term 4	1	3	1	1

Figure 3.1: Count matrix for word representations.

So far, all representations are explainable, meaning each value in the vector has a purpose. However, the growing number of vocabularies and documents requires a huge memory for storing the matrix. Moreover, these representations can become sparse, so the matrix can contain a huge amount of invaluable information, such as 0 in the vectors. Therefore, more complicated methods, such as neural networks, are used to generate dense embedding. In 2013, Mikolov et al. introduced word2vec [102], a word embedding method based on both co-occurrence of words and neural network and became the state-of-the-art at the time. By using a neural network, the embedding became challenging to explain, and the values in the coordinate are no longer meaningful.

Even though dense embeddings learned directly from the end-to-end deep learning models achieve good performance and are widely utilised not only in NLP but also in other types of representation learning. It has a limitation of extracting the hidden attributes carried in representations. As a result, Disentangled Representation Learning (DRL) [14] was introduced to recognise and separate the underlying attributes that are concealed in the representations of the observable data. A disentangled representation can be broken down into smaller parts. This can improve the interpretability of learned models, as each building block of the latent space corresponds to a distinct feature or characteristic. Furthermore, the disentangled representations make it easier to control and manipulate the data representation.

Learning and decomposing the disentangled representation provides a new aspect to better comprehend black-box models and transfer learned knowledge to new tasks. Understanding and eliminating biases in machine learning models can also benefit from disentangled representations. It may be easier to identify and adjust for variables that relate to sensitive traits such as race or gender by decomposing the factors of variation.

DRL can be used in a variety of machine-learning applications. For example, in computer vision, DRL can be applied to disentangle the image representation into scene structure and artistic appearance. The style feature can then be used in another domain [86]. Other fields like graph learning [96] and NLP [11] also apply to disentangle representations to boost the performance of downstream tasks.

In NLP, one particularly interesting phenomenon people observed in word2vec embeddings is that after enforcing "co-occurrence" relations between words while building the embeddings, semantic and even grammatical relations are also represented by the embeddings. This may be regarded as a form of "compositionality", meaning the vector representation is made up of its elementary parts. The question is whether "compositionality" exists in the sentence embedding.

The compositionality found in word embeddings, motivated our investigation in sentence embedding. The research from Mikolov et al. [101] highlighted that word embeddings can present compositionality, to such an extent that simple analogies can be performed in that representation, as in the standard examples:

- $\Phi(germany) - \Phi(berlin) \approx \Phi(france) - \Phi(paris)$
- $\Phi(waitress) - \Phi(waiter) \approx \Phi(actress) - \Phi(actor)$

Shwartz and Dagan [137] assessed the word representation compositionality by six tasks that addressed the meaning shift and implicit meaning phenomena: (1) given a verb followed by a preposition, determine if it is a verb-particle construction; (2) identify if a verb is a light verb construction; (3) given a noun compound with two nouns, determine if the meaning of the target word in the compound is literal; (4) determine if a phrase can describe the noun compound; (5) given an adjective-noun composition in a sentence, determine if an attribute is implicitly conveyed; (6) Given a sentence, determine the BIO tags for each token. Their research showed that contextualised embedding based on pre-trained language models was compositional.

Andreas [6] gave an idea on measuring the compositionality by how well the observed representation can be approximated by composing the representations of inferred primitives. They introduced a method TRE to measure the multiply composition: $\Phi(Observed) = \Phi(Primitive_a) \times \Phi(Primitive_b)$. The method measured compositionality between the visual descriptive phrase [135] and the representations of coloured number images from Colored MNIST dataset [7] by the vector distance. However, we are more interested in the compositionality that can be captured with a learned sentence embedding in an additive form.

3.3.1 Compositionality in Sentence Embedding

There is an increasing amount of research focusing on evaluating the compositionality in sentence embedding. There are two main approaches: task-based and task-independent. Task-based methods measure the compositionality by evaluating the performance through specific language features, such as semantics, synonym, and polarity. The performance on these tasks defined the compositionality of sentence embedding.

Ettinger et al. [50] constructed a dataset for identifying semantic role and scope within the embedding. The sentence embedding was required to determine the entity-event relation for the semantic role. For instance, classifying if “professor” (entity) is an agent of “recommend” (event). For semantic scope, Ettinger et al. created data with negation by changing the sentence’s meaning while holding its lexical content relatively constant.

Dasgupta et al. [41] generated a new dataset to understand how words combine in the embedding. To do so, pairs of sentences with different natural language inference relations were generated by changing the order of the words or adding an extra word. There were three types of changes within the dataset: same type (only change the order), more/less type (add the word “more/less” and change order), and not type (add “not”).

Bhathena et al. [17] defined that a sentence representation with good compositionality should be able to sense the change of polarity of the sentence even though only one sentiment word switches while other content remains the same. A polarity sensitivity scoring $PSS = \frac{1}{n}\Sigma_{i=1}^{n}\mathbb{1}[\hat{y_s} = y_s \wedge \hat{y_{s'}} = y_{s'}]$ was introduced to evaluate the embedding compositionality. Where $\hat{y_s}$ is the predicted label for sentence s, y_s is the ground truth, and s' is the sentence that changes the polarity. They evaluated the compositionality of various encoding models [28, 114] and showed that BERT has a stronger computational understanding.

These task-based methods targeted revealing the language understanding level of the sentence representation. Another approach, the task-independent approach, looks at the compositionality of general auxiliary tasks, such as sentence length, word content, and word order [1].

These tasks do not require task-specific labelled data, hence, task-independent.

Adi et al. [1] introduced three methods that can evaluate any sentence embedding model. Length task measures how well an embedding can predict the length of a sentence. Word-content task measures whether the sentence embedding can identify a word within the sentence by giving the concatenation of the sentence and word representations. Word-order task determines the order of two words within the sentence by giving the concatenations of the sentence and two word representations. They applied the evaluation framework on sentence embedding achieved from an encoder-decoder model and found that the LSTM auto-encoder was good at word-content and word-order tasks.

The research mentioned in this section aims to measure the compositionality within sentence embedding. Moreover, some of them showed that some of the existing sentence embedding methods could reveal high-level primitive elements without pointing them out during the learning process, in other words, compositional. Nevertheless, no existing research attempts to break down sentence embedding into its attributes. Although Adi et al. tried to identify if the building blocks of a sentence, words, were captured by the sentence embedding, the method they used to obtain the word representation was the same as the sentence embedding. Besides, the relation between these word representations and their corresponding sentence embedding remains unknown.

As a result, in our study, we intend to decompose sentence embedding into word representations and understand if words are the attributes for sentence embedding. Furthermore, the word representation learned from the existing sentence representations can deduce a new sentence embedding. We measure the compositionality by the vector space distance between the actual sentence embedding and the inferred vector that builds from the property vectors [6].

3.4 Question Processing

In the previous research, question processing is considered a process in some early question answering systems [52, 89]. This process includes query formulation, answer type detection and keyword extraction. The question processing module intends to improve the performance of the question answering system. Nevertheless, questions are always a special type of text. In education, a question is used to reflect the student's ability and concerns [32]. In the meanwhile, question data have become popular and more accessible with the growing use of voice assistants and online query communities, which gives an opportunity for applying machine learning techniques to analyse the questions. Therefore, it is interesting to define question processing with regard to data analysis and find out what we can learn from it.

NLP research pays attention to three types of question-related tasks: question classification, measuring question similarity, and question answering. Question classification is a specific case of text classification. Question similarity is similar to measuring text similarity. And question answering is a combination of text classification, similarity measuring, and information retrieval.

In this section, we will explore the research and methods based on these three tasks.

3.4.1 Question Types and Topics

Topic classification/modelling is a classic field in machine learning and NLP. Different approaches have been successfully implemented in the field of question classification as well, including the statistic approach, machine learning approach, and deep learning approach.

LDA (Latent Dirichlet Allocation) [18] is one of the most popular statistical models for unsupervised topic analysis. It classifies text in a document to a particular topic based on the co-occurrence patterns of words in the document. In the case of short text, such as Twitter, LDA failed to achieve a successful performance because the data sparsity affects the efficiency of the model [60].

Another approach for topic classification is using supervised learning models. A traditional pipeline for supervised text classification includes text, pre-processing, feature extraction, classifier, and output. This pipeline has been proven effective in many different areas [69, 142]. With the development of the deep neural network, the modern pipeline no longer requires pre-processing and feature extraction. Instead, it contains an embedding process used to learn and represent the text with a dense vector. As explained in this chapter, many methods exist for representing and understanding the text. Their combinations have achieved state-of-the-art results in various text classification tasks, including topic labelling [144].

As a result, instead of focusing on the existing machine learning methodologies for categorising questions, this section will discuss what kind of taxonomy people use on questions in both education and machine learning.

One of the most famous methods to categorise the questions for teaching is Bloom's taxonomy [19]. It divided the questions into knowledge, comprehension, application, analysis, synthesis, and evaluation. The taxonomy was generated in a way to help students learn. Some people [147] divided questions into four levels, which reveals the depth of learning. Level 1 to 4 indicated summary/definition/facts, analysis/interpretation, hypothesis/prediction, and critical analysis/evaluation/opinion, respectively. These methods provide a theoretical approach to classifying the question. They give an idea of what people expect to know from the question under a general educational setup. There are no existing labelled question datasets for these taxonomies, and the questions can only be automatically classified based on some rules. For instance, facts are definition questions that are more likely to be asked with keywords "who" and "what" etc.

In machine learning, question classification used to be widely utilised to improve question answering performance. The questions are classified based on the expected type of answer [100]. Early work like Lehnert's taxonomy [87] had 13 conceptual classes for the questions, such as verification, causal consequent, disjunctive, and so on. However, it did not focus on categorising the question in a semantic way, which we are more interested in while analysing the questions.

Recent work like TREC [89] focused on factual questions and created a hierarchical taxonomy that separated the questions into 6 coarse classes (abbreviation, entity, description, human, location and numeric value) and 50 fine classes (animal, body, mountain, order etc.). However, to further analyse the comprehension content from the questions' complexity, the question type categorisation needs to be general, including factual as well as non-factual, regardless of the theme. Besides, even though Li and Roth introduced a two-layered taxonomy, it did not reflect the theme of the questions. For instance, class NUMERIC-PERCENT indicated that the expected answer is a percentage instead of the theme the questions are related to.

Zhang et al. [169] introduced a question topic dataset originating from Yahoo! Answer. They categorised the questions into ten topics according to the top 10 popular topics from Yahoo! Answer websites. The topics included science, society, technology, and others. This taxonomy was able to show the theme of each question.

Mohasseb et al. [106] proposed a grammar-based question classification framework. The questions were divided into six types: confirmation, factoid, choice, hypothetical, causal, and list questions. The way they classified the question can be related to the educational taxonomy and somehow reflect the complexity of the question. In this thesis, we will not train the model to predict the exact types as in the reference [106] because the dataset is not publicly accessed.

To provide an in-depth analysis, we define question taxonomy with two separate methods: type and topic classification. The question type intends to show the nature of questions, which can be leveraged to further analyse the complexity of the questions. We will classify questions on the basis of the "interrogative word", a function word used to ask a question, such as wh-words. For the question topic, we follow Zhang et al.'s taxonomy [169] and aim to demonstrate the theme of the questions.

3.4.2 Similar Question Classification

Question similarity is a sub-field of measuring sentence similarity. Sentence similarity can be referred to lexical similarity that compares the words occur in different sentences. However, in this thesis, we particularly look at semantic similarity which compares the meanings of the sentences with different wordings.

There are various techniques for comparing sentence semantic similarity. Chandrasekaran et al. [29] distinguished the semantic similarity methods into knowledge-based, corpus-based, and deep neural network-based methods.

The first approach is knowledge-based methods that use a knowledge source for words like lexical databases [104], dictionaries, thesauri etc., to measure the similarity. The underlying knowledge base provides a structured representation of terms connected by semantic relations for these approaches. It provides an unambiguous semantic measure as the actual meaning of the terms is taken into account [130]. Knowledge-based methods highly rely on the knowledge source. Thus, it requires computational resources to update the underlying source frequently.

Besides, although lexical databases like WordNet exist for English, it is difficult to obtain similar resources to implement the knowledge-based method for other languages and domains.

The corpus-based approach measures semantic similarity by utilising the information from large corpora. Vector representations learned from the corpora are used to encode the text. This process is also known as embedding. As we discussed in the previous sections, the embedding methods used to capture sentence similarity have evolved in the past decades, from word embedding [102, 113] to sentence embedding [8, 36] to deep contextualised embedding [115, 125]. This leads to the final method to measure semantic similarity, deep neural network-based methods. Among all the deep learning models, pre-trained language models [43, 82, 93] created a top performance in capturing semantic similarity. More details were explained in section 3.2.

While measuring the similarity between sentences, the distance between the sentences is calculated in a vector space. The most common metric used to measure semantic similarity is to calculate the length distance between vectors, such as Euclidean distance, cosine distance, and Manhattan distance. The distance is inversely proportional to the relationship. Cosine distance/cosine similarity are primarily used among NLP researchers [105].

3.4.3 Question Answering

Question answering is always a challenging research task in information retrieval [81] and natural language understanding. A question answering system aims to answer questions raised by humans in a natural language automatically.

The research for question answering can be traced back to the 1960s. Early question answering systems can be categorised into four different systems [138]: list-structure database system [55], graphic database system [78], text-based system [139], and logical inference system [20]. These systems only worked on a specialised subset of English and were mostly rule-based.

Nowadays, most question answering systems are developed based on machine learning / deep learning models. Moreover, they are usually trained on large-scale datasets [45, 72, 103, 121]. Research in open-domain question answering has been highly successful in recent years. An open-domain system can deal with questions without identifying a specific domain. It mostly focuses on factoid questions that can be answered using the given "context" (namely open-book). Wikipedia is commonly used as a knowledge source in academic research because it is general and contains a large scale of facts from the real world, which can be challenging for a question answering system. There are four main approaches for question answering systems.

Traditional Question Answering Pipeline

A traditional question answering system mainly contains three stages: question analysis, document retrieval, and answer extraction [110].

The question analysis assisted document retrieval and answer extraction in achieving better performance. A typical question analysis module contained query formulation, which used POS tagging [80, 81], stemming [81], parsing [80], and stop word removal [23, 51] to extract keywords in the queries; and question classification that identified the type of the answer [89].

In the document retrieval stage, the system filtered the irrelevant documents and kept only a small group of documents that might contain the correct answer for the next stage. In the past decades, the documents were retrieved with the Boolean model, vector space models, probabilistic models, etc.

The answer extraction stage is responsible for giving a precise answer to the given query. This stage heavily relied on the question analysis results, especially name entity recognition, in some early studies [80, 107]. Matching methods such as word or phrase matching [81] and syntactic structure matching [80] were widely used to extract the answer from the given contexts.

In recent research, deep learning approaches have been primarily used for the question answering system. Since the neural network simplifies the process of feature selecting, the question analysis stage is replaced by leveraging a vector representation to encode the query.

Retriever-Reader/Retriever-Generator Approaches

As the name of this approach indicates, a retrieval-reader/retrieval-generator approach first retrieves the relevant passage from a knowledge source, and then extracts/generates an answer span from the passage.

In 2017, Chen et al. [30] proposed DrQA, the first neural open-domain question answering system. The system contained a document retriever that retrieved relevant documents from Wikipedia and a document reader that extracted the answer from the document. The document retriever was a TF-IDF weighted term vector model and was not trainable. The document reader was constructed with multiple bi-directional LSTM layers to predict the span of tokens that is most likely the correct answer, which was considered a reading comprehension task.

BERTserini [166] improved DrQA by utilising an Anserini retriever [165] and a BERT reader. The scoring function depended on both the retriever and the reader. It increased the percentage of correctly answered questions from 27.1 to 38.6 on SQuAD dataset [121].

Other methods focused on ranking the retrieved passages with distant supervision [90] and reinforcement learning [153]. Training a re-ranker could help further identify whether a document is relevant or not. And the results showed that the re-ranker improved the performance of the question answering system. By contrast, Wang et al. [154] trained an answer re-ranker to rank the answers extracted from all relevant documents.

So far, all the models have applied a statistical method for the retriever. The retriever was not trainable while training the question answering system. The question was if the retriever could be trained jointly with the reader.

End-to-End Learning

The end-to-end approach was designed to train the retriever and reader simultaneously as one whole system. Usually, it uses a dense vector representation to represent the query, answer, and evidence instead of a sparse representation. Despite the fact that it has been proven that using a learned dense vector such as word embedding or sentence embedding can increase the performance in various NLP tasks [27, 102, 113, 125], dense vectors never showed a better performance than sparse representation before 2019 in question answering. With the increasing number of queries and knowledge sources used to train the question answering system, it becomes challenging to encode, store, and index these texts. On the other hand, these large amounts of labelled query-document pairs enable the dense embeddings to be trained and utilised in the question answering system. Moreover, the latest technique and tools, such as Faiss [70], supports fast inner product search between embeddings.

Lee et al. [85] introduced ORQA, which is the first model to learn retriever and reader jointly. Both reader and retriever were trainable with three BERTs for the question, evidence block, and reader, respectively. During training, the model only learned from question-answer pairs without reading comprehension datasets. And the retriever was pre-trained with a new task, Inverse Cloze Task, that treated a sentence as a pseudo-question and its context as pseudo-evidence. The goal was to select the right content from a set of random options. The results demonstrated that ORQA achieved better performance on genuine information-seeking questions.

REALM [57] applied pre-training on both the retriever and reader with a masked language model. They masked only name entities and dates in the query, document, and other unlabelled pre-training corpora. Pre-training did improve the results of answering the questions. Nevertheless, Karpukhin et al. [75] argued that pre-training was not always necessary, especially on a larger dataset. For smaller datasets, the model could outperform REALM by mixed training with the bigger datasets.

Seo et al. [134] tried to encode and index the evidence at phrase level instead of paragraph or document level so that the model did not require an explicit reader. They encoded the queries and answers independently with BERT. Encoding at the phrase level reduced the inference time by trading off some accuracy. However, the independent encoding missed the attention between query and evidence. Thus, Seo et al. applied a sparse TF-IDF vector to the query to represent the relationship with the evidence. Lee et al. [83] improved the model by applying contextualised sparse representations that have different weights for different sparse terms. It outperformed DenSPI [134] on curatedTREC [12] and SQuAD [121].

Retrieval-Free Model

Language model like BERT has played an important role in the end-to-end models. The language model is pre-trained with Wikipedia or other large corpora, and the key question is if it can act as a knowledge source and obtain the answer directly without external evidence (known as close-book question answering) [116].

Previous research [116, 127] showed that a large amount of knowledge learned from large-scale unstructured corpus can be stored in the underlying parameters. For instance, GPT-2 [118] was able to generate the correct answer with only the question. Furthermore, compared to previous state-of-the-art fine-tuning methods, GPT-3 [24] achieved competitive performance in few-shot learning sessions. The results from another model, T5 [127], showed that the model size highly impacted the performance of retrieval-free models. The T5 with 11 billion parameters obtained similar performance to the DPR model [75] that had only 330 million parameters. Overall, retrieval-free models still had an impressive performance on various benchmarks.

3.5 Chapter Summary and Conclusion

This chapter reviewed the literature on recent techniques used to understand and represent natural language. We also looked at the applications of these tools and techniques in one specific field: question processing.

We notice that the path of the language model, sentence embedding, and question processing is synchronised with the development of the neural network. The performance of these models keeps improving by using a more complicated network structure and larger corpus. Undoubtedly, the rise of the pre-trained language model has changed the status quo in NLP research and brought machine intelligence to a new state.

Question/topic classification, text similarity, and question answering are used to be three different directions in NLP. Nevertheless, with the pre-trained language model and multi-task learning approach, the same architecture can deal with three different tasks, potentially more tasks in other domains and data in other formats (image, audio, video, etc.).

In the meanwhile, new challenges occur. With deep learning-based methods and complex structures, we need an approach to interpret the reason behind the machine's decision. Thus, we also explored some existing approaches for understanding the composition of dense sentence embedding. Inspired by word embedding compositionality research, we raise the concern that no current research in sentence embedding compositionality tries to break down the dense vector representation into attribute blocks. Decomposing the sentence embedding could benefit the research in detecting unwanted bias in the representation, explainability of AI decisions based on these representations, and the possibility of performing analogical reasoning or counterfactual question answering.

Furthermore, the reviews showed that the performance of the models is highly dependent on the scale of the model and data. It is computationally expensive to train a separate model for each task. Hence, training one model for multiple tasks has the potential to create a generalist and improve the accessibility of AI. A generalist is expected to be good at many problems but is not required to excel at any task.

In the next chapter, we will introduce the methodologies used to address these problems in this thesis, including a decomposition method that breakdown the sentence embeddings and a multi-task question processing model that generates shared representation for various question-related tasks.

Chapter 4

Compositionality in BERT Sentence Embedding

In the previous chapter, we reviewed the work on existing pre-trained language models and sentence embedding. We noticed that one Transformer-based model, BERT, had established a new baseline for various language processing tasks and sentence embedding. Hence, in our studies, we apply BERT as our foundational framework to generate a sentence representation for NLP. This chapter gives an in-depth introduction to BERT and how it maps sentences into vector representation that can reveal the semantic similarity of sentences.

After reviewing the compositionality studies in the domain of word embedding and sentence embedding, we also raised our concern that the properties within the sentence embedding remain unexplained with the current methods of measuring compositionality. Therefore, inspired by research in word embedding, this chapter presents a linear system that can be utilised to decompose sentence embedding into its attributes. Moreover, these attribute vectors can perform analogy reasoning.

In mathematics, a set of items and their relationships can be described by representing those items as points in a vector space. The relations of interest between those objects are thereby approximated by geometric or algebraic relations between those coordinate vectors. The process that models items as vectors is called "Embedding". With embedding, standard algebraic techniques can be used to perform inferences on the data. The distance between two vectors is often used to represent the presence of a relation, and their coordinates are chosen by an algorithm on the basis of the relations that we wish to incorporate.

Embedding is widely used in representing language. Various algorithms exist, based on different principles, to calculate “embedding vectors” for words and sentences. Before BERT was widely applied to generating representation, the word representations used to highly rely on the co-occurrence matrix, such as FastText [21], word2vec [102], and GloVe [112]. The value of the coordinates is calculated based on how often a word occurs and how often this word occurs with other words in a given text corpus. Even before that, people used to represent the word by counting the occurrences of words in the text. How these vectors represent the words can be easily explained by going through the elements within the vectors. However, while BERT and its related networks improve the performance on NLP tasks, the whole process, including embedding, becomes a "black box" that is harder to interpret. Each coordinate's values are no longer considered to be meaningful in an embedding. They cannot be used to interpret what information is being used by the AI system. This considerably limits the possibility of explaining the decisions of a system.

In this chapter, BERT sentence embeddings are used as an example for measuring compositionality. We develop and demonstrate a series of tests to investigate the attributes that build up the BERT sentence embedding [163]. A simple sentence dataset is created for this purpose. The sentence embeddings generated from these simple sentences are then decomposed with a linear system into phrase representations. With the phrase representations, we can perform analogy and reconstruct the embedding of a sentence which has never been seen before.

The reconstructed embedding has 98.44% similarity with the actual BERT embedding. Furthermore, the reconstructed embedding is able to retrieve its corresponding BERT embedding with 99.5% accuracy.

4.1 BERT and Sentence Embedding

To embed the sentence, we apply a pre-trained sentence embedding model SBERT [125], which is a sentence embedding version of BERT [43].

BERT is a pre-trained language model that follows up the architecture of the Transformer [148] encoder and generates bidirectional encoder representations for input tokens. It can be used on various language understanding tasks, such as question answering, natural language inference, sentence classification, etc.

Different from Transformer, in BERT, the network can take a sequence pair as input by separating two input sequences with a special [SEP] token. The model breaks the input sequence into word tokens using a subword-based tokeniser. The input representation is the sum of three different embeddings. First is the word's pre-trained token embeddings called Word Piece embeddings [161]. Then the segmentation embeddings distinguish which sequence the token belongs to. Lastly, the position embeddings indicate the position of each token. Moreover, a special [CLS] token is added at the beginning of the input sequence.

For each input token, BERT generates a vector representation as the output, so that

$$\Phi_{BERT} : Input\ Representation \rightarrow \mathbf{x}$$

Where $\mathbf{x} \in \mathbb{R}^{d_{BERT}}$. In the BERT setup, $d_{BERT} = 768$. Similar words have closer vector in the vector space. The output vector of the [CLS] token is usually used for classification tasks because it can represent the information of the entire input sequence.

As a pre-trained language model, BERT is trained with two unsupervised tasks, **masked language model** and **next sentence prediction**, using existing corpus like BooksCorpus [170] and English Wikipedia.

For the masked language model, BERT takes sentences with words that are randomly masked with [MASK] tokens as input. BERT outputs the word vectors of all the input words. A full connected layer and a softmax function are used for every word vector to find the distribution of each word in the vocabulary. During pre-training, BERT minimises the cross entropy loss of the [MASK] words. The goal of the tasks is to predict the masked words instead of the whole sentence compared to an auto-encoder [149]. The masked language model helps BERT understand the bi-directional context within the sentence.

For the next sentence prediction, BERT takes two sentences as input, separated by the [SEP] tokens. BERT determines if the first sentence is followed by the second. These two tasks are trained simultaneously. The output of [CLS] token is used to classify if the second sentence follows the first. By doing this, BERT understands the context across sentences. Combining these two tasks, BERT can result in a good understanding of language.

Thereafter, the pre-trained BERT network can be fine-tuned for specific NLP tasks using different output layers. For instance, the output layer produces the start and the end position of the answer phrase for the question answering task; the output layer for the tagging task outputs the name entity recognition tag for each token. While training on the specific downstream tasks, the parameters in the encoder block of the model are slightly fine-tuned. The output layer is trained from scratch with a specific dataset. Therefore, training a specific task is fast once BERT is pre-trained.

There are two setups for BERT: $BERT_{BASE}$ and $BERT_{LARGE}$. $BERT_{BASE}$ contains 12 encoder layers, 768 hidden units, and 12 self-attention heads (110M parameters). And $BERT_{LARGE}$ has 24 layers, 1024 hidden units, and 16 self-attention heads (340M parameters). Due to the limitation of computational resources, we use $BERT_{BASE}$ in our experiments. We tried to pre-train BERT following Devlin et al. [43] at the early stage of our research. Nevertheless, with the computer power we had at the time, we failed to reproduce the same results presented in the reference. Therefore, we focused on fine-tuning BERT from the existing pre-trained models in this thesis.

BERT generates word representations that can capture word similarity. For sentence representation, the sentence with similar meanings will ideally be close to each other in the vector space. However, the [CLS] representation that reflects the information of the sentence fails to capture sentence similarity in such a way. Thus, we used SBERT [125], a version of fine-tuned BERT trained specifically for generating sentence representation. It also created a leading performance on semantic textual similarity (STS) task [26]. Figure 4.1 illustrates the examples of sentence embeddings generated by SBERT.

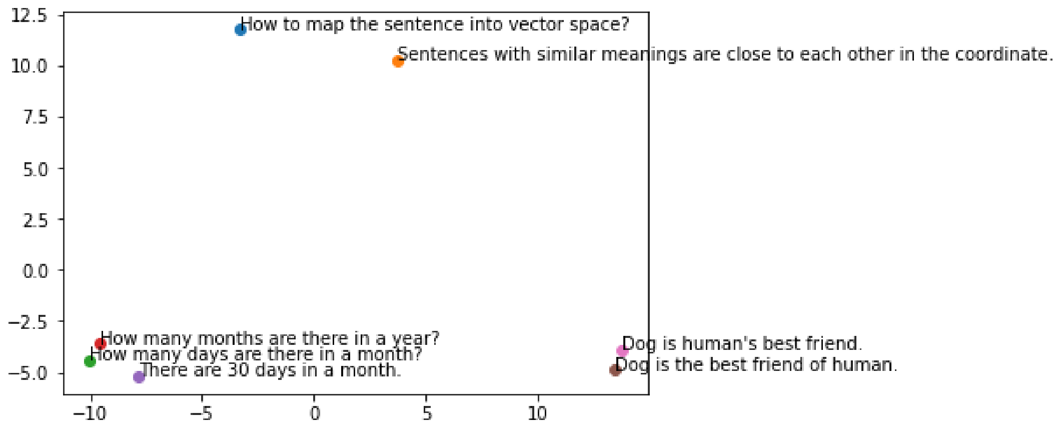


Figure 4.1: Sentence embeddings generated by SBERT. The embeddings are plotted with PCA in 2-dimensional space. The sentences with similar meanings are close to each other in the vector space.

SBERT introduces a Siamese structure as shown in figure 4.2. In such a way, BERT can be used to generate fix-sized sentence embedding that can be compared with cosine similarity. Furthermore, it reduces the time for finding a similar pair compared to the original pre-trained BERT [125]. With optimised index structure [70], SBERT can be used for similarity search in large-scale corpus, such as Wikipedia dump.

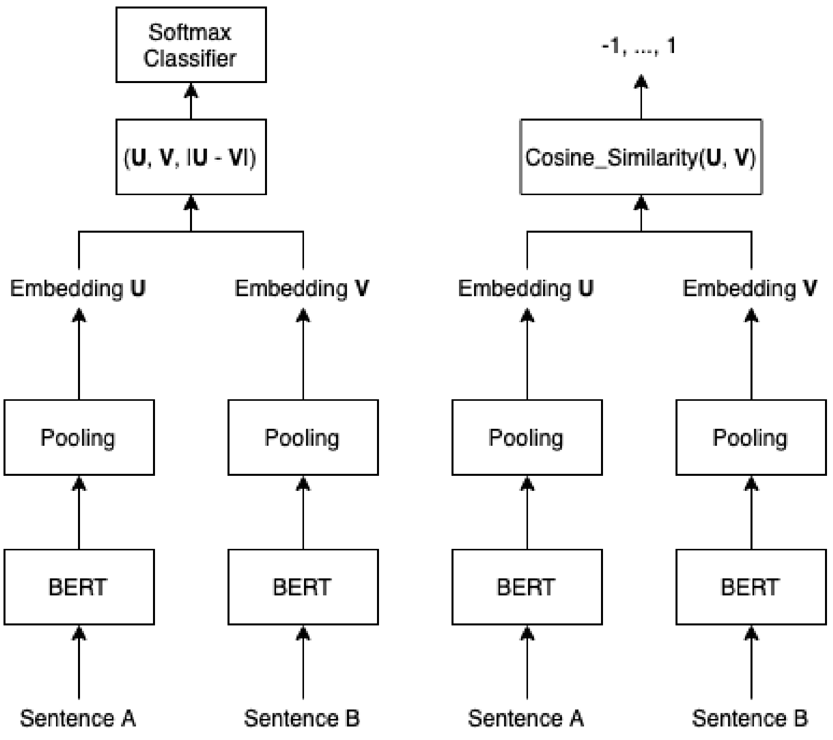


Figure 4.2: SBERT architecture [125]. The BERTs in the networks have tied weights. Left: Training, Right: Inference.

SBERT applies a pooling layer to derive the sentence embedding. There are three pooling methods used while training SBERT: (1) the output vector of [CLS] token; (2) the mean of all the output vectors; and (3) a max-over-time of the output. Reference [125] compared these three strategies and found that the mean strategy achieved the best performance.

During training, embedding **U** and **V** are concatenated with the element-wised difference $|\mathbf{U}-\mathbf{V}|$ before the softmax layer. SBERT was trained with natural language inference datasets (SNLI [22], Multi-Genre NLI [159]) by optimising the cross entropy loss. During inference, SBERT calculates the cosine similarity between two sentence embeddings. The sentences with similar meanings have a larger cosine similarity in the embedding space.

In this thesis, we use SBERT as an example to present our experiments on decomposing sentence embedding. Moreover, we fine-tune SBERT as a generalist that can perform multiple question tasks.

4.2 Decomposing Sentence Embedding with Linear System

As mentioned in section 3.3, it is difficult to understand the embedding generated by a deep neural network. In this section, we will propose a method [163] that can decompose an embedding to its building blocks.

When decomposing learned word embeddings like word2vec [101], research [102] found that the embedding can include both semantic and syntactic relationships between words, which can be revealed using simple linear algebra. For example, $\Phi(king)-\Phi(man)+\Phi(women)\approx\Phi(queen)$ (semantic) and $\Phi(dog)-\Phi(dogs)\approx\Phi(cat)-\Phi(cats)$ (syntactic). The question here is if sentence embeddings can be decomposed in a similar way. Sentences are compositional structures that are built from words. Therefore, it is natural to ask if the learned representations reflect the compositionality. We assume that there is an additive compositionality between words and sentences so that the sentence representation can be decomposed in terms of

$$\Phi_{BERT}(Sentence)\approx\Phi(Word_1)+\cdots+\Phi(Word_n)$$

As a result, we leverage a linear system to learn the word representations given a set of sentence embedding.

With a set of items I and its embeddings $\Phi(I)$, while decomposing the embedding, we look at the additive compositionality in the embeddings. Based on the definition of additive compositionality

$$\mathbf{b}_I=\sum_{i=0}^{n-1}\mathbf{x}_i$$

we postulate the existence of (unknown) component vectors that can be used as building blocks to create the representation of an item. This relation between the item and its component can be represented by a linear system. We solve the component vectors by finding the best fit for a linear system.

We will assume that the embeddings of a set of items are called **B**, and the unknown components that formed them are called **X**, while the information about which components are combined to form which item is stored in a composition matrix of coefficients **A**, so that

$$\mathbf{AX} = \mathbf{B}$$

Since the sentence is built up with words, we assume the sentence embedding is composed of word representations. A simple sentence in English usually comprises a subject, a verb, and an object. Therefore, assuming that there are three types of components i, j, k that determine the embedding of an item $\mathbf{b}_{i,j,k}$, we postulate the existence of some phrase vectors $\mathbf{x}_{i,*,*}$, $\mathbf{x}_{*,j,*}$, and $\mathbf{x}_{*,*,k}$ that added a contribution to $\mathbf{b}_{i,j,k}$ in terms of equation 4.1.

$$\begin{gathered}\Phi_{Composed}(I_{i,j,k}) = \Phi_{Composed}(Sbj_i) + \Phi_{Composed}(Verb_j) + \Phi_{Composed}(Obj_k)\\ \mathbf{b}_{i,j,k} = \mathbf{x}_{i,*,*} + \mathbf{x}_{*,j,*} + \mathbf{x}_{*,*,k}\end{gathered} \tag{4.1}$$

This can be written in matrix form $\mathbf{AX} = \mathbf{B}$ by introducing the binary compositional matrix $\mathbf{A}_{m,n}$, where $\mathbf{A}_{m,n}$ records whether a given building block (indicated by n) is present in a given item (indicated by m).

As the list of building blocks is formed by three lists, for three types of building blocks, the indicator n can be obtained from the indicators of these three lists. In the case where there are 10 elements for each type of building block, we can write: $m = 100 * i + 10 * j + k$.

$$\mathbf{A}_{m,n} = \begin{cases} 1, & \text{if } n = i \; or \; n = 10 + j \; or \; n = 20 + k \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

$\mathbf{A}_{m,n}$ indicates which three attributes to be chosen for constructing the sentence embedding.

Given a set of embeddings **B** for all items in a set of interest, we can find all the unknown vectors $\mathbf{x}_{i,*,*}$, $\mathbf{x}_{*,j,*}$, and $\mathbf{x}_{*,*,k}$ by solving the linear system $\mathbf{AX} = \mathbf{B}$. The linear system is illustrated in Figure 4.3.

$$\begin{array}{cccc} \mathbf{A} & \mathbf{X} & & \mathbf{B} \\ \begin{bmatrix} 1000000000 & 1000000000 & 1000000000 \\ \vdots & \vdots & \vdots \\ 0000000001 & 0000000001 & 0000000001 \end{bmatrix} & \begin{bmatrix} Attribute\ 1 \\ Attribute\ 2 \\ \vdots \\ Attribute\ 29 \\ Attribute\ 30 \end{bmatrix} & = & \begin{bmatrix} Embedding\ 1 \\ Embedding\ 2 \\ \vdots \\ Embedding\ 1000 \end{bmatrix} \end{array}$$

Figure 4.3: The linear system. Assuming there are three types of attributes. Each type includes ten different attributes.

However, the embedding might contain bias so that it cannot be perfectly broken down into component vectors. As a result, instead of calculating the exact solution for the linear system, we utilise the least square method to find an estimated **X** that best fits the linear equation. The process of the least square method is explained in equation 4.3.

$$
\begin{aligned}
L(I,\mathbf{X}) &= \|\mathbf{AX}-\mathbf{B}\|^2 \\
&= (\mathbf{AX}-\mathbf{B})^T(\mathbf{AX}-\mathbf{B}) \\
&= \mathbf{A}^T\mathbf{X}^T\mathbf{AX}-\mathbf{A}^T\mathbf{X}^T\mathbf{B}-\mathbf{B}^T\mathbf{AX}+\mathbf{B}^T\mathbf{B} \\
\frac{\partial L(I,\mathbf{X})}{\partial \mathbf{X}} &= -2\mathbf{A}^T\mathbf{B}+2\mathbf{A}^T\mathbf{AX}=0 \\
\mathbf{A}^T\mathbf{B} &= \mathbf{A}^T\mathbf{AX} \\
\mathbf{X} &= (\mathbf{A}^T\mathbf{A})^{-1}\cdot\mathbf{A}^T\mathbf{B}
\end{aligned}
\tag{4.3}
$$

We evaluated the linear system by calculating the loss (L) as equation 4.4. The smaller the loss, the more accurately the embeddings can be represented as a composition of components **X** (which is our definition of compositionality).

$$
L_{Norm} = \|\mathbf{AX}-\mathbf{B}\|^2 \tag{4.4}
$$

4.3 Experiments

To investigate the compositionality in BERT sentence embedding, we leverage a linear system outlined in the previous section to break down the sentence embedding into word representations. For the purpose of this investigation, we construct a sentence corpus of 1,000 simple sentences. Each sentence is made up of the three simplest elements needed to complete a sentence: subject, verb, and object.

4.3.1 Data Generation

For this study, we generated the sentence corpus with 30 components, equally divided into subject (Sbj), verb, and object (Obj). These components were combined into 10x10x10 $(Sbj, Verb, Obj)$ triplets. Then the $(Sbj, Verb, Obj)$ triplet makes up short, simple sentences with the same preposition and article. For example, for triplet (cat, sat, mat), the sentence is generated as "The cat sat on the mat.". There are 1,000 sentences in total. By creating the SVO sentence corpus, we can look into and understand each part we decompose with a linear system.

BERT tokenises the sentence into word tokens with a subword-based tokeniser. Some words such as "bookshelf" will be tokenised into two words ("book" and "shelf"). To keep all the sentences in the same number of tokens, we carefully picked the words we used to build the corpus. Note that BERT takes punctuation as an input token as well; there are 7 tokens in total for each sentence. The $(Sbj, Verb, Obj)$ candidates are shown in Figure 4.4.

For a given sentence (**I**), we notate the subject, verb, and object phrases with indices i, j, and k, respectively. Thus, $I_{i,j,k} = Sbj_i + Verb_j + Obj_k$. After that, the simple sentences are mapped

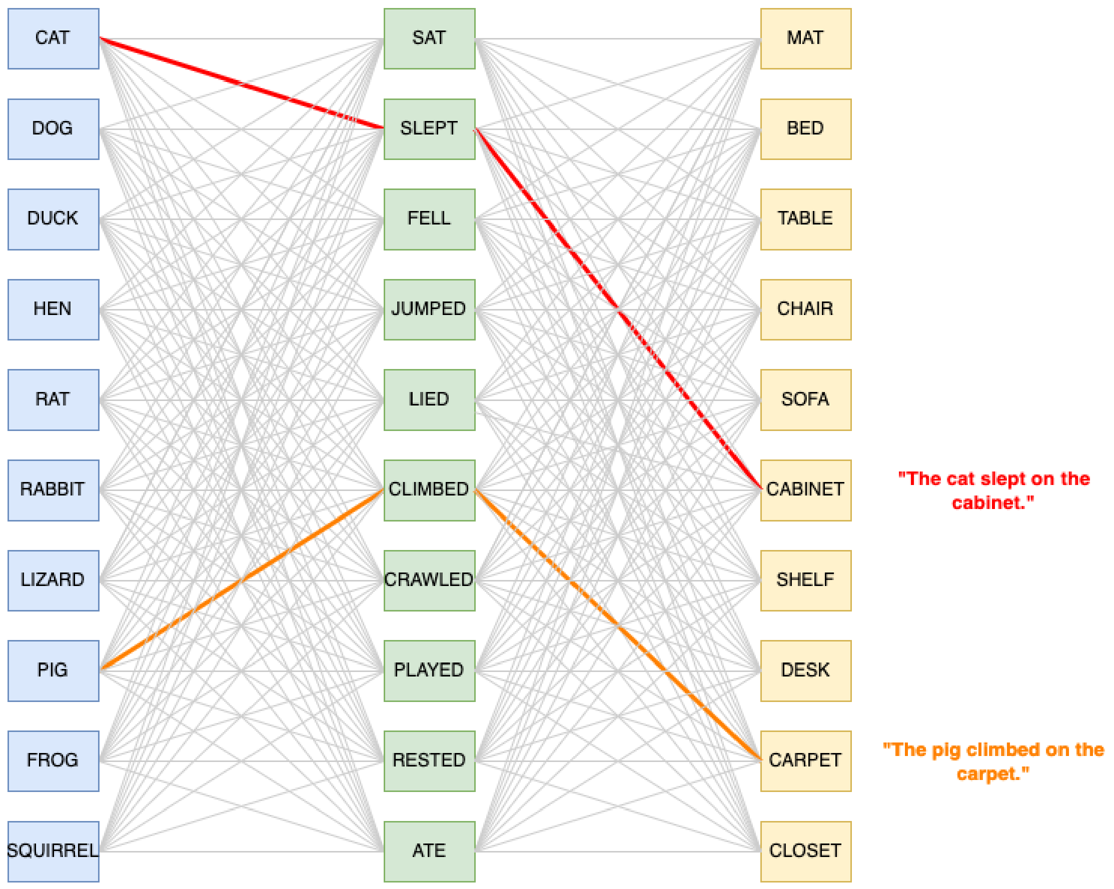


Figure 4.4: Sbj, Verb, Obj instances in the datasets. The sentences are build in terms of "$The'' + Sbj\,(Blue) + Verb\,(Green) +$ "$on\ the'' + Obj\,(Yellow)$. With the combination of all instances, there are 1,000 sentences in total. While decomposing the sentence embedding, "The cat" is considered the subject phrase; "sat on" is the verb phrase; and "the mat." is the subject phrase.

with a fine-tuned sentence BERT into a high dimensional space described in section 4.1. The distance between embeddings is able to describe the semantic similarity between sentences.

$$\begin{gathered}\mathbf{b}_{i,j,k} = \Phi_{BERT}(I_{i,j,k}) \\ \Phi_{BERT} : \mathbf{b}_{i,j,k} \to \mathbb{R}^d, d = 768\end{gathered} \tag{4.5}$$

Although the MEAN pooling strategy achieved the best performance for BERT sentence embedding [125], in this study, we use the vector representation of the [CLS] token as our sentence representation. MEAN pooling strategy takes the average of the token representations generated by BERT so that the sentence embedding is naturally built up with a linear combination. Besides, the vector representation of [CLS] tokens are also widely used to represent the input sequence for classification in BERT and its derived models. It will be interesting to interpret the attributes within it.

4.3.2 Solving the Linear System

Having computed a set $\mathbf{B}$ of embeddings for all our sentences, we can find the unknown phrase vectors $\mathbf{x}_{i,*,*}$, $\mathbf{x}_{*,j,*}$, and $\mathbf{x}_{*,*,k}$ by solving the linear system $\mathbf{AX} = \mathbf{B}$, where $\mathbf{A} \in \mathbb{R}^{1000\times 30}$, $\mathbf{X} \in \mathbb{R}^{30\times 768}$, and $\mathbf{B} \in \mathbb{R}^{1000\times 768}$.

This system of 1,000 equations with 30 variables does not have (in general) an exact solution, so we approximate the solution by solving a linear least squares problem (Equation 4.3).

However, $\mathbf{A}^T\mathbf{A}$ is not full ranked in this case. As a results, we use Moore-Penrose pseudo inverse as equation 4.6 to solve the linear system.

$$\mathbf{X} = (\mathbf{A}^T \cdot \mathbf{A})^{+} \cdot \mathbf{A}^T \cdot \mathbf{B} \tag{4.6}$$

These 30 word representations can then be used to reconstruct/predict the sentence embedding.

4.3.3 Sentence Embedding Analogy Reasoning

In word embedding, research [102] showed that word2vec embedding manages to capture semantic relationships without learning them during the training process. This relation captured by the learned word embedding allows performing analogy reasoning. For example, the embedding of “queen” can be speculated with the embedding of “king” by simply using algebraic operation, such as

$$\Phi(king) - \Phi(man) + \Phi(women) \approx \Phi(queen)$$

This finding shows that the word embedding not only recognises “king” and “queen” have a similar meaning but also understands “king” is similar to “man” in the same sense that “queen” is similar to “woman”. Another example from the word2vec embedding shows that $\Phi(Germany) - \Phi(Berlin) \approx \Phi(France) - \Phi(Paris)$, which reveals that the word embedding manages to capture more complicated relations, such as “the capital of”. The question is: can sentence embedding describe some relationships within the sentence and perform analogy reasoning as well? We try to reproduce the same experiment under the sentence embedding setup to answer these questions.

We notice that for sentences “The cat sat on the mat.” and “The dog sat on the mat.”, the cosine distance of their BERT embeddings is 0.38. However, by removing the “cat” and “dog” component representations obtained from the linear equation, the cosine distance between these two sentences becomes closer (0.13). Therefore, the BERT sentence embeddings seem to capture the relationship *sat on the mat*.

To investigate if BERT can capture the relationships within the sentence content, we try to find a sentence that is similar to “The cat sat on the mat.” but in a sense of “dog” by calculating $\Phi_{BERT}(\text{“}The\ cat\ sat\ on\ the\ mat.'') - \Phi_{COMPOSED}(\text{“}cat'') + \Phi_{COMPOSED}(\text{“}dog'')$. In a more general case, for any unknown sentence I, we speculate its embedding $\mathbf{b}$ with a similar sentence as equation 4.7. Then, we search in the embedding space for the sentence closest to I. It will successfully find the sentence if there is a linear relation between word attributes and sentence embeddings. Leave one out method is utilised for learning the word representations.

$$\begin{gathered}\hat{\mathbf{b}_{i,j,k'}} = \mathbf{b}_{i,j,k} - \mathbf{x}_{*,*,k} + \mathbf{x}_{*,*,k'},\ k \neq k' \\ \hat{I_{i,j,k'}} = argmax(D_{cosine}(\hat{\mathbf{b}_{i,j,k'}}, \mathbf{B}))\end{gathered} \tag{4.7}$$

During analogy, we separated the test into three groups: subject, verb, and object. For each group, only the corresponding component is changed during the analogy.

By using the leave one out method, we remove the target sentence from the dataset during the learning process of the linear system. As a result, the word representations are learned without previous knowledge of the target sentence $I_{i,j,k}$.

We will test the quality of decomposition by trying to predict the embedding of a new sentence from that of its components, and then comparing it with the BERT embedding of that sentence. This comparison is done in two ways: 1) by computing the cosine similarity between the predicted and the actual embedding, and 2) by using the analogical embedding to retrieve the correct one from a set of 1,000 candidates. To do this, we repeat the process of leaving a sentence out, solving the linear system with the remaining 999 sentences, then using those components to predict its embedding.

Another interesting challenge is if we can predict the sentence embedding $\mathbf{b}_{i,j,k}$ with the word representations solved by the linear system without seeing the actual sentences. To test this, we utilise the leave one out method to solve the linear system and reconstruct the sentence embedding by adding up the word representations we obtained with equation 4.6 so that

$$\begin{gathered}\Phi_{Composed}(I_{i,j,k}) = \Phi_{Composed}(Sbj_i) + \Phi_{Composed}(Verb_j) + \Phi_{Composed}(Obj_k) \\ \hat{\mathbf{b}_{i,j,k}} = \mathbf{x}_{i,*,*} + \mathbf{x}_{*,j,*} + \mathbf{x}_{*,*,k}\end{gathered} \tag{4.8}$$

Given a target sentence $I_{i,j,k}$, the composed embedding $\hat{\mathbf{b}_{i,j,k}}$ are constructed with the sentence attribute vectors $\mathbf{x}_{i,*,*}$, $\mathbf{x}_{*,j,*}$ and $\mathbf{x}_{*,*,k}$ as shown in equation 4.8. We compare BERT sentence embedding $\mathbf{B}$ and composed embedding $\hat{\mathbf{b}_{i,j,k}}$ with cosine similarity.

$$\hat{I_{i,j,k}} = argmax(D_{cosine}(\hat{\mathbf{b}_{i,j,k}}, \mathbf{B})) \tag{4.9}$$

Furthermore, inspired by the word2vec experiments, we try to retrieve the actually BERT sentence embedding from the corpus with the composed embedding. The purpose of this experiment is to confirm whether the composed embedding of an unknown sentence is the closest to its corresponding BERT embedding.

Note that the word representations are learned without the prior knowledge of the target sentence $I_{i,j,k}$. The distance is compared between $\hat{\mathbf{b}_{i,j,k}}$ and every sentence in a given text corpus $\mathbf{B} = \{\mathbf{b}_{i,j,k}\}_{i,j,k=0}^{n-1}$. Moreover, the BERT embedding with the closest distance is considered as the result sentence $\hat{I'_{i,j,k}}$ as equation 4.9. By measuring the times that target sentence embedding comes at first in the retrieving results, we can understand whether the reconstructed embedding can represent the BERT sentence embedding.

4.4 Measuring Compositionality in Sentence Embedding

We measure compositionality in the BERT sentence embeddings by measuring the performance of the linear system based on the experiments described in the previous sections. The performance of learned component vectors **X** is compared with the component vectors that are learned from the corrupted data generated by shuffling the sentence embeddings **B** while solving the linear sentence **AX** = **B**. By shuffling the BERT embedding, it breaks the connection between the sentence (represented by coefficient matrix **A**) and the sentence embedding **B**.

To test if there is additive compositionality within the BERT sentence embedding, we calculate the euclidean norm as equation 4.4 with the unknown **X** learned from the linear system. Moreover, we compare the result with the random permutation result by shuffling sentence embeddings **B**. The smaller the norm, the more accurate the representation **X** can be used to decompose the BERT sentence embeddings.

In the experiments, we also try to reconstruct the sentence embedding by adding the component vectors. The performance of the composed embedding is measured with cosine similarity between BERT sentence embedding **B** and composed embedding $\hat{\mathbf{B}}$ as equation 4.10. The higher the cosine similarity is, the closer the composed sentence embedding is to the actual embedding.

$$D_{cosine}(\mathbf{b}_{i,j,k}, \widehat{\mathbf{b}_{i,j,l}}) = \frac{\mathbf{b}_{i,j,k} \cdot \widehat{\mathbf{b}_{i,j,k}}}{\|\mathbf{b}_{i,j,k}\| \, \|\widehat{\mathbf{b}_{i,j,k}}\|} \tag{4.10}$$

Furthermore, we measure the accuracy of retrieving the target sentence by giving a similar sentence and the word representations as follow. For the embedding analogue, we measure the accuracy by changing subject, verb, and object, respectively.

$$\begin{aligned} \widehat{I_{i,j,k'}} &= argmax(D_{cosine}(\widehat{\mathbf{b}_{i,j,k'}}, \mathbf{B})) \\ Accuracy &= \frac{Number\ of\ correct\ \widehat{I_{i,j,k'}}}{Number\ of\ I} \end{aligned} \tag{4.11}$$

Similar to embedding analogue, the accuracy for retrieving the correct target BERT sentence embedding with the composed embedding is calculated as follows.

$$\begin{aligned} \widehat{I_{i,j,k}} &= argmax(D_{cosine}(\widehat{\mathbf{b}_{i,j,k}}, \mathbf{B})) \\ Accuracy &= \frac{Number\ of\ correct\ \widehat{I_{i,j,k}}}{Number\ of\ I} \end{aligned} \tag{4.12}$$

To further validate the results, we perform a statistical hypothesis test to test if our null hypothesis can be rejected.

4.4.1 Statistical Hypothesis Testing

The linear system encodes the assumption that the embeddings of the items and the components are related. We test this assumption by introducing the null hypothesis that no such relation exists and using a non-parametric test. This is done by generating randomly shuffled data, in such a way that item embeddings and those of the building blocks are randomly matched. This null hypothesis can be discarded if the test statistic computed on the original data has a value that cannot be obtained, with high probability, in the randomised data.

To test that $\mathbf{B}_{i,j,k} \approx \mathbf{x}_{i,*,*} + \mathbf{x}_{*,j,*} + \mathbf{x}_{*,*,k}$, we apply three evaluation matrices and compare the results with 100 random permutations by shuffling embeddings **B**. Using the randomly reshuffled pairs of (component, shuffled item) breaks down the connection between embedding and its building blocks.

Test Statistic: We use three different test statistics: (1) the loss of the linear system; (2) the cosine similarity between **B** and analogical embedding $\hat{\mathbf{B}}$; (3) the accuracy of retrieving **B** with composed $\hat{\mathbf{B}}$.

Null Hypothesis (H_0): The embeddings of an item and its components are independent. There is no benefit in the decomposing item into components.

Alternative Hypothesis (H_a): The item embeddings can be represented by adding up the components.

Significance threshold: $\alpha = 0.01$

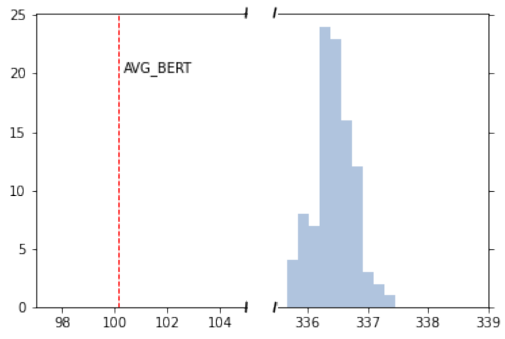
4.5 Results

Table 4.1 and Figure 4.5 illustrates the performance of decomposing BERT sentence embedding. These results show that the p-value for the non-parametric testing is smaller than the significance threshold ($\alpha = 0.01$), and manages to reject H_0. In other words, the BERT sentence embedding can be represented by adding sentence attributes (words) and can be decomposed into three separate components: subject, verb, and object. And those components can then be used to predict the embedding of a new sentence.

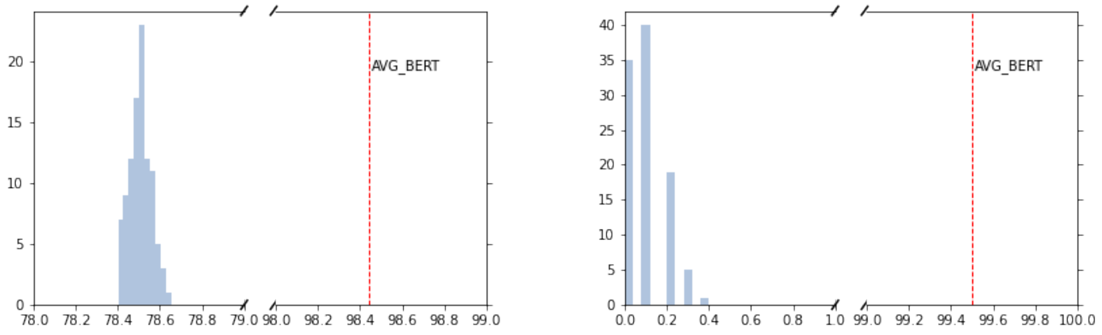
Table 4.1: Statistical Hypothesis Test (non-parametric test) for different test statistics (TS). For TS1, the lower value results in better performance. For TS2 and TS3, the higher percentage indicates better performance. The results show that all the TSs manage to reject H_0.

	Decomposition Performance	Random Permutation Range		P-value
		Min	Max	
TS1	100.14	335.65	337.46	<0.01
TS2	0.98	0.78	0.79	<0.01
TS3	99.50%	0.00%	0.30%	<0.01

The linear system decomposes the sentence embeddings with an average loss of 100.14, which is lower than the lowest norm from random permutations (335.65). This means that the linear system can decompose the BERT embedding into word vectors better than random. And the word vectors can be utilised in the analogy reasoning experiments.



(a) Linear System Loss



(b) Cosine Similarity

(c) Retrieval Accuracy@1

Figure 4.5: The test statistics for sentence embedding decomposition. AVG_BERT is the average performance of analogical embedding $\hat{\mathbf{B}}$ learned from the BERT embedding. The bars are the distribution of the results from random permutations that run for 100 times. $p\_value < 0.01$.

Table 4.2 shows the accuracy of retrieving the target sentence by shifting the embedding with component vectors. The inference embeddings are able to retrieve the target sentences around 98% of the time, which shows that the well-trained BERT sentence embedding is able to capture the word relationship. This allows inference of a sentence embedding with some simple algebraic operations first discovered in the word embedding.

Table 4.2: The accuracy of retrieving the target sentence embedding with a speculating embedding obtained from a similar sentence.

Distinct Component	Sbj	Verb	Obj
Accuracy (%)	98.04	98.08	97.57

Moreover, it is possible to approximately reconstruct the embedding of a sentence just from the Sbj, Verb, Obj components learned from the linear system. This composed embedding $\hat{\mathbf{B}}$ can be compared with the reconstruction that would result from using the same method but based on randomised (attributes, embeddings) pairs with cosine similarity. The cosine similarity between

$\hat{\mathbf{B}}$ and the BERT embedding is 98.44%, which is higher than any randomised trial.

The composed embedding can retrieve the actual embedding with 99.5% accuracy. On the other hand, the composed embedding with randomised attribute/embedding pairs failed to retrieve the randomised embeddings, with the highest accuracy of 0.4%.

4.6 Chapter Summary and Conclusion

BERT and its derived models have been widely used in NLP, including natural language understanding and sentence embedding. In this chapter, we explained the architecture of BERT and how it was trained (masked language model and next sentence prediction). We also described SBERT, a fine-tuned version of Siamese BERT, which can be used for generating sentence embeddings containing semantic meaning.

Embeddings are generated to model the items by learning from the supervised relations. The properties of the distributional context are found to persevere in the learned embeddings. This is known as the "compositionality" of the embedding. This chapter has looked at the additive compositionality in sentence embedding and provided a new insight into decomposing BERT embedding.

The sentence embeddings produced by BERT present some compositionality, that is some of the information contained in them can be explained in terms of known attributes. This creates the possibility to manipulate those representations to remove bias, explain the algorithm's decisions using them, or answer analogical or counterfactual questions.

Therefore, a linear system was introduced to further understand BERT sentence embedding by answering **RQ1a: What properties does BERT sentence embedding contain?**. It can represent the linear relation between the embedding and its attributes in additive compositionality. By solving the linear system with the least square method, we will be able to find the approximate attribute representation that can add up to the embedding. This method can be applied as a general approach to identify compositionality within the embedding.

To examine the properties of sentence embedding, we generated an SVO sentence corpus and embedded it with BERT. By applying a linear system, it has shown that the BERT sentence embedding can be decomposed into word representation so that $\Phi_{BERT}(I_{i,j,k}) \approx \Phi_{COMPOSED}(Sbj_i) + \Phi_{COMPOSED}(Veb_j) + \Phi_{COMPOSED}(Obj_k)$. This allows inference of a sentence embedding with simple linear algebra.

The composed sentence embedding can have 98.4% cosine similarity compared to the BERT embedding, and can find the actual sentence in the vector space 99.5% of the time. The learned word representation can be used to predict the embedding without seeing the sentence. The word representations were also used to present analogy reasoning with BERT embeddings, revealing the relationships BERT managed to capture without prior knowledge. All these results showed that the BERT sentence embedding is compositional.

In the next chapter, we will use BERT to represent a specific type of text: Questions. Moreover, we will process the questions in various tasks with the help of multi-task sentence embedding.

Chapter 5

Training QBERT for Processing Questions

There is increasing attention to the problem of learning generalist agents (as opposite of specialist) in a way that the same representation can be used in a range of tasks, even if it does not excel at any specific task [124]. While a specialist should be expected to excel at its one task, a generalist is expected to be good at many problems. In this study, we address the problem of developing generalist representations of text that can be used to perform a range of different tasks rather than being specialised to a single application.

Training a deep neural network can help process numerous new questions generated every day without human intervention. Some state-of-the-art deep learning models like Transformer [148] are widely used in natural language processing (NLP). They resulted in leading performance for various tasks [43, 118, 168]. Training a language model requires lots of data. Therefore, researchers had to create pre-trained language models using large-scale unsupervised tasks, which are less expensive. Then fine-tune them with labelled task-specific data, which are more expensive. However, labelled data for a specific task are always limited and hard to obtain. Besides, a language model can have a size of millions or billions of parameters. It is usually expensive to train and use a separate network for each task. A generalist model can help address these problems by applying multi-task learning, a learning approach that improves generalisation by adding extra tasks and domain information. The training signals in related tasks are considered as inductive bias for a multi-task learning model [25].

There are two main strategies for multi-task learning. One standard approach is adding extra tasks, also referred to as auxiliary tasks, to improve the performance of the target task. Empirically, adding auxiliary tasks to a pre-trained network is more similar to transfer learning, which improves primary tasks with additional tasks. Another [99] is learning all the tasks jointly without identifying the primary task so that all the tasks can achieve balanced performance, which can be leveraged for training a generalist agent.

As we mentioned in section 3.1.1, there are many different types of tasks included in multi-task natural language understanding. For example, single sentence classification like sentiment analysis, pairwise classification like natural language inference, and regression task like sentence similarity. We noticed that current models focus on general language understanding tasks like GLUE [151] and decaNLP [99]. In this chapter, we focus on training a generalist model for processing a special type of short text: the Questions.

The development of online communities produces a massive amount of text every day. For example, in the question domain, with the rise of commercial voice assistants such as Siri and Alexa and communities such as Quora, numerous questions are asked on a daily basis. This creates a new type of text, Question, in the field of natural language processing (NLP). Processing these questions can provide a new perspective on understanding communities and people's interests.

Question processing is currently recognised as an important stage for question answering systems. Typically, a question processing module includes question parse, classifying the questions based on expected answer type (numeric, location, entity, etc.), and keyword word extraction [61]. By processing questions in such a way, it aims to improve the performance for question answering. However, we think that processing questions has the potential to produce more information than act as an pre-processing stage for question answering systems. In this study, we define *processing questions* with a range of tasks in the question domain.

- detecting topic of questions
- detecting equivalent questions with similar meaning but different wording
- locating potential answers to these questions

In this chapter, we will introduce a multi-task approach for fine-tuning BERT. Using a single model across various tasks is beneficial for training and applying deep neural sequence models. We address the problem of developing generalist representations of text that can be used to perform a range of different tasks rather than being specialised to a single application. We focus on developing a model we called QBERT [162] to generate an embedding for these questions that is useful on a diverse set of problems, such as question topic classification, equivalent question recognition, and question answering. The model targets analysing the semantic and syntactic information in the questions.

QBERT is based on SBERT [125], a Siamese BERT [43] that projects the sentences into high-dimensional vector space. This process is known as embedding. The sentence embeddings with similar semantic meanings are close to each other in the vector space. Research [92, 120] shows that MTL and pre-trained language models are complementary and can be combined to generate better performance on learning text representations. Note that our intention is not to design a new algorithm but to fine-tune SBERT in a multi-task way so that the same representations can be used for processing questions in multiple tasks.

This chapter will explain how QBERT is trained in details. We aim at training QBERT with all the tasks jointly without identifying primary and auxiliary tasks. These tasks share the same domain, which is referred to as inductive bias MTL [160]. After training QBERT, the embeddings generated from the input sequence can be used for both classification and retrieval tasks.

An earlier study we completed [164] on the question-related multi-tasking model shows that the training curriculum is critical. The study trained the tasks one at a time. It shows that one certain curriculum could obtain a balanced performance on all the tasks. More details of this study can be found in Appendix B.

In this chapter, we will introduce a new training curriculum for QBERT in section 5.3, and compare its performance with the one-by-one strategy used in the previous research.

During inference, QBERT produces the representation of the input sequence without any task-specific modules. Instead, it contains a threshold filter to determine the cosine similarity of the embedding pairs, which will be explained in section 5.4. Compared to the standard multi-task structure, reducing task-specific layers simplifies the complexity of the network. The network shares all the weights between tasks, also known as hard parameter sharing.

5.1 Tasks

In this thesis, we define task (T) by data (x), label (y), and loss function (L) as follow.

$$T \doteq \{P(x), P(y \mid x), L\} \tag{5.1}$$

Where $P(x)$ is the distribution of the input data, $P(y \mid x)$ is the distribution of label y given data x, and L is the loss function.

QBERT combined 3 different types of tasks: question topic classification, equivalent question recognition, and question answering. These tasks targeted common natural language understanding problems such as single-sentence classification, pairwise classification, and information retrieval.

Question Topic Classification (QT): Given a question, the model labels the topic of the question. The questions are categorised into 10 different topics according to Yahoo! Answer [169].

Equivalent Question Recognition (QE): In this task, we consider two questions equivalent when they have different wordings but seek the same answer. There are two sub-tasks included in QE, classification and retrieval. In classification, the model aimed at classifying if the question pairs are similar or not, and based on that, retrieving all similar questions from a question corpus with the given question. The model is expected to group similar questions in the vector space.

Question Answering (QA): Given a question, the model searches the answer from a list of candidate sentences. The QA task is trained as classification task and evaluated as retrieval task. We determined this task as an open-domain open-book QA in which the question has no limitation in domains, and the model allows answering the question with the content provided. For retrieval, the model picks one sentence with high confidence from the given context as the answer.

5.2 Generalist BERT for Processing Questions

There is a wide range of tasks included in natural language understanding. For example, single sentence classification like sentiment analysis, pairwise classification like natural language inference, and regression task like sentence similarity. In this section, we propose a multi-task approach to train BERT for processing short questions on a diverse set of NLP tasks, such as multi-class classification (question topic classification), pairwise classification (equivalent question recognition), and regression (similar question mining and question answering). We name our model QBERT.

For question answering, we identify our research as open-domain and open book answer retrieving. Comparing to answering questions using a knowledge base, open-domain question answering is more challenging in using large-scale unstructured knowledge sources and machine comprehension. Previous research [30, 57, 75, 85, 166] leveraged a retriever-reader or retriever-generator that retrieved the relevant passage from the knowledge source and extracted an answer span from the passage. The passage can be a document, paragraph, sentence or fixed-length segment. However, this two stage system is computationally expensive. Inspired by DenSPI [134], QBERT encodes all the sentences in the passage and searches the most relevant sentence with the query.

Our method is based on SBERT [125], which projects input sequence (sentence in this case) in high dimensional space with a Siamese BERT architecture. In such a way, cosine similarity can be calculated between sentence representations produced by the model. Additionally, we can apply our model in binary classification by introducing a similarity threshold Θ, so that

$$Label = 1, \ \ when \ Cosine\,Similarity > \Theta$$

A previous study [164] applied a BERT-based model to process questions with multiple tasks. However, one of the limitations of the previous study is that the model lacked consistency on different question tasks. The model performed topic classification with a single BERT structure, others with Siamese BERT. This is because topic classification is a multiclass classification that requires single input instead of pairwise.

To improve upon this previous study by performing these three tasks with one generalist model, we consider the multiclass classification as a pairwise classification by taking the (Sequence, Class) as the pairwise input. We minimise the distance between sequence and class during training. On the other hand, we retrieve the closest class to a sequence instead of categorising the class for each sequence during inference.

Figure 5.1 illustrates the architecture of QBERT. During training, all the tasks are trained as to minimise the cosine distance using the binary labels and update the shared BERT layer. Task-specific loss functions are introduced for different types of data. While inference, the model only requires shared layers without task-specific layers, which manages to simplify the model.

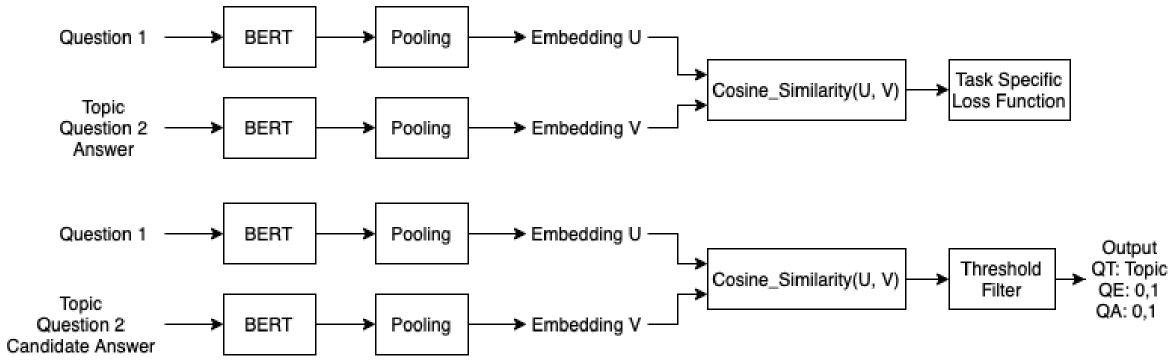


Figure 5.1: QBERT architecture. The architecture is based on SBERT but trained to have a balanced performance on various tasks. Top: Training as binary classification, Bottom: Inference by calculating the cosine similarity between input sequences. All BERTs share the same parameters.

Input layer: $S = (s_1, ..., s_n)$ is an input sequence with n words. The sequence can be either a topic, question, sentence, or paragraph. The model takes a pairwise input (S, S') such as a question pair, question-topic pair, or question-answer pair. The pairwise input is then passed to two BERTs that share the parameters.

BERT layer: The shared embedding layer following the setup of $BERT_{base}$ which takes the sequence input as word tokens and generates an output for each token as well as a [CLS] token at the beginning of the output sequence. $BERT_{base}$ uses the an encoder containing 12 layers and 110M parameters and is pre-trained with two unsupervised tasks: masked language model and next sentence prediction. The output of the BERT layer is in $\mathbb{R}^d$ vector space, and according to BERT, $d = 768$.

Pooling layer: Similar to SBERT, the model leverages a mean pooling strategy that computes the mean of all output tokens (except [CLS]) of the sequence from BERT. After the pooling function, the model generates a pair of embedding U and embedding V as equation 5.2 , where $U \in \mathbb{R}^d$ and $V \in \mathbb{R}^d$.

$$Embedding = \frac{1}{n} \sum_{i=1}^{n} \Phi_{BERT}(s_i) \tag{5.2}$$

We apply two different loss functions for different types of data: online contrastive loss for binary classification tasks that have both positive and negative sample, and multiple negatives ranking loss for information retrieving datasets that does not contain positive nor negative label. Adam optimiser [76] minimises the loss based on the cosine similarity $D_{cosine}(U, V)$.

Pairwise classification specific layer: QBERT introduces the contrastive loss [58] for pairwise classification. It aims to gather positive pairs in the vector space while separating negative pairs. For embedding U, V, the loss is calculated as follows.

$$L_{contrastive} = \frac{1}{2}\{Y(1-D_{cosine})^2 + (1-Y)[max(0, m-(1-D_{cosine}))]^2\} \tag{5.3}$$

Where Y is the binary label. $Y = 1$ if U and V are similar. And the distance $D = 1 - D_{cosine}$ between U, V is minimised. When $Y = 0$, the distance increases between U, V until larger the given margin m. In particular, we apply online contrastive loss that only computes the loss between hard positive and hard negative pairs.

Retrieval specific layer: One of the advantages of applying multiple negative ranking loss is that the training dataset no longer requires either positive or negative labels. For a given positive sequence pair (S_i, S'_i), the function assumes that any (S_i, S'_j) is negative when $i \neq j$. For example, in question answering, for question set $Q = \{q_1, ..., q_m\}$ and answer set $A = \{a_1, ..., a_m\}$, (q_i, a_i) is a positive pair given by the dataset, (q_i, a_j) is a negative pair randomly generated from the dataset. The cross-entropy loss of all the sequences pairs is calculated as follows.

$$L_{multiple\_negative} = -(Ylog(D_{cosine}) + (1-Y)log(1-D_{cosine})) \tag{5.4}$$

During inference, QBERT no longer leverages a task specific layer. Instead, it introduces a threshold filter. QBERT calculates the $D_{cosine}(U, V)$, the cosine similarity between embeddings U and V, and applies different similarity thresholds for each task to determine if two sequences are related in terms of topic, equivalent question, or corresponding answer. The threshold of best performance is selected after training.

5.3 Training Curriculum

In this thesis, we refer training curriculum as the learning order for all the tasks. During training, the data in each dataset get divided into batches $Z = \{z_1, ..., z_n\}$. In each step, one batch z_i is selected randomly, and the model parameters are updated by stochastic gradient descent.

As shown in the previous research [164], the training curriculum was critical for multi-task question processing. In [164], the tasks were trained once at a time, from QE to QA to QT (QT was trained with different network architecture). However, the tasks learned in the earlier stage had a worse performance compared to the tasks learned in the later stage. To improve this, we train QBERT in a fixed-order round robin (RR) curriculum and compare the results with one by one (OBO) curriculum.

In the OBO approach, we train QBERT following QT, QE, and QA orders. Every dataset is divided into multiple batches, each with specific batch size, and is trained one at a time. The parameters are updated during the training and shared amongst all the datasets.

On the contrary, QBERT trains all the tasks simultaneously in the RR curriculum. The data in each task-specific layer are built as mini-batches and divided into two task-specific loss functions. During each step, the model is trained and updated by batches with online contrastive loss and multiple negative ranking loss. QBERT-RR alternates between tasks during training which prevents the model from forgetting about the tasks learned at the beginning of the training. The RR training procedure is explained in algorithm 1.

Algorithm 1 Training QBERT-RR

```
Randomly initialise model parameters.
Pre-train shared BERT layers following BERT and S-BERT
for d in Datasets do
    Pack d into mini-batches $Z_{binary}$ or $Z_{retrieval}$ for binary classification loss and retrieval loss
end for
for e in 1,2, ..., $Epoch_{max}$ do
    Shuffle $Z_{binary}$ and $Z_{retrieval}$
    for s in 1,2, ..., $Step_{max}$ do
        if $z_{binary}$ then
            Compute loss L with Eq. 5.3 (binary classification)
        else if $z_{retrieval}$ then
            Compute loss L with Eq. 5.4 (retrieval)
        end if
        Compute gradient
        Update model
    end for
end for
```

5.3.1 Data Prefix Identifier

In QBERT, the input sequences from all different datasets are embedded using two identical BERTs. The encoder will try to learn a common representation that can be used for all the tasks. This may lead to a symmetric problem, which restricts the network's capacity to learn task-specific features.

Therefore, we try to break the symmetry by adding prefix identifiers for different tasks. As illustrated in Figure 5.2, three types of prefixes are added to all the datasets. QBERT-DI is trained following the setup of the RR curriculum.

5.3.2 Fine-tuning QBERT

Inspired by MT-DNN [92], we are also interested in how fine-tuning can improve the performance of QBERT on a specific task. Thus, we fine-tune QBERT based on transfer learning, which identifies primary and auxiliary tasks during training. In this case, we do not use one model for all tasks. Instead, we try to explore the performance of QBERT when it acts as a "specialist".

During training, we first train all the auxiliary tasks simultaneously following the RR curriculum in Algorithm 1. Then we fine-tune the model with the primary task. For example,

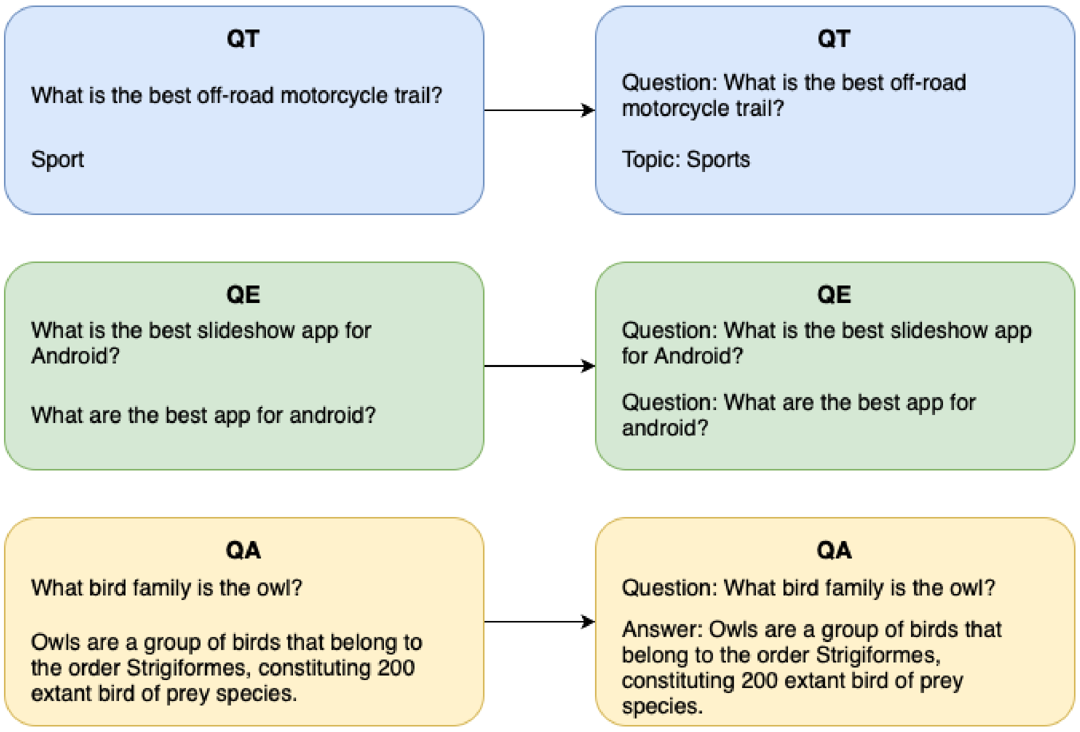

QT
What is the best off-road motorcycle trail?
Sport
QT
Question: What is the best off-road motorcycle trail?
Topic: Sports
QE
What is the best slideshow app for Android?
What are the best app for android?
QE
Question: What is the best slideshow app for Android?
Question: What are the best app for android?
QA
What bird family is the owl?
Owls are a group of birds that belong to the order Strigiformes, constituting 200 extant bird of prey species.
QA
Question: What bird family is the owl?
Answer: Owls are a group of birds that belong to the order Strigiformes, constituting 200 extant bird of prey species.

Figure 5.2: Adding data prefix identifier for input sequence pairs from different tasks.

to fine-tune QBERT for specialising in topic classification, we pre-train QBERT on question recognition and question answering followed by a topic classification dataset.

5.4 Threshold Filter

In QE and QA, apart from classifying if the sequences are related, it is also crucial for the system to search all the related sequences (equivalent question or answer) for the given queries. The problem is how to quantify "related" with embeddings. A cosine similarity threshold is introduced in this model. Using a threshold simplifies the network structure of QBERT during inference by removing the task-specific layer. This threshold we introduced can be considered as a margin that filters related sequence pairs from others. For instance, for a pair of sequences (S, S'), if the cosine similarity between them is larger than the threshold, S is related to S'. All the unrelated sequences to the query in the corpus can be removed by applying a threshold filter.

With this threshold, the network can not only search for the information that is closest to the query but can also identify if the information is related (close enough) to the query. For example, a question might be unique in the corpus so that its closest sequence is not equivalent to it if the cosine similarity between the question and the sequence is smaller than the given threshold. Or a question might not have a high-confidence answer from the candidate corpus, so the closest candidate with a cosine similarity smaller than the threshold will not be considered the right answer.

To decide the threshold, first, all the sequences in the training set are embedded with the fine-tuned model. The sequence pairs are classified as positive if they have more similarity than the threshold. The similarity threshold with the best accuracy in the training set is calculated in order to quantify any question pairs during testing. With a threshold, the model is capable of searching and grouping all the related sequences in a given candidate corpus.

5.5 Experiments

Training QBERT includes pre-training and multi-task training. We utilise the pre-training methods of BERT and SBERT. Then we perform MTL on question related datasets. We train QBERT on five different datasets and evaluate it on four of them.

5.5.1 Datasets and Metrics

Quora Question Pair (QQP) [38] first released on Quora in 2017. It is a dataset that contains 404k question pairs collected and annotated by Quora. QQP labels if the questions are duplicated or not. There are 537k unique questions in the dataset. Training on QQP, we aim at improving the performance of QE tasks for QBERT. We then evaluate QQP in both binary classification and similar questions retrieval.

WikiQA [167] is a question answering dataset which has the questions from query logs on Bing and one-sentence answers from Wikipedia's summaries. The questions in WikiQA are factual questions that start with WH words like who, what, and when etc. The candidate answers are extracted sentences from the first paragraph of Wikipedia articles (also known as Wikipedia Summary). For each question, the sentences from a Wikipedia Summary are given as candidate answers. The dataset contains human labels of correct answers or not for each sentence in the summary. The dataset includes 3,047 questions and 26k candidate sentences; 1,239 of the questions have a correct answer from Wikipedia. We train WikiQA as a binary classification task and evaluate it as an answer selection task.

Yahoo! Answer [169] data were originally collected by Yahoo! Research Alliance Webscope program. Zhang et al., built up a corpus which contains 1.46M samples within 10 most popular topics on *Yahoo! Answer*. The sample includes the topic, question title, question content, and the best answer provided by the user. We apply *Yahoo! Answer* corpus in both QT and QA tasks. For QT, QBERT takes the question title and topic as the input sequence pairs. Question titles and best answers are leveraged for training QA tasks. To distinguish the data applied for different tasks, we name it YT for the data used in QT and YQA for data in QA.

Stanford Question Answering Dataset (SQuAD) [121] is a corpus that contains questions, answers and contexts for reading comprehension tasks. The contexts are paragraphs extracted from Wikipedia. We use SQuAD 1.1 as all the questions have corresponding answer phrases in the given context for training. There are 98,169 question-answer pairs in the dataset. To train

QBERT, we take the question and the specific sentence in the context that contains the answer phase as input.

Table 5.1 summarises all the corpora and their corresponding evaluation metrics.

Table 5.1: Summary of the training datasets. Note that the test set of SQuAD is confidential from the researcher. The number of test data states here is the validation set that is publicly accessed. The metrics for SQuAD in this paper is "exact match in sentence" which will defined in section 5.6.

Dataset	**#Train**	**#Test**	**Label**	**Metrics**
YT	1,400,000	60,000	10	Accuracy
QQP	283,001	121,286	2	Accuracy/F1
WikiQA	23,080	6,116	2	Accuracy/F1
YQA	14,000,000	600,000	1	-
SQuAD	87,355	10,539	1	EM*

5.5.2 Implementation Details

For each input sequence, the length is limited to 35 tokens because we use two BERTs to read the sequence pair instead of concatenating two sequences into one as the input. Besides, most questions in the datasets have less than 35 tokens. The sequence is truncated at the end if it is longer than the limitation.

Usually, the machine takes a single sentence as an input and performs multi-label classification for question topic classification. To adapt the QT task into the QBERT Siamese network architecture, we convert the topic classification into a topic retrieval task. QBERT embeds the questions as well as the topics, and labels all the question topic pairs as related. During training, the model minimises the multiple negative ranking loss between the topic and questions. During inference, the model calculates the cosine similarity between the question and all candidate topics. The candidate with the largest similarity is considered as the topic of the question.

We train QBERT with the online contrastive loss for QE. We define the similarity threshold for QE based on the best accuracy on the training set. Then we evaluate the model on both QE classification and retrieval tasks. The QE retrieval candidate corpus is constructed by sampled queries in the QQP test set.

For QA, we train WikiQA with the online contrastive loss and YQA and SQuAD with multiple negative ranking loss. This is because YQA and SQuAD only contain question answering pairs and do not come with negative samples. However, WikiQA has both positive and negative samples. In the meanwhile, there are questions with no answers in the dataset. Thus, a threshold is needed to identify whether the closest candidate to the question is the high-confidence answer. The threshold is defined as the one that creates the best precision in the WikiQA training set. Using the threshold that creates the best precision ensures the retrieving candidates have a low false positive rate. In other words, the answers selected by the model are more likely to be the correct answer.

The implementation of QBERT is based on PyTorch and SBERT. The training parameters are shown in Table 5.2. The margin for positive samples and negative samples is 0.5. We train the model for 5 epochs with a batch size of 32 and a learning rate of $2e-5$. 10% of the training data is used for warm-up.

Table 5.2: Training parameters for QBERT.

Epoch	5
Batch Size	32
Sequence Length	35
Learning Rate	2e-5
Warm-up Steps	10% of the training data
Margin for Online Contrastive Loss	0.5

In our experiments, we train the generalist QBERT with two curricula: QBERT-OBO and QBERT-RR. And we compare the results with the specialist models SBERT-QT, SBERT-QE, and SBERT-QA. Each of the specialist models fine-tunes SBERT with only its task-specific datasets. Furthermore, we fine-tune QBERT into a group of specialist models called QBERT-FT as described in section 5.3.2.

We train QBERT with one GeForce GTX TITAN X GPU. To train QBERT-OBO, it takes 45.5hr, and 93hr for QBERT-RR. Even though training the model is time-consuming, once trained, the model is much faster during inference. It takes 1.5ms, 5.44ms, 19.62ms, and 49.76ms per question in YQT, QQP, WikiQA, and SQuAD, respectively.

5.6 Performance of QBERT

We evaluate QBERT with YT for QT classification, QQP for QE classification and QE retrieval, and WikiQA and SQuAD for QA retrieval.

For QE, we evaluate classification and retrieval task accuracy with the QQP dataset. If the question pair has a similarity larger than the threshold, it is categorised as equivalent in classification. To perform similar question mining, we create a question corpus based on QQP. First, all the relevant questions for the given query are included in the dataset, ensuring that there is always a relevant question in the corpus. Second, we fill the rest of the corpus with irrelevant questions. There are 104,033 samples in total. While mining the similar questions from the corpus, the candidate with the highest similarity larger than the threshold is defined as the duplicate question.

We assess the QA performance only on WikiQA and SQuAD, because the answers in the YQA are paragraphs provided by Yahoo! users, and it is hard to construct a candidate corpus used for single-sentence answer retrieval. We count the number of questions that can correctly identify the answer (or *None* for the questions without an answer) from the corpus while evaluating QA tasks.

In WikiQA, the question is not guaranteed to have an answer. Therefore, for each question, the model takes the sentence with the highest cosine similarity score in the candidate set and compares it with the threshold. If the similarity is above the threshold and the sentence is labelled as a correct answer, then the prediction is correct. For SQuAD, each query has a corresponding answer in the given context. Thus, we take the sentence with the highest cosine similarity score as the candidate. Note that for SQuAD, the ground truth answer is a short answer phrase extracted from the given context. Since QBERT retrieves one sentence as the answer, we evaluate the exact match phrase in the sentence, which depends on whether the answer phrase is in the selected sentence.

To understand the performance of MTL, we use SBERT, which is fine-tuned with natural language inference dataset [22, 159] and semantic textual similarity dataset [26] as our baseline. We also compare QBERT with the single-tasks model. The result is shown in Table 5.3.

Table 5.3: The performance of QBERT-RR and QBERT-OBO compares with the performance of single-task SBERT trained on QT, QA and QE. SBERT without training on any question related dataset is used as the baseline.

Curr.	**YQT**	**QQP-C**	**QQP-R**	**WikiQA**	**SQuAD**
	Acc.	Acc.	Acc./F1	Acc./F1	Acc.
Baseline	35.27±0.58	74.80±0.32	54.53±0.89/53.01±0.82	77.46±2.82/58.24±12.48	67.04±0.93
QT	72.44±0.39				
QE		89.79±0.23	56.98±0.65/55.36±0.60		
QA				79.05±5.89/72.50±11.08	78.59±0.82
OBO	59.84±0.32	78.85±0.44	57.46±0.78/55.87±0.70	80.16±6.03/69.29±9.00	76.09±0.66
RR	73.77±0.58	90.13±0.19	58.22±0.78/56.53±0.75	81.90±5.60/73.73±8.12	71.42±1.45

SBERT was only trained on natural language inference dataset and semantic textual similarity dataset containing sentence pairs with labels. It therefore, manages to detect similar question pairs albeit with poor performance. However, SBERT was not trained to group sentences with the same topic, and it is unable to identify the question topic. Since SBERT achieves a similar accuracy to other models on WikiQA dataset, it has a worse F1 score compared to others.

In Table 5.3, model SBERT-QT, SBERT-QE, SBERT-QA represent single-task training. It leverages the same architecture as QBERT. However, for each task, it has a separate model. While fine-tuning the single-task model, we update the BERT layer to minimise the task-specific loss for each dataset. The results show that QBERT-RR achieves similar performance on most question datasets compared to the single-task model, except for retrieving answers from SQuAD, with the generalist representation.

The previous research [164] proved that the training curriculum is important for training a multi-task network. Thus, we investigate two different training strategies. The QBERT-OBO shows better performance on WikiQA; on the other hand, it has worse performance on QT and

QE compared to the single-task models. When training QBERT-OBO, we train one dataset after another. This causes the model to “forget” what it learned during the early stages.

In contrast, while training with the RR curriculum, the model achieves a balanced performance on each task. Although QBERT-RR does not excel in any task compared to the single task model, it is able to generate a representation that can be used to perform a range of question tasks. Figure 5.3 shows the performance of QBERT-RR on QE and QA classification tasks.

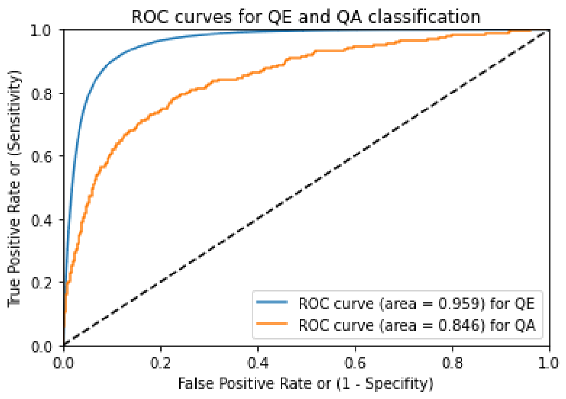


Figure 5.3: ROC curve for QE and QA classification from model QBERT-RR. The black dashed line represents the performance of a random classifier.

While assessing WikiQA, some questions do not have an answer in the given contents. Therefore, we leverage a similarity threshold to identify whether the candidate answer with the highest similarity is the correct answer. QBERT-RR determines 89.72% of the questions without correct answers in the candidates by using the chosen threshold.

Table 5.4 compares QBERT-RR with E5 [152], a state-of-the-art general-purpose embedding model that achieved strong performance in classification, clustering, and retrieval. Compared to QBERT, E5 trained the shared encoder with a prefix identifier for the data. The results show that E5 obtain a better performance on QA. On the other hand, QBERT-RR outperforms E5 on QT and QE tasks.

Table 5.4: The performance of QBERT-RR, QBERT-DI, and QBERT-FT compared to E5.

Curr.	**YQT**	**QQP-C**	**QQP-R**	**WikiQA**	**SQuAD**
	Acc.	Acc.	Acc./F1	Acc./F1	Acc.
E5	55.16±0.44	77.72±0.21	60.06±0.95/58.40±0.89	82.54±3.55/72.88±6.86	77.66±1.06
RR	73.77±0.58	90.13±0.19	58.22±0.78/56.53±0.75	81.90±5.60/73.73±8.12	71.42±1.45
DI	73.48±0.37	90.05±0.27	58.21±0.93/56.56±0.87	84.92±3.57/76.67±8.28	71.09±1.65
FT	72.34±0.73	90.08±0.26	57.49±0.82/55.87±0.73	80.16±4.15/73.72±8.35	76.04±1.09

Similar to E5, we introduce data prefix identifiers while training QBERT. QBERT-DI has a similar performance to QBERT-RR on most of the datasets. However, it creates a better performance on WikiQA.

Table 5.4 also shows the results of the specialist QBERT. Apart from SQuAD, QBERT-FT does not significantly improve from the generalist QBERT-RR. Moreover, QBERT-FT does not outperform the task-specific model, which means that adding more in-domain question data makes a negligible difference in QBERT performance. As a result, we will use QBERT-RR as an example to analyse a new question corpus in chapter 6.

Furthermore, we evaluate the accuracy@K among different retrieval corpus sizes for QE using the QQP test set. Accuracy@K is a top-K accuracy classification score. In QE, it counts the number of times where the relevant question is contained in the top K candidates. According to the results illustrated in Figure 5.4, it is more challenging to retrieve among the larger corpus. When all the queries in the dataset are included in the retrieval corpus, the accuracy@1, accuracy@3, accuracy@5 are 58.150%, 82.573%, and 89.233%, respectively. More than 80% of the related questions are located in the top 3 candidates. However, only 58.150% of them are the closest to the given query, which can be improved in the future.

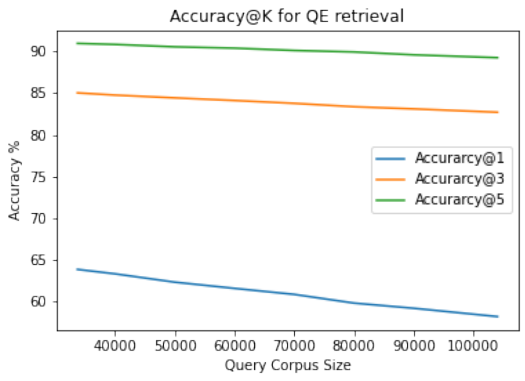


Figure 5.4: Accuracy@K of different corpus sizes in QE retrieval task

Lastly, we notice one limitation while evaluating QA retrieval with the SQuAD dataset. When creating the candidate corpus, we leverage a sentence tokeniser to split the paragraph into sentences. However, the sentence tokeniser splits the sentence based on the punctuation. For example, “Washington, D.C.” is considered two sentences: “Washington, D.” and “C.”. During evaluation, we compare the selected sentence with the answer phrase. In this case, retrieving a sentence may yield an incomplete answer to a question.

5.7 Chapter Summary and Conclusion

The ever-increasing amount of question/answer data generated through smart voice assistants and online query communities are primarily used for training automated question answering systems. We noticed more potential in harnessing this vast amount of data to extract other valuable information. Therefore, we raised research question **RQ2a: What tasks can be included in processing questions?** To answer this question, this chapter defined “Processing Question” as a collection of three tasks: question topic classification, retrieving equivalent questions, and best answer selection.

Typically, a different deep learning network is trained for each of these tasks. Building these multiple networks, achieving high predicting performance, and using them is challenging and expensive since a vast amount of training data and parameters is required, and the resulting networks are typically large. With this concern, we raised a multi-task learning approach to fine-tune BERT, which intends to use one shared representation for multiple tasks. Hence, we introduced MTL while training the question processing network.

In this chapter, we answered the research question **RQ3: How does multi-task learning affect the performance of processing questions?** by proposing and training a generalist model QBERT to process questions in a variety of tasks. The idea is that sometimes a generalist model can be useful even when it does not beat specialist models at their own speciality.

We observed that one version of the generalist model QBERT-RR turned out to perform similarly to the specialists in many cases except for QA retrieval on the SQuAD dataset. The specialist methods in this study for comparison were SBERT models fine-tuned respectively on QT (YQT), QE (QQP) and QA (WikiQA, SQuAD). Instead, another generalist method, QBERT-OBO performed worse than the specialists on QT and QE (classification). The reasons for this performance need to be further investigated, but it might happen because the OBO curriculum resulted in forgetting the tasks that were learned in the earlier training stage. Moreover, the results of fine-tuning QBERT into a specialist model showed that adding more data on processing specific question tasks was negligible.

In the next chapter, we will keep looking at the information that can be learned from the question. A pipeline will be presented for processing new questions. Furthermore, as an example, we will apply QBERT-RR to analyse a new question corpus collected by “We The Curious”, a science centre in Bristol, UK. By analysing the question corpus, we intend to understand people’s interests.

CHAPTER 6

Analysing a New Question Corpus: WTC Corpus

In the previous chapter, we defined three tasks for processing questions, and we introduced QBERT-RR, which is able to perform these three tasks with one single model and achieves a balanced performance across multiple open-source question datasets. In this chapter, we are going to apply this model to a new question corpus and analyse its contents based on the model outcomes.

In 2017, the "We The Curious" science centre in Bristol started "Project What If" to encourage Bristolians to write down their questions and explore their curiosity. No rule was given during the question collections. People can ask all types and topics of questions, following their curiosity. The project managed to collect more than 10,000 questions on various topics, and more are still being collected on a daily basis. Therefore, by analysing these questions, the science centre is able to have a perspective on visitors' curiosity. And this can help the science centre create exhibitions or content that are more relevant to the visitors' interests and lives in the future.

Questions have a critical role in learning and teaching. Many studies [46, 141] showed that asking questions improved educational outcomes and promoted a further understanding of the learning material. People ask questions to obtain information and express interest in ideas. Moreover, encouraging learners to generate questions can promote critical thinking in the learning content. Research [39] showed that questioning is one of the essential thinking processing skills for critical thinking, creative thinking, and problem-solving. Learners' questions play a critical role in both learning and teaching [32]. Students' questions can help construct knowledge, self-evaluate, and motivate their interest in a topic during learning. On the other hand, teachers can diagnose students' understanding and evaluate their thinking through their questions.

Different kinds of questions can stimulate different extents of learning [132]. For example, knowledge-based questions generated from interest or to better understand and extend the knowledge have a higher order than text-based questions that are asked in response to given content. Scardamalia and Bereiter's research also showed that students tend to ask questions about basic information for less familiar topics but more wondering questions for familiar topics. Thus, categorising the questions can be beneficial to understanding the questioners and tailoring the learning content.

Meanwhile, with the development of online communities and smart voice assistants, many people are asking questions and searching for answers on a daily basis. A huge number of questions are produced every day. The idea of the online question community like Quora and Stack Overflow also encourages users to ask questions and connect with people who have the same question or have unique insights and quality answers.

The challenge is, with this large amount of data collected by the "We The Curious" science centre or any other educational platform, it is time-consuming to process and analyse all the questions manually. Therefore, applying artificial intelligence techniques and models allows the questions to be learned and processed efficiently without human intervention. Moreover, these numerous queries produced also lead to a challenge in applying artificial intelligence: Can we use the machine to understand and analyse these large scales of queries collected from different circumstances? And how can the model benefit education?

Both the field of education and the field of AI have a great interest in questions. To automatically analyse the corpus of questions, we apply one of the question processing models we demonstrated in the previous chapter. It can transform short sentences into high-dimensional vectors, and we use it to process the questions in terms of recognising the topic, equivalence and potential answers to a given question.

Figure 6.1 illustrates the pipeline for analysing a question corpus. The processes include data collection, manual moderation, automated processing, question embeddings, and question processing. Manual moderation is optional and depends on the different needs and data policy. In our experiment, since the data is for educational purposes, we manually remove personal data and inappropriate content.

We will demonstrate these processes with one version of the trained QBERT-RR on the new question corpus collected by "We The Curious" science centre. QBERT-RR is utilised to embed the questions and the external knowledge source. More details for each step will be explained in this chapter.

As shown in Figure 6.1, we apply three tasks while processing the questions: question taxonomy, equivalent question recognition and question answering defined in section 5.1. On top of the topic classification task that QBERT does, we also include a type classification when analysing the questions. Besides, we introduce an external knowledge source as a candidate answer set when searching for answers to a given question. In such a way, we try to reveal more information contained in the question corpus. For example, from the question taxonomy, we can

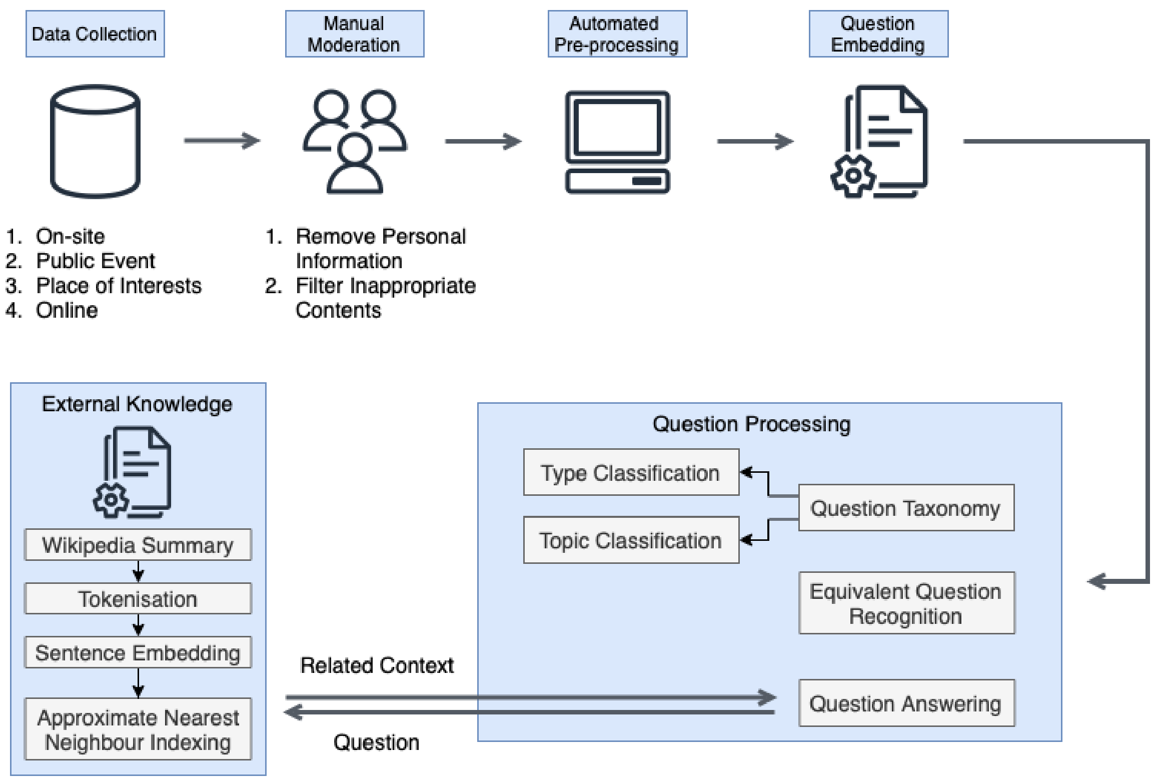


Figure 6.1: The pipeline for analysing question corpus.

understand the people who asked the questions from their interests and comprehension level of the content; equivalent question recognition can group the common questions and help locate identical and popular questions; and the question answering task can identify the questions that can be answered by an external source, which potentially reduces the workload on answering queries. As a result, the educational content providers can adjust their focus on the common questions or questions that a machine cannot answer based on the analysis.

We will present this chapter following the pipeline of the question analysing system. Firstly, we introduce a new question corpus collected by "We The Curious" in section 6.1, including data collection and manual moderation, followed by automated pre-processing in section 6.2. Section 6.3 demonstrates the question processing with QBERT-RR and its results. At the end, content analysis and discussion of the corpus are given in section 6.4.

6.1 WTC Corpus Overview

We will call our dataset of open-domain questions "the WTC corpus", this section describes its origin and main features. The dataset is originated from the "Project What If" run in Bristol (UK) by "We The Curious" (henceforth WTC), an educational charity and science centre.

6.1.1 Data Collection

Between January 2017 to October 2019, “We The Curious” collected over 10,000 open-domain questions from a diversity of sources: “We The Curious” venue in Bristol, offsite, and online. Offsite question gathering ensured questions were received from Bristol postcodes BS1 - BS16, out of 37 postcode districts in Bristol. In the meanwhile, all Bristol postcodes were covered through general submissions. The data came from people of all ages and backgrounds. All these data collected started a digital database of questions held by “We The Curious”. There are four main methods for question collection.

Project What If Cube: this was a space set up on the venue floor whereby visitors could enter a large wire cube, write down their questions on a piece of paper and then attach it to the cube. Questions were routinely taken down and stored, after which they were inputted by hand into the question database spreadsheet.

Curious Cube outings: this was a portable cube that portrayed asked questions through a mirrored surface in LED lights. The cube could be connected to an iPad through which questions could be entered, stored, and displayed on the cube. The team collected questions at various events around Bristol.

Question gathering: this was a series of events whereby “We The Curious” staff visited places of interest, such as schools and community centres, where they facilitated question input of participants using question cards. Questions were then inputted by hand into the question database spreadsheet.

Online input: An entry point was made available on the “We The Curious” website, whereby any user could enter a question digitally, and the question would be stored. When a question was entered through the website, no personal information was taken.

6.1.2 Manual Moderation

Based on this initiative, “We The Curious” created a digital database of questions, which is in “We The Curious”’s possession. All the questions collected were represented verbatim. “We The Curious” is responsible and accountable for protecting the personal data of individuals submitting this information alongside their questions. All personal data is held by “We The Curious” in compliance with GDPR protocol, and personal data is not shared with other parties, including the analysis team of this project. For the purpose of the present study, a smaller dataset was generated by removing all the personal data associated with the questions, and only this was shared with us for analysis.

The raw corpus also included identical questions, various types and topics, and other non-question sentences. Questions were first moderated manually by WTC staff. The questions in the database were also screened for any possible identifying information or potentially offensive or inappropriate language or content. These were removed from the database manually. After moderation, the resulting dataset contained 10,073 questions.

This anonymised and moderated textual dataset is what we will call the WTC corpus in this thesis. And all the analyses are produced with the question-only corpus.

6.2 Automated Data Pre-processing

In a traditional NLP pipeline, text pre-processing is a process that cleans up the data before feeding it to a machine learning model. It usually includes removing punctuation and stop words, stemming, lemmatisation, and lower casing the corpus. However, the text curation process is simplified by using a pre-trained language model like BERT.

BERT leverages a multi-head self attention mechanism that uses all the text's information, including stop words and punctuation (as described in section 4.1). Moreover, while tokenising the input text, BERT uses WordPiece [161], a subword segmentation algorithm that breaks down rare used words into meaningful subwords. Thus, stemming is no longer necessary while using BERT-based algorithms.

Even though we do not process the text following all the processes in the traditional NLP pipeline, some simple pre-processing is performed based on the nature of the questions, such as removing the identical questions and sentences that are shorter than three words. After these steps, the filtered WTC dataset contained 8,600 questions using 5,732 unique words. The length of questions is between 3 and 55 words, with an average of 7.21 words. 87.96% of the questions are within 10 words. The distribution of the question length is illustrated in Figure 6.2.

Table 6.1: WTC Corpus Overview. Size: # of questions. Vocabulary, Length: # of words.

	Size	**Vocabulary**	**Question Length**		**Average Length**	**Median Length**
			Min.	**Max.**		
WTC Corpus	8,600	5,732	3	55	7.21	6

The word cloud in Figure 6.3 illustrates the high-frequency words in the WTC corpus, excluding the stop words. The size of the word in the figure is proportional to its frequency in the corpus. The figure shows that the questions cover the universe and space, the human body, energy and climate change, animals and plants, chemistry and materials, the future, and other topics outside the typical science categories listed before.

6.3 Processing the WTC corpus with QBERT

In the WTC corpus, we have questions with various contents from different people. The questions cover multiple topics and overlap in content. Through the analysis, we intend to understand the question content, like the types and topics, so that people can further understand the questioner's interest and understanding level.

Some questions in the dataset are equivalent but in different wordings. For example, the question "How does the earth spin around?" and the question "What way does the Earth spin?"

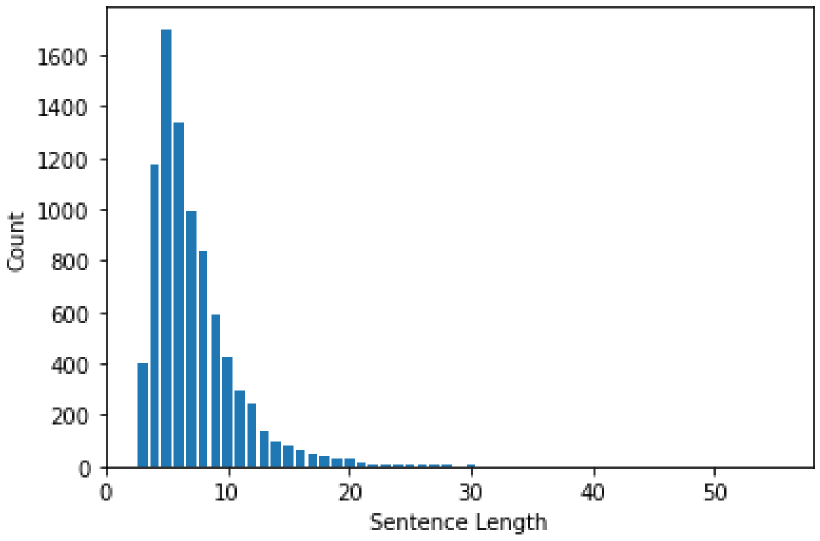


Figure 6.2: The distribution of question length in the WTC corpus.

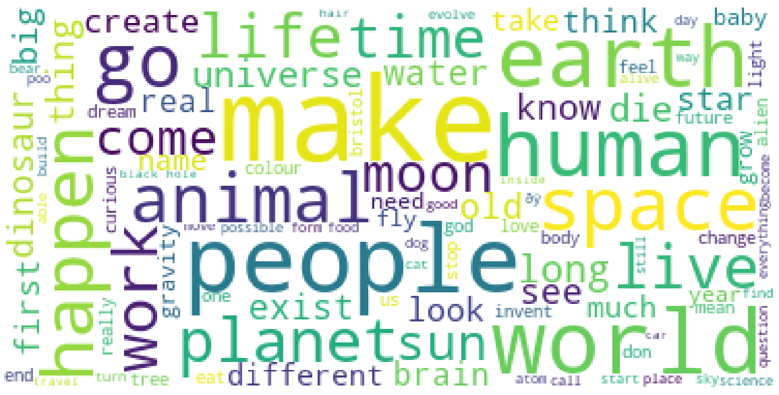


Figure 6.3: The word cloud is generated from the curated and filtered WTC corpus (with 8,600 questions). The words were lemmatised before generating the graph.

are in different wordings but can be answered with the same content. During pre-processing, we cannot identify these equivalent questions. However, with QBERT, we can filter these duplicate questions by projecting them into a vector space. In such a way, we can reduce the workload for further analysis. QBERT can also link any new question collected in the future with the existing data by comparing them with cosine similarity. Furthermore, the similar questions can reveal the common doubt of the visitor, which can help comprehend the questioner.

QBERT also identifies the questions in the WTC corpus that can be answered with high confidence by an external knowledge source, such as Wikipedia Summary. Under an educational scenario, the teacher can pay more attention to the knowledge-based questions that the machine cannot answer.

After automatically pre-processing the corpus, we apply QBERT-RR (explained in section 5.2) to embed the questions and knowledge source, so that

$$\mathbf{B} = \Phi_{QBERT\_RR}(I),\ \mathbf{B} \in \mathbb{R}^{768\times no_of_Q}$$

Where I is a set of questions or knowledge souce, and **B** is the set of 768-dimensional embedding vectors. After embedding, the questions are processed with three tasks: question taxonomy, equivalent question recognition, and question answering.

6.3.1 Question Taxonomy

In the WTC corpus, there are confirmation questions such as "Are all babies born with blue eyes?"; factual questions such as "Who built the internet/electricity?"; and counterfactual questions such as "How long will the earth and humans last if we carry on damaging it and nothing changes?". Different types of questions can indicate the questioner's depth of thinking [31]. Identifying the type and topic of the questions provides insight into the visitor's understanding and thinking [156].

When categorising the questions by type, we focus on classifying them based on the grammatical forms: interrogative words. The type of a question is categorised into the following categories: WHAT, WHO, HOW, WHEN, WHERE, WHY, WHICH, IF, and OTHER, which will be further discussed below. Keyword matching is leveraged for type classification. On the other hand, for topic classification, we focus on the semantic topic of the questions, which includes: Business & Finance, Computers and Internet, Education & Reference, Family & Relationships, Health, Politics & Government, Science & Mathematics, Society & Culture, Sports. The topic of a question is classified by QBERT-RR. We had nine types and ten topics, and therefore 90 question "Themes" to which we can allocate the over 8,000 distinct questions that have survived the various stages of filtering.

Identifying the interrogative words in a question help identify grammar-based question categories used in linguistic research, such as confirmation, factoid, hypothetical, and casual.

Confirmation questions are also known as yes-no questions. Yes-no question is a polar question that contrasts with a non-polar question like the wh-question. And it expects the answer to be one of two choices. Since the answer to the confirmation question can be a simple "yes" or "no", the question is considered less complicated compared to others.

A factoid question is a question related to facts. Most of them contain question words like "what", "who", "which", "when" and "how". Usually, factoid questions can be answered with a concise sentence from a knowledge source. Both confirmation and factoid questions are mostly asked to help understand the conceptual knowledge of given content.

Causal questions usually begin with "why" or "how". This category of questions usually requires further explanation in the answer and can be more challenging to answer with one sentence.

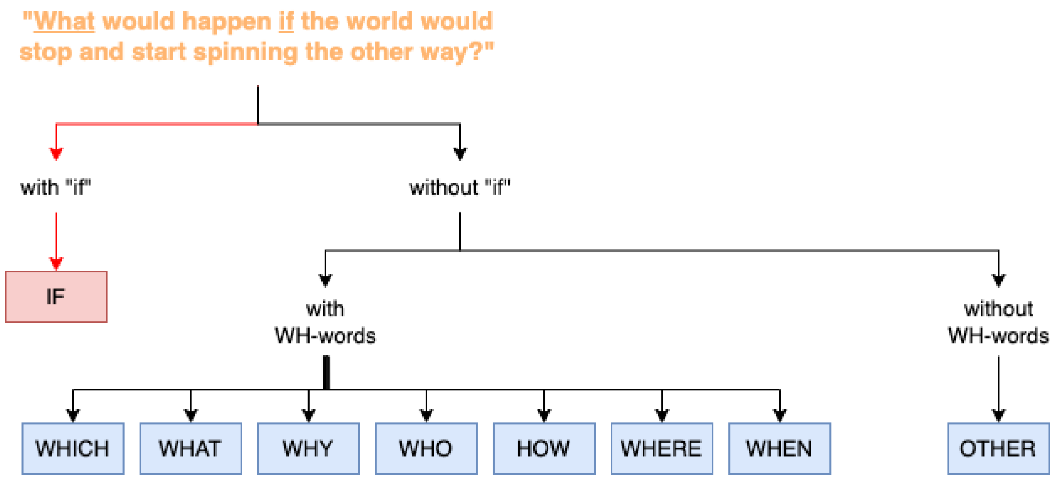


Figure 6.4: Example of type classification. The type of the question depends on the interrogative words in the text.

A hypothetical or counterfactual question is defined as a question of the type: "what would happen if X was true". The understanding is that X is not a true fact, but the people who ask the question are considering the possible consequences of X being true. Counterfactual question takes its name from being "counter to the facts", is often used in defining the notion of causality (e.g. in [111]), and indicates a mental process directed at understanding the mechanism behind observations. Causal and hypothetical questions are asked with further thinking and interest, and effort to make sense of the world.

Although there are overlaps between interrogative words and grammar-based question categories, classifying the questions into various types can still contribute to diagnosing people's understanding.

Given a question, we assign it to the type of the first keyword from our list that is found in it, with one notable exception for type IF. For example, the question "Why do we get butterflies when we like someone?" is categorised as WHY regardless of any other keywords that are in the query after "why". However, questions that contain the keyword "if", such as "what if" and "How ... if ..." are classified as IF questions regardless of the position of "if". The category OTHER includes yes/no questions or sequences that do not fit into other categories. Figure 6.4 shows an example of type classification.

Define a further class of IF question intends to find a simple way to approximate the counterfactual questions of the type "what if", which, however, are difficult to capture precisely by keyword matching but can well be approximated in this context by checking for the use of the word "if".

After categorising the type of queries, QBERT-RR identifies the topics for each question in the WTC corpus. Since QBERT-RR is trained with Yahoo! Answer dataset, it classifies the WTC corpus following the ten most popular question themes from Yahoo! Answer. It is essential to notice that there are many non-scientific questions in this list, which was part of the initial intent of the overall project: to assess the scope and breadth of the curiosity of an entire community.

When classifying the topic, QBERT-RR embeds the topics and compares the cosine similarity between the question and all ten topics. The topic that has the highest similarity with the question is considered the correct topic.

Table 6.2 shows the frequency distribution of the questions across types and topics. The most "asked" topics in the WTC corpus are science & mathematics and society & culture, which make up 67.94% of the corpus. Besides, 50.90% of the questions are HOW and WHY questions. The theme " HOW + Science & Mathematics" contains 1,387 questions, which is the highest among all 90 themes.

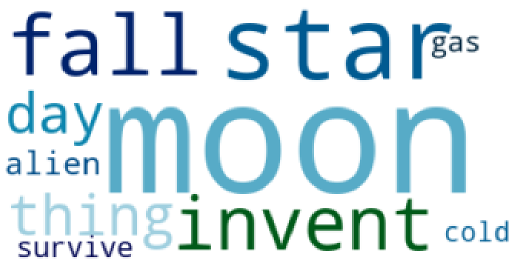
Table 6.2: Contingency table for topics and types in the WTC corpus.

	what	**when**	**where**	**which**	**who**	**how**	**why**	**if**	**other**	%
Business & Finance	80	8	18	1	21	131	168	23	141	6.87
Computers & Internet	14	2	2	0	3	38	16	3	25	1.20
Education & Reference	82	12	6	0	54	126	76	11	70	5.08
Entertainment & Music	53	10	17	0	17	51	68	32	108	4.14
Family & Relationships	26	5	6	0	3	47	78	14	64	2.83
Health	80	18	11	0	6	146	288	33	86	7.77
Politics & Government	13	4	1	0	8	28	55	22	32	1.90
Science & Mathematics	658	92	98	16	54	1387	1161	400	921	55.66
Society & Culture	155	31	22	0	48	120	304	121	255	12.28
Sports	20	4	0	0	14	38	51	8	61	2.28
%	13.73	2.16	2.10	0.20	2.65	24.56	26.34	7.76	20.50	8600/8600

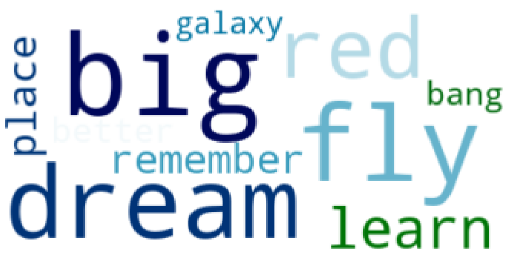
6.3.1.1 Comparing the results with LDA topic modelling

We also try to identify the topics using an unsupervised approach: Latent Dirichlet Allocation (LDA) [18]. In order to compare the resulting topics with QBERT-RR, we set the number of topics for LDA as 10. When using LDA to classify the topics, we curate the questions by removing the stop words and punctuation, lower casing the questions, and lemmatise the words. Figure 6.5 illustrates the LDA topic modelling results. It shows a combination of keywords with the highest contributions to the topic. The size of the word indicates the weight of a keyword.

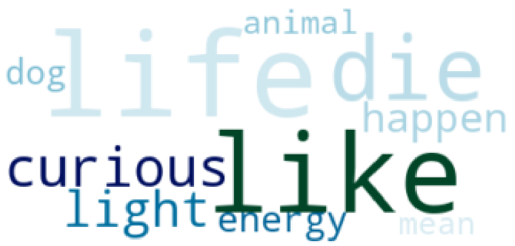
One of the limitations of using LDA is that it does not tell the exact topics. Instead, experts in the field must decide the topic for each group with the high weighted words given by LDA. For example, in topic 1, there are words like "moon", "start", and "alien". Thus, we can probably



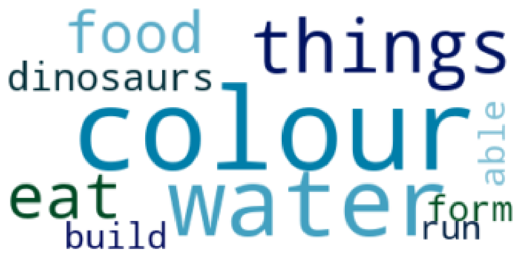
(a) Topic 0



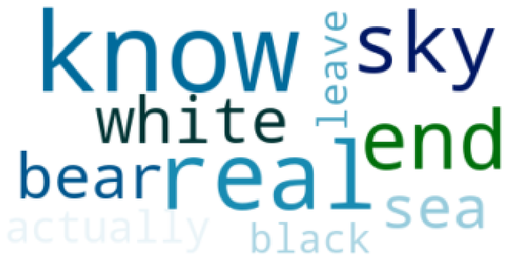
(b) Topic 1



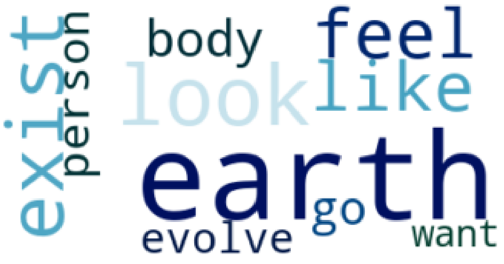
(c) Topic 2



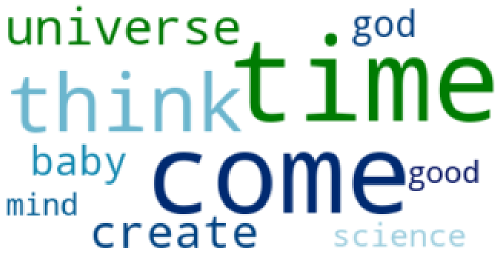
(d) Topic 3



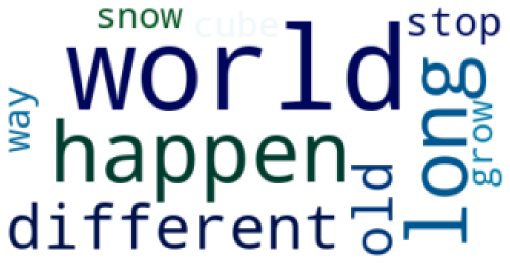
(e) Topic 4



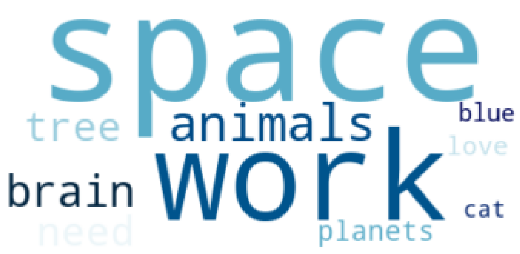
(f) Topic 5



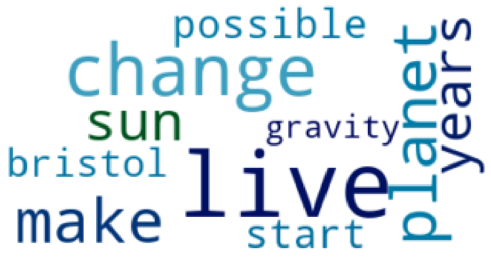
(g) Topic 6



(h) Topic 7



(i) Topic 8



(j) Topic 9

Figure 6.5: LDA topic modelling results.

conclude that topic 1 is related to space. However, sometimes it is difficult to summarise the topic from a list of given words.

6.3.2 Equivalent question recognition

The other task we introduce to process the WTC corpus is equivalent question recognition. Grouping the equivalent questions is beneficial for identifying common concerns and reducing the workload for further analysing.

We identified the equivalent questions in the corpus by mapping the questions into the 768-dimensional embedding space with QBERT-RR. These representations can be compared using cosine similarity in the embedding space. Two questions are deemed to be equivalent if their cosine similarity is above the given threshold.

To evaluate the performance of QBERT-RR on recognising equivalent questions in the WTC, we generate some question pairs and manually label them. 1,000 questions are randomly sampled from the WTC corpus. These questions are embedded by the SBERT trained with the NLI dataset [159]. After that, a list of candidate question pairs is generated depending on the cosine similarity. The top 10 questions with the largest cosine similarity in the sampling data are selected as similar question pair candidates for each query. For question pairs such as [Q1, Q2] and [Q2, Q1], we only keep one of them for annotation.

The question pairs are labelled with 0 or 1, where 0 represents different, and 1 for equivalent. The question pair is considered equivalent when the same piece of text can be used to answer both questions. We label 5,022 pairs of candidate questions in total. Of them, 728 pairs are similar, and 4,294 are different. Table 6.3 provides some examples of the data we label.

Table 6.3: Examples from the labelled WTC question pairs.

	Question 1	**Question 2**	**Label**
1	How does earth spin around?	What way does the Earth spin?	equivalent
2	Who made us?	Who invented people?	equivalent
3	How do you make glass?	How is glass made?	equivalent
4	How are mirrors made?	How are crystals made?	different
5	What will happen if the world ends?	If the world stops spinning, what would happen?	different
6	When did the humans come alive?	How did humans first exist?	different

When the cosine similarity threshold is 0.809, QBERT-RR obtains 90.76% accuracy on the sampling WTC data, which is a similar performance to QBERT-RR on the QQP dataset (90.13% accuracy). This proves that QBERT-RR can be applied to a different corpus of unseen questions. We also compare the classification results with SBERT and SBERT-QE. The results are shown in Table 6.4.

Table 6.4: Results of equivalent question recognition for the WTC corpus

	Threshold	**Accuracy**	**F1**
SBERT	0.814	85.07%	53.12%
SBERT-QE	0.833	90.08%	60.22%
QBERT-RR	0.809	90.76%	60.27%

To have a better qualitative understanding of QBERT-RR performance, we look at some examples in the WTC corpus that are labelled wrong by at least one model. Table 6.5 shows examples of question pairs with their true labels and the results from each model.

Table 6.5: Examples that are labelled wrong by at least one model.

	Q1	**Q2**	**SBERT**	**SBERT-QE**	**QBERT-RR**	**Label**
1	When will we meet aliens?	How long will it take for humans to discover alien life?	Wrong	Wrong	Correct	equivalent
2	What if there was another ice age what would happen to us?	What will we do in the next ice age?	Wrong	Wrong	Correct	equivalent
3	How was the Earth made?	When was the earth formed?	Wrong	Wrong	Correct	different
4	Why are the dinosaurs extinct?	How did the dinosaurs actually die?	Wrong	Correct	Correct	equivalent
5	Why is the North Pole so cold?	Why is it so cold?	Wrong	Correct	Correct	different
6	If you had a brain transplant would you still be ‘you’?	If you undertake a brain transplant do you implannt the persons soul?	Wrong	Wrong	Wrong	equivalent

We try to form hypotheses based on the examples. For example, from row 1 in Table 6.5, we hypothesise that “meet aliens” is the synonym of “discover alien life” and “when” is the synonym of “how long”. Other than QBERT-RR, the rest of the models wrongly label the question pairs, which can mean that QBERT-RR is better at capturing synonyms. We can prove or reject the hypothesis by removing one challenging aspect in Q2 and observing the change in the results. Table 6.6 shows the process of proving the hypothesis.

From the results, we observe that QBERT-RR is more sensitive to the synonyms such as “formed/made” and “meet/discover”. However, when it is an uncommon synonym requiring more association based on the content, all models fail to identify it. For example, from row 6 in Table

Table 6.6: Example question pair for understanding what the models learn, revealed by modifying one of the challenging aspects.

	Q1		**Q2**	**Label**
	SBERT	**SBERT-QE**	**QBERT-RR**	
1	When will we meet aliens?		How long will it take for humans to discover alien life?	equivalent
	Wrong	Wrong	Correct	
2	When will we meet aliens?		When will it take for humans to discover alien life?	equivalent
	Wrong	Wrong	Correct	
3	When will we meet aliens?		How long will it take for humans to meet aliens?	equivalent
	Correct	Correct	Correct	
4	How was the Earth made?		When was the earth formed?	different
	Wrong	Wrong	Correct	
5	How was the Earth made?		When was the earth made?	different
	Wrong	Correct	Correct	
6	How was the Earth made?		How was the earth formed?	equivalent
	Correct	Correct	Correct	

6.5, “still be you” is similar to “implant the person’s soul”.

Furthermore, we notice that the models training on question data can identify the change of interrogative words. For example, from row 5 in Table 6.6, QBERT-RR and SBERT-QE correctly label the question pair with the only difference in the interrogative words.

Another experiment we perform to recognise equivalent questions is applying a “graph community detection” method [34, 59] in order to group similar questions. To cluster the questions, a graph is built with question nodes using the cosine similarity matrix. An edge exists between nodes if the cosine similarity between a pair of questions is larger than the threshold. There are 5,930 communities found in the WTC corpus, which represents 5,930 different questions in the corpus. Of these, 5,337 questions do not have any similar question groups in the corpus.

6.3.3 Answering the Questions with a Large-Scale Knowledge Source

The last task in the question analysing pipeline is to answer the questions with an unstructured knowledge source. In this thesis, we utilise Wikipedia Summary [133] as our knowledge source, and we are interested in finding out how many WTC questions can be answered with high confidence by QBERT-RR and Wikipedia Summary. Figure 6.6 gives an overview of how QBERT-RR search the answer to a given question.

Wikipedia Summary includes the title and the first paragraph as the summary for each Wikipedia article extracted in September 2017. The raw texts of Wikipedia have 116M sentences

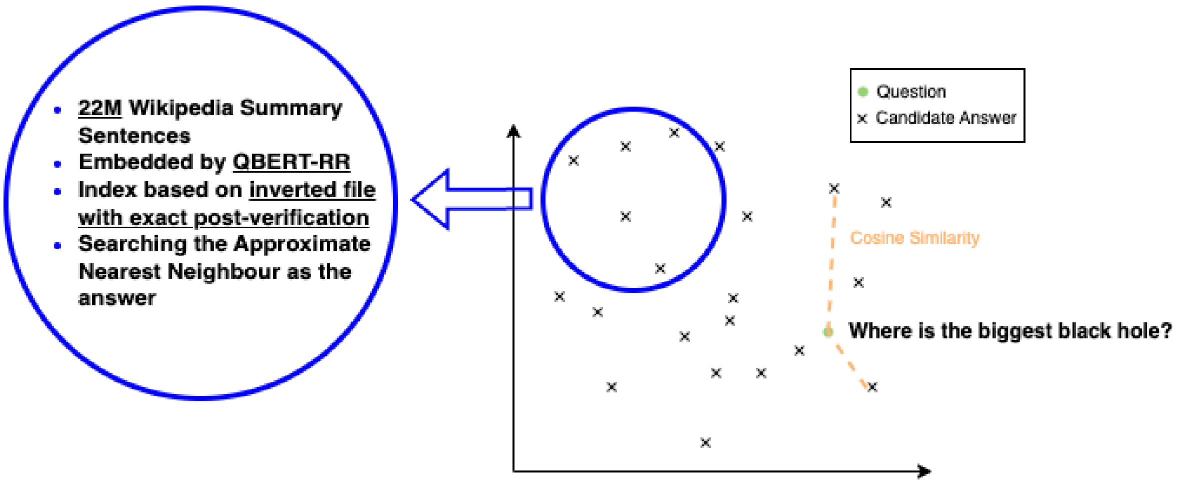

22M Wikipedia Summary Sentences
Embedded by QBERT-RR
Index based on inverted file with exact post-verification
Searching the Approximate Nearest Neighbour as the answer
Question
Candidate Answer
Cosine Similarity
Where is the biggest black hole?

Figure 6.6: Answering the questions with QBERT-RR and Wikipedia Summary. QBERT-RR embedded both the knowledge source and the questions. After embedding them, the answer is selected based on the distance to the question. It takes an average of 0.02s to search for an answer from 22M given sentences.

initially. Of these, 22M are in the summaries. After embedded with QBERT-RR, there are around 21M distinct sentences left. The summary of Wikipedia provides the article's primary information. In the meanwhile, it reduces about 80% of the sentences from the original Wikipedia.

While retrieving answers from a large-scale knowledge Source, it is important to make sure that the answer is the closest to the question and with high confidence. We define confidence score with the cosine similarity threshold. If the similarity between the question and the candidate sentence is larger than the threshold, the candidate is considered the answer to the given question. We leverage the threshold calculated in section 5.6 from the WikiQA dataset as the confidence threshold for the WTC. The best-scored sentence from Wikipedia Summary with a higher cosine similarity than 0.795 is believed to be the correct answer for a given question.

Seven GeForce GTX TITAN X GPUs are used to embed all the sentences from Wikipedia Summary with QBERT-RR. It takes 1.5 hours to encode all the 21M sentences. Due to the scale of the dataset, it is time-consuming to perform an exhaustive search among all candidates. Instead, we locate the answers to questions by using an approximate approach: approximate nearest neighbour (ANN), which trades off accuracy for searching speed. When searching with ANN, it takes a few hours to build the index for the candidates at the beginning. Nevertheless, it takes less than a second per query to search.

The Wikipedia Summary index is trained and built using the inverted file with exact post-verification for 4 hours [70]. After building the index, it takes 3 minutes to search the candidate answers for all 8,600 WTC questions with one GPU, an average of 0.02s per question.

After filtering the high confidence answers, there are 463 questions in the corpus can find an answer within Wikipedia Summary using QBERT-RR. The number of answered questions in different types and topics are shown in Table 6.7.

Table 6.7: Number of questions in WTC can be answered with high confidence over the size of the groups.

	who	what	when	where	which	how	why	if	other	All
Business & Finance	6/21	13/80	0/8	1/18	1/1	19/131	3/168	0/23	14/141	57/591
Computers & Internet	2/3	1/14	0/2	0/2	0/0	1/38	0/16	0/3	0/25	4/103
Education & Reference	5/54	10/82	0/12	0/6	0/0	6/126	1/76	0/11	2/70	24/437
Entertainment & Music	3/17	4/53	1/10	1/17	0/0	3/51	1/68	0/32	7/108	20/356
Family & Relationships	1/3	6/26	0/5	2/6	0/0	3/47	1/78	0/14	5/64	18/243
Health	0/6	4/80	0/18	0/11	0/0	5/146	0/288	0/33	0/86	9/668
Politics & Government	0/8	2/13	0/4	0/1	0/0	1/28	3/55	0/22	2/32	8/163
Science & Mathematics	8/54	64/658	11/92	7/98	0/16	121/1387	27/1161	3/400	14/921	255/4787
Society & Culture	3/48	31/155	0/31	1/22	0/0	4/120	5/304	0/121	10/255	54/1056
Sports	2/14	1/20	1/4	0/0	0/0	1/38	0/51	0/8	9/61	14/196
All	30/228	136/1181	13/186	12/181	1/17	164/2112	41/2265	3/667	63/1763	463/8600

6.4 Result Discussion of the WTC Corpus

“Project What If” was launched in 2017 across Bristol and involved thousands of people. It aimed to explore the questions of Bristolians rather than the answers, to see what they said about the local community. It aimed to observe similarities and differences in people’s curiosity regardless of their age, gender, and geography. By not setting any rules or prescribing what topics to explore, it was hoped the science centre’s exhibitions and the educational content might better reflect the interests of its community. In this section, we try to discuss the result produced by QBERT-RR and present further analysis of the data.

The automated QT analysis of the corpus revealed that more than half of the questions are in the domain of Science & Mathematics (55.66%), followed by Society & Culture (12.28%), and then by Health (7.77%). The most frequently asked type of question is the type WHY (26.34%) followed by HOW (24.56%).

It is expected to have most of the questions collected under the Science & Mathematics domain as many of the questions collected were in the science centre setting or during experiences facilitated by WTC staff. Therefore, people are already in that frame of mind. Apart from science-related subjects, the questioners are most interested in Society & Culture. The results show that the questioners are least interested in Computer & Internet related subjects.

The word cloud in Figure 6.7 illustrates the high-frequency words for each topic. We notice that words like “people” and “human” have a high frequency in most topics. Humans are naturally related to all these topics. Therefore, in order to better understand other information unique to each topic, we remove “human/people” as well as the stop words while visualising the topics. The size of the words is proportional to their frequency in the corpus.

Word clouds give us an insight into the key information held in different topics by showing high-frequency keywords. In Computer & Internet, the keywords include “computer”, “internet”, “game”, and “fortnite”, which are not spotted in other topics. In Family & Relationships, the most frequent word is “love”. Besides, it includes many words about feelings and relations. Since Science & Mathematics is a broad topic, the keywords illustrated in the figure also contain different varieties such as “earth”, “animal”, and “planet” etc. These observations again show that QBERT-RR is able to distinguish the topics in the questions.

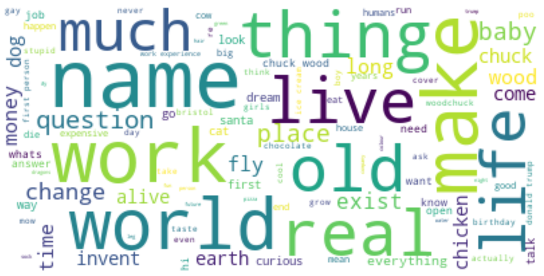
On the other hand, the type classification also reveals some interesting patterns. More than 50% of the questions are type WHY and HOW, which are causal questions, showing that the questioners went through further thinking while asking the questions. For factoid questions, WHAT questions are much more than other types like WHO, WHEN, and WHERE.

Since the questions are collected under the “Project What If”, it inspires people to ask IF questions. By navigating the IF question, we observed that most of the questions are counterfactual such as “What if we never went to sleep?” and “If you could hear in space, how loud would the Sun be?”. However, the corpus contains some factual questions in type IF as well, as would be expected in a science centre setting. For example, “I’d like to know if atoms are made up of other atoms.”. For factual questions in type IF, it can be further filtered in future work.

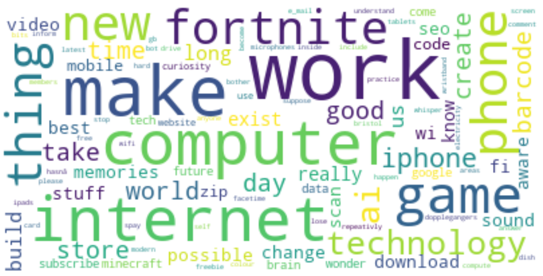
Furthermore, we calculated the $P(type,topic)$ and $P(type)*P(topic)$ to understand the associations between type and topic. Figure 6.8 illustrates the associations. The question-type WHO is strongly associated with Education & Reference and with Sport because the $P(Who,Sport)$ is 3 times larger than the probability of $P(Who)*P(Sport)$. The topic Education & References is strongly associated with types: HOW, WHAT, WHEN, WHO. The type IF associates strongly with Politics & Government, but not with Education & Reference. We also observe that people expected more explanations on the Health and Family & Relationships topic because type WHY associates more with these two topics rather than others.

One limitation of using QBERT-RR for topic classification is that it only takes the top 10 topics from Yahoo! Answer regardless of all other possible topics. Nevertheless, we notice that many questions do not belong to any of the groups in the Yahoo! Answer dataset. This can be improved by training the model with more question topics or introducing a threshold to determine the questions that are not close to any topic.

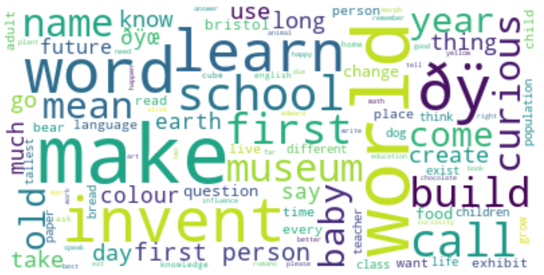
When clustering the similar questions by distance with graph community detection, we find that one of the groups has 858 questions, which by definition, are supposed to share similar meanings within the same graph. This happens because queries are connected with an edge when they have a cosine similarity larger than the threshold; however, the questions were not fully connected within the group, which means not all the questions within the subgraph are equivalent. For example, a group contains a set of questions $[Q_1,Q_2,Q_3]$. (Q_1,Q_2) and (Q_2,Q_3) are connected with an edge, respectively, but the similarity between Q_1 and Q_3 is smaller than



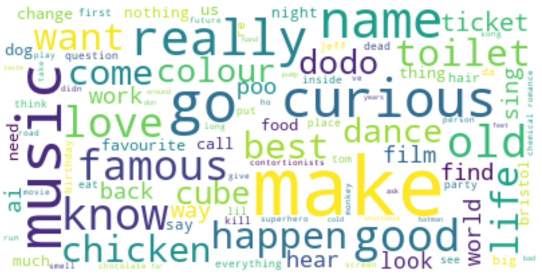
(a) Business & Finance



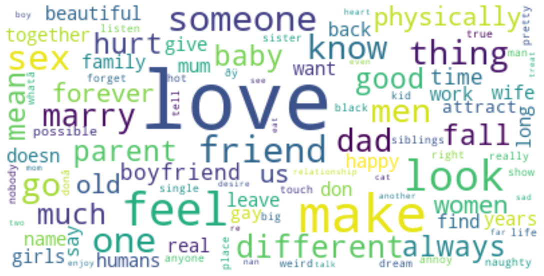
(b) Computers & Internet



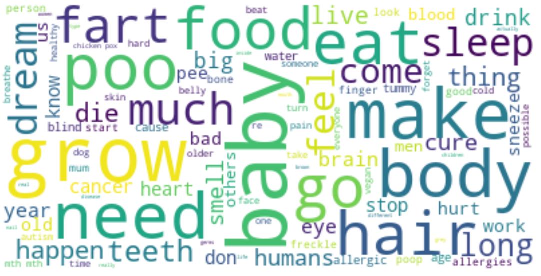
(c) Education & Reference



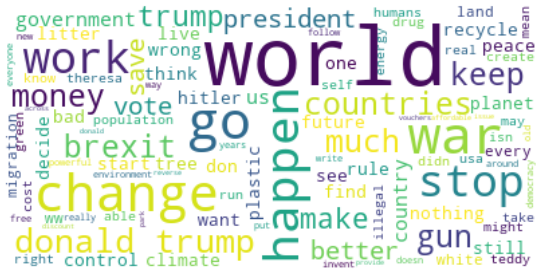
(d) Entertainment & Music



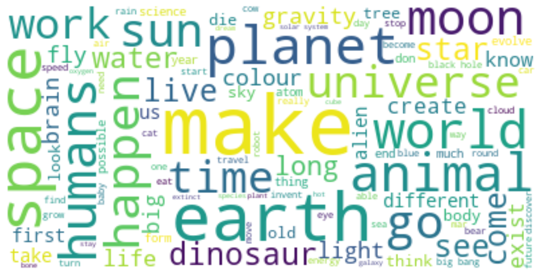
(e) Family & Relationships



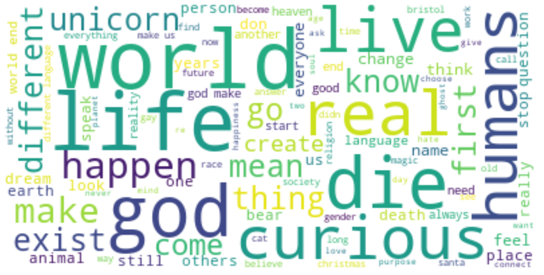
(f) Health



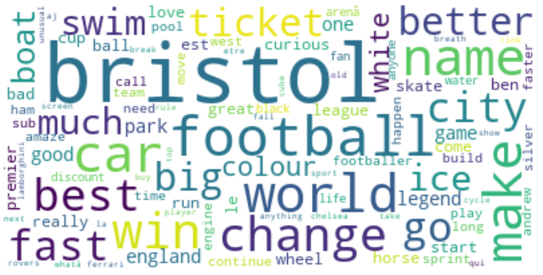
(g) Politics & Government



(h) Science & Mathematics



(i) Society & Culture



(j) Sports

Figure 6.7: High frequency words in each topic.

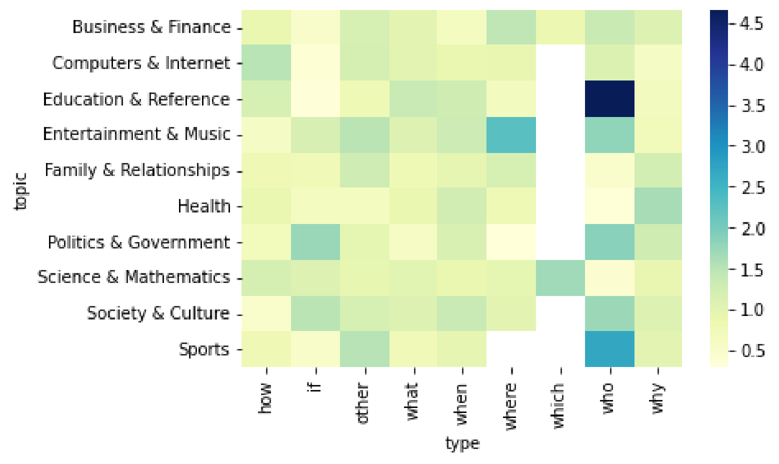


Figure 6.8: The associations between types and topics for the WTC corpus. The association was calculated with $P(type, topic)/(P(type) * P(topic))$.

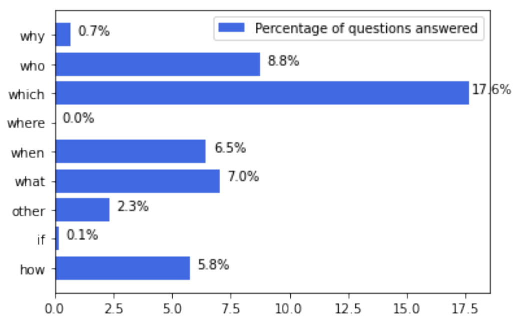
the threshold, which means that they are not an equivalent pair and are not connected. In this section, we analyse the similar questions regardless of this particular group that included over 800 queries.

The most popular questions in the WTC corpus are related to life outside of earth. There are 144 of them in the corpus, such as "When will we find intelligent life in the universe?" and " Is there any life on any other planets or solar systems in the universe?". Following that, there are 143 questions related to the end of life/world, such as "What would happen if all humans were extinct and the world stopped spinning?" and "What does it feel like on the moment you die?". Overall, 26 groups of questions have more than 10 similar queries within the datasets. 3,263 questions in the WTC corpus have at least one similar query. Thus, applying similar question detection can help reduce the corpus size for further processing for the data collector.

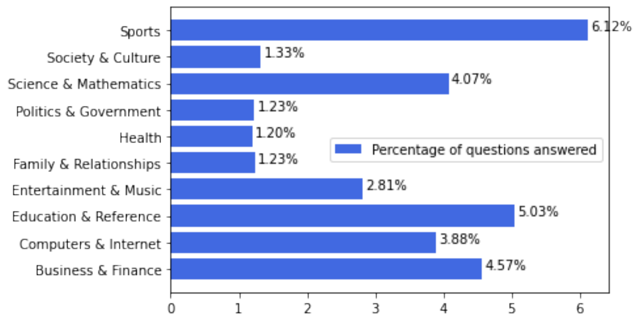
With QBERT-RR, we find answers to 463 out of 8,600 questions in the WTC corpus. However, we notice that answers with high confidence (over 0.92 cosine similarity) are equivalent questions that QBERT-RR finds in the knowledge source. For example, QBERT-RR is tricked by a sentence in Wikipedia Summary, "How did the universe come about?", and considers it the answer to the question "How did the universe begin?". We consider the questions in Wikipedia as false positive answers.

To improve the answer retrieval, we evaluate the second closest sentence instead of the closest candidate that contains "?" in the text. After filtering the false positive answers, there are 297 questions can be answered by QBERT-RR. The percentage and number of the questions that can be answered from each type and topic are illustrated in Figure 6.9.

Comparing to IF questions and confirmation questions (included in type OTHER), WH questions, such as WHICH, WHAT, HOW, WHERE, and WHO are more likely to be answered by Wikipedia Summary. Due to the QBERT mechanism, the answer is supposed to be one sentence from Wikipedia. In this case, factoid questions, which can be answered with facts



(a) Type



(b) Topic

Figure 6.9: Percentage of the questions answered by QBERT-RR.

expressed in a short and concise sentence, are more likely to be answered. This explains the reason WH questions have higher answer rates. Furthermore, non-factoid questions, like some of the WHY or IF questions, that require more explanation in the answer are harder to match with a one-sentence answer from a knowledge source.

The questions under the topic of Education & Reference, Science & Mathematics, Sports, and Business & Finance are more likely to be answered confidently by Wikipedia Summary. On the other hand, QBERT-RR can only answer around 1.33%, 1.23%, 1.20%, and 1.23% of the questions in Society & Culture, Politics & Government, Health, and Family & Friendships, respectively.

Here are some examples of the answers given by QBERT-RR. The questions are from the WTC corpus, and the answers are extracted from the Wikipedia Summary.

- **Q1:** Hello is a invented word but what does it mean?
- **A1:** Hello is a greeting in the English language. (Score: 0.895)
- **Q2:** How are clouds made?
- **A2:** On Earth, clouds are formed as a result of saturation of the air when it is cooled to its dew point, or when it gains sufficient moisture (usually in the form of water vapor) from an adjacent source to raise the dew point to the ambient temperature. (Score: 0.873)
- **Q3:** What is a black hole?
- **A3:** A stellar black hole (or stellar-mass black hole) is a black hole formed by the gravitational collapse of a massive star. (Score: 0.920)
- **Q4:** Who discovered pluto?
- **A4:** It was discovered by a team of astronomers from the Institute for Astronomy of the University of Hawaii led by David Jewitt and Scott S. Sheppard and Jan Kleyna in 2001, and given the temporary designation S/2001 J 3. (Score: 0.901)

From the question-answer pairs retrieved by QBERT-RR, we notice that the retrieved answer is adapted to a specific attribute rather than a general situation as human understanding. From Q3 in the examples, the answer adapts to “stellar black hole” instead of giving an answer to the general black hole.

QBERT-RR is only able to retrieve ONE most relevant sentence as the answer. Hence, the answer is not always included in one sentence. For example, for (Q4, A4), the answer itself might look promising. Nonetheless, the original summary contains A4 is “Hermippe, or Jupiter XXX, is a natural satellite of Jupiter. It was discovered concurrently with Eurydome . . . and given the temporary designation S/2001 J 3.”. In this case, QBERT-RR fails to capture the entity “Hermippe” related to A4, which is also essential for answering the question.

While performing question answering, we observe one particular case. For the question “What is two plus two?”, QBERT-RR gives an answer as “2 + 2 = ?”. Even though it is not the correct answer to the question, it is still interesting to see that QBERT-RR “translate” the text into notations. This can be further investigated in future research.

6.5 Chapter Summary and Conclusion

People always have curiosity and doubts. Questions have been collected from various sources during the past ten years. Access to information is easy and comprehensible through digital and online channels in our modern world. Educational institutions like schools and museums are the places that come up with most of the curiosity and questions and have therefore been challenged to adapt to this changing environment.

Therefore, we raised the research question **RQ4: how can processing questions help in education, such as in teaching and educational avenues like science centres and museums?** To answer this, we introduced a question analysis pipeline and applied it to analyse a new question corpus collected by “We The Curious” science centre. From analysing the WTC corpus, we aimed to learn more information in the questions that can assist the science centre in understanding more about their visitors and creating diverse content.

To further answer **RQ4a: What information does the WTC reveal?** And **RQ4b: Can we understand questioner’s interests by analysing these questions?** We applied QBERT-RR for question content analysis and performed question taxonomy, equivalent question recognition, and question answering tasks. By leveraging QBERT-RR, we categorised the questions into 90 themes, identified the common questions in the corpus and answered 297 questions with Wikipedia Summary. The curiosity of the question contributors was revealed from the question taxonomy and equivalent question recognition.

In the results, we see that the contributors to “Project What If” were very interested in Science, Society, and Health; and asked many questions of the WHY and HOW type. This is not surprising within the setting of the WTC as a science institution. But the next finding revealed a lot more: questions of the type IF tend to relate to Politics & Government topics and not with Education & Reference topics. Curiosity about Society & Culture is also very revealing. This is an emerging theme in the sector of science centres, where there is an ongoing discussion about expanding from Science Centres to Science & Cultural Centres. More generally, there is a movement in the sector currently to explore society and culture alongside traditional science such as Biology, Engineering, Chemistry etc. This seems to be reflected in the kind of questions Bristolians have been asking.

We believe that question-analysing specific AI models can benefit educational institutes in the future. For teaching, the teacher can understand the students' comprehension level through the difficulty of the questions. Teachers can also minimise their workload by letting the machine answer some of the questions and figure out the most common and essential problem to work on. With AI software to identify the difficulty of the questions, the students can also use the different levels of quizzes to improve learning, from conceptual questions to questions that require more comprehension of the content, even questions that expand the knowledge beyond the learning content.

Chapter 7

Conclusions

In this chapter, we will summarise the progress we have made and our contribution in each chapter. Then, we will revisit the research questions we raised in the introduction of this thesis, followed by a discussion of limitations in our research and possible future work and direction.

7.1 General Discussion

Questioning is an essential process of thinking and learning. For many years, questions have been used as a medium for teachers to diagnose students' understanding through exams [47, 155] and evaluate higher order thinking [44], and for students to direct learning and drive knowledge construction [31]. Outside educational institutions, there is an increasing trend of people using smart devices and online communities to find explanations or show learners' interests by asking questions. Hence, analysing questions has the possibility to benefit more than teaching and learning. For example, we can understand a community by processing their questions. Moreover, the increasing amount of questions posted online makes labelled Question data more accessible for training a supervised model to process growing questions without human intervention. The initial goal of our research was to process and analyse the Questions, a special type of short text, automatically with deep learning and NLP approaches.

In chapter 2, we gave some preliminaries for deep learning in NLP, including some popular techniques and neural networks that have been used in NLP research. Based on the objectives, chapter 3 reviewed the background research about deep learning approaches in various NLP research and applications on sentence embedding. With these chapters, we established this thesis's research scope and approach.

During our investigation, we noticed that the pre-trained language models had achieved impressive results on various NLP tasks, such as sentiment analysis, text classification, text similarity, and question answering [43, 118, 127]. They were pre-trained on unlabelled data and fine-tuned on specific tasks. One of the pre-trained language models, BERT, had shown an ability to process large-scale text and language [43]. Besides, it provided an easy route to fine-tune the model for specific language contexts and problems. This motivated us to understand BERT and fine-tune it for specific tasks in the question domain.

As a language model, BERT was pre-trained on unlabelled large text corpora to understand language and was able to generate dense word and sentence vector representation for downstream tasks. Research showed that embedding could capture the information that had not been shown during training, such as tense, gender, and polarity [17, 50, 101]. One problem that occurred while using dense representation in NLP was that the coordinates' value was not considered individually meaningful. Compared to statistic-based embedding, it was harder to interpret the information used by the AI system. This limited the possibility of explaining the decisions of a system or auditing the system for possible biases. As a result, one of the goals of this thesis was to understand how and what BERT learns. Chapter 4 explained the mechanism of BERT, including how to pre-train BERT and fine-tune it for capturing sentence semantic similarity. We also proposed a linear system to decompose the embedding into its attributes. Moreover, in chapter 4, we generated a simple sentence corpus for sentences that contain only the subject, verb, and object with the same preposition and articles and embedded it using BERT. We found that BERT sentence embedding was compositional because it could be decomposed into word attributes by solving the linear system. The attribute representations could predict the sentence embedding of an unseen sentence. The predictions achieved 98.4% similarity compared to the BERT sentence embeddings.

On the other hand, fine-tuning BERT was challenging because the model scale was large and the labelled data were limited. Therefore, we intended to train a generalist model that was good at multiple tasks for processing questions, even if it did not excel at any specific task. Chapter 5 introduced a generalist model called QBERT. During training, QBERT was optimised by the task-specific loss function. On the other hand, during inference, a similarity threshold was leveraged for some classification tasks. The simplified QBERT without task-specific layers could process questions in terms of question classification, equivalent question recognition, and question answering with one model that trains all tasks simultaneously without identifying a primary task. In addition, we trained and evaluated the model on multiple question datasets. One version of QBERT we trained with the RR curriculum obtained a balanced performance among all question tasks. It also created a similar result compared to the single-task models that fine-tuned multiple SBERTs for separate tasks.

In chapter 6, we presented a pipeline for analysing question corpus. Then we applied QBERT-RR on a new question corpus (WTC corpus) collected by "We The Curious" science centre in

Bristol as an example of our pipeline. With this experiment, we explored the possibility of understanding a community (in our case, science centre visitors) by processing the questions they raised. A question taxonomy, including type and topic classifications, was leveraged to provide an interpretation of the questioners' interests. The questions were divided into nine types and ten topics. The results showed that people were most curious about Science & Mathematics, followed by Society & Culture and Health. We also identified the popular questions by grouping similar questions in the corpus. Moreover, QBERT-RR attempted to answer these questions with one sentence from Wikipedia Summary. 294 questions were answered with confidence within the WTC corpus.

7.2 Research Questions

At the beginning of the thesis, we raised four research questions. In this section, we will attempt to answer these questions by summarising the results obtained in this thesis.

RQ1: How does BERT learn a sentence representation?

BERT was constructed by the Transformer encoder that took token inputs and encoded them into dense vector representations. It was pre-trained with large general corpora such as English Wikipedia and book corpus. During pre-training, it performed masked language model task, which predicted the random masked word in the input sequence, and next sentence prediction task, which identified if two input sentences were adjacent. These two tasks helped BERT learn the bi-directional context within the sequence and the content. The output of the special token [CLS] was usually considered as the representation of the entire input sequence. By fine-tuning BERT with natural language inference datasets, it could capture the semantic similarity between sentences.

What properties does BERT sentence embedding contain?

We then decomposed the sentence representation with a linear system. When solving the linear system, we assumed that there is a linear relationship between the sentence embedding and its word attributes. The statistical testing results showed that BERT sentence embedding could be reconstructed by adding up the representation of the attribute learned by the linear system. 99.5% of the time, the reconstructed embedding can retrieve the actual sentence embedding from the corpus.

RQ2: What information can we learn from Questions?

When reviewing the existing work on question processing, we noticed that question processing was intended to assist early question answering system [52, 89]. But as a special type of text, questions can reflect more information, such as the questioner's concern and ability. Thus, we included three different tasks to process the questions. These tasks could help identify the questioners' interests and reflect their level of thinking.

What tasks can be included in processing questions?

While processing questions, we introduced three different tasks. Question taxonomy decides the type and topic of the questions. Equivalent question recognition identifies similar question pairs in different wordings and retrieved all similar questions within a corpus for a given query. Question answering task retrieves a one-sentence answer from a given knowledge source.

Type classification categorised the questions based on interrogative words, such as what, who, where, if, etc. The level of thinking could be revealed by further analysing the types. Topic classification paid attention to the theme of the questions, which was able to indicate the questioners' curiosity. Grouping similar sentences contributed to the analysis in two different ways. First, it can reduce the workload for further analysis by removing similar queries or direct questioners to the existing similar questions. Second, from the similar question groups, we could also understand the common concerns of the community. Finally, we included question answering to comprehend what kinds of questions can be automatically answered by a general knowledge source (Wikipedia Summary).

RQ3: How does multi-task learning affect the performance of processing questions?

In chapter 5, we trained QBERT on Yahoo! Answer datasets, QQP, WikiQA, and SQuAD without identifying the primary task. The aim was to use one model to generate vector representations that can be utilised for multiple tasks. We investigated the difference between two training curricula: one-by-one strategy and fixed-order round robin. QBERT-OBO trained the dataset one after the other following a certain order. By contrast, QBERT-RR trained small batches of data from each dataset jointly at each step. QBERT-OBO performed worse than the specialists on topic and similar question pair classification. This might happen because the OBO curriculum resulted in forgetting the tasks that were learned in the earlier training stage.

Can one model perform multiple tasks for processing questions?

The results showed that the generalist model QBERT-RR turned out to achieve similar performance to the specialist models in question classification, similar question recognition, and question answering for WikiQA dataset. However, the retrieving accuracy for the generalist model on SQuAD is 7% lower than for the specialist. Even though QBERT-RR did not surpass the specialist model in all the tasks, it was able to perform multiple tasks by measuring the cosine similarity between the general representations.

Does adding more question-related data improve the performance on a specific task?

Furthermore, we fine-tuned QBERT on each task to investigate if in-domain data will improve the performance of QBERT. QBERT learned all the additional tasks in a fixed-order round robin curriculum and was fine-tuned on the primary tasks. Nevertheless, adding more data on processing specific question tasks was negligible. The reason behind this requires further investigation.

RQ4: How can processing questions help in education?

Though out the WTC corpus study, we investigated the possibility of processing questions by demonstrating a pipeline for analysing questions with QBERT-RR. In the past few years, "We The Curious" collected more than 10,000 questions from various sources. These questions covered all types and domains without any human labels. By leveraging QBERT-RR, we provided a taxonomy for the questions collected and identified the common questions in the corpus. Furthermore, QBERT-RR answered 297 questions with the answers from Wikipedia Summary.

What information does the WTC corpus reveal?

In the results, we discovered that the contributors to the WTC corpus were interested in Science, Society, and Health. This could be common within the setting of a science institution. Regardless, we also noticed that questions of the type IF tended to relate to Politics & Government topics rather than to Education & Reference topics. This could indicate that more critical thinking and hypothesis were involved in Politics. Curiosity about Society & Culture is also very revealing in the WTC corpus, which supported the potential movement in the sector to explore society and culture alongside traditional science. Among all these questions collected, Bristolians were most concerned about "When will we find intelligent life in the universe?". In addition, WH questions were more likely to be answered by QBERT-RR. Since Wikipedia Summary was used as our knowledge source, topics like Sports, Science & Mathematics, and Eduacation & Reference had higher answered rates.

7.3 Limitation and Future Work

Despite all the contributions mentioned in this thesis, our studies still have some limitations. In this section, we will outline the limitations of our work and provide a suggestion on some potential future work.

While decomposing the BERT embedding, we generated a simple sentence corpus in order to understand the properties of the embedding. The length and properties within the sentence had a strict set-up. However, in the real world, the sentences are more complicated than "The cat sat on the chair.". Our study only provided a basic possible approach to decomposing the embedding. Considering this, future work can focus on decomposing more complicated sentences or even more attributes rather than words, such as bias. Furthermore, applying the decomposition on downstream NLP tasks, such as answering counterfactual questions, is also an interesting direction.

A further prominent limitation of the QBERT proposed in our study is that QBERT retrieved a sentence as the answer instead of a phrase from the given content. In our research, we built our model based on the WikiQA dataset, which identified if the sentence from Wikipedia Summary was the correct answer. This strategy worked on WikiQA. However, when we applied it to the WTC corpus, we noticed that the answer was not always included in one completed sentence. The answer needed to combine the information from a few sentences or a paragraph. One possible solution for this problem is to relax the one-sentence policy to the entire summary (paragraph). This could be a challenge for embedding and results in less concision in the answer. However, this can be improved by adding a “reader” that extract the exact answer after retrieving the related paragraph from Wikipedia.

We believed that the analysis of the WTC corpus was just a start for processing questions to reveal the contributors’ curiosity and to contribute to education and other domains. In our study, we used keyword matching to categorise the type of questions, which provided a coarse classification. To further investigate the level of the questions, the question types could be further classified into hypothesis, facts, opinion, etc. This could be improved by collecting more labelled data on question types in the future.

Appendix A

Appendix A Data Declaration

All datasets [1,2,3,4] used to train the QBERT are open assessed for research purposes. For the question corpus produced by “We The Curious” science centre, “We The Curious” is responsible and accountable for protecting the personal data of individuals submitting this information alongside their questions. All personal data is held by “We The Curious” in compliance with GDPR protocol, and personal data is not shared with other parties, including the analysis team of this study. For the purpose of the present study, a smaller dataset was generated by removing all the personal data that was associated with the questions, and only this was shared with the analysts. The anonymised and moderated dataset is available from We The Curious on reasonable request for research purposes [5].

[1] QQP: `http://qim.fs.quoracdn.net/quora_duplicate_questions.tsv`

[2] WikiQA: `https://www.microsoft.com/en-us/download/details.aspx?id=52419`

[3] SQuAD: `https://rajpurkar.github.io/SQuAD-explorer/`

[4] Yahoo!Answer: `https://www.kaggle.com/datasets/jarupula/yahoo-answers-dataset`

[5] Please contact `information@wethecurious.org`

APPENDIX B

Appendix B An Early Study on Processing Questions

B.1 Training and Fine-tuning

Following S-BERT, we trained the model with classification tasks. The basic S-BERT was only trained on natural language inference dataset and semantic textual similarity dataset that contain sentence pairs with labels such as SNLI [22], NLI [159], and STS [26] dataset. Thus, it has poor performance in detecting similar question pairs. Following the architecture of S-BERT in figure 4.2, we trained the sentence embedding model to learn the similarity between questions with data from QQP. The length was limited to 35 tokens for each input sequence because 99.93% of the questions are shorter than 35 words in the WTC corpus. The sequences with more than 35 words were truncated after the limited length.

We have used the classification technique described in S-BERT [125] for QE and QA classification. For QQP and WikiQA datasets, the model took questions pairs or questions answer pairs as the input and produces the label in terms of 1 or 0. 1 represents that the input sequences are similar or related. Each input sequence was tokenised and embedded by BERT-base then produce an embedding with 768 dimensions. BERT-base used in this model is a smaller BERT version containing 12 layers and 110M parameters. All the weights in BERT were updated during training. Comparing to softmax loss in S-BERT, the contrastive loss is more capable of mapping the similar vector in high dimensional space into nearby points in a lower dimension [58]. Hence, we minimised the online contrastive loss and optimised it by Adam optimiser with a learning rate of 2e-5. The contrastive loss combines loss from both positive samples and negative samples with a margin of 0.5. The margin ensures that negative samples have a more significant distance than the margin value. YQA was introduced as a supplement dataset for QA task because WikiQA did not have enough data considering the size of the model. Since YQA only contains corresponding question-answer pairs, multiple negatives ranking loss that requires only positive labels is applied instead of online contrastive loss.

In QE and QA, instead of classifying if the sequences are related, it is more important that the system can retrieve all the related sequences for given questions. The problem is how to quantify ‘related‚Äô with embeddings. A cosine similarity threshold was introduced in this model. First, all the sequences in the training set were embedded with the fine-tuned model. The sequence pairs were classified as positive if they have higher similarity than the threshold. We used 2 different strategies to decide the threshold for QE and QA. For identifying similar question pairs, the similarity threshold with the best accuracy in the QQP was found to quantify any questions pairs during training. On the other hand, for question-answer pairs, we leveraged the threshold with the best precision in WikiQA instead. While retrieving answers from the knowledge base, there are usually millions of candidates and we wanted the answer to be as reliable as possible. With both sequence embeddings and the threshold observed above, the model is capable to classify and search all the related sequences in the corpus by calculating the cosine similarity between sequences.

Contrary to QE and QA, QT took one question as input and predicted the question topic with the embedding. We have used the similar classification technique described in S-BERT [125], but only one BERT was needed in the network. An additional softmax layer was applied after BERT to map the embedding into probability for each topic. We fine-tuned the trained BERT and the softmax layer with extra data in YT.

QBERT was trained for 10 epochs, respectively for each dataset, with the training data divided by the data provider for each task. Except for YT, all the data was applied during training and trained for 5 epochs. For QE and QA, the network was trained with a batch size of 150. Moreover, for QT, the batch size was 350. The model was evaluated by accuracy for all classifications. In answer retrieval, we evaluated accuracy, precision and recall for the first answer.

In this experiment, the networks were trained with 1 GeForce GTX TITAN X GPU. It took 7 hours, 0.5 hours, 9 hours, 16.5 hours to train on QQP, WikiQA, YQA, and YT, respectively.

B.1.1 Performance of the model

The results for models fine-tuned with different datasets are illustrated in table B.1 and table B.2. In QE and QA tasks, the embedding network was trained with pairs of sentences, so with more semantic textual similarity datasets, they both achieve better performance. However, for QT, it was trained with a different structure fine-tuned based on QE and QA results with the same YT dataset. QT task does not require the embedding model to capture the sentence similarity. Therefore, pre-training with more semantic textual similarity datasets does not significantly affect the result of QT. Furthermore, the fine-tuned QT model performs worse on other tasks because it only trained with one BERT and limited the performance on capture sentence similarity.

While pre-training the sentence embedding model with all three datasets for QE, we observed that the order in which the training data are presented has a great effect on the final result. As

Table B.1: Performance of QT, QE, and QA classification tasks. The models are trained with different datasets. The model's name indicates the training sequence of each dataset.

	QT		**QE**			**QA - Classification**		
	Accuracy			**Accuracy**			**Accuracy**	
	Train	**Test**	**Threshold**	**Train**	**Test**	**Threshold**	**Train**	**Test**
STS+NLI	85.71%	72.44%	0.800	74.39%	74.77%	**0.910**	**94.89%**	**95.24%**
QQP	82.86%	72.32%	0.850	90.48%	89.59%	-	-	-
QQP+WikiQA	86.21%	72.32%	0.825	91.42%	85.44%	0.713	99.99%	94.70%
QQP+WikiQA+YQA	86.57%	72.51%	0.825	82.08%	80.13%	0.755	95.16%	94.74%
YQA+WikiQA+QQP	86.78%	72.14%	0.875	99.20%	90.29%	0.797	96.34%	93.92%
QQP+YQA+WikiQA	86.58%	72.37%	0.850	78.27%	76.49%	0.756	99.97%	95.18%

Table B.2: Performance of QA Retrieval tasks. The models are evaluated on the WIKIQA dataset. The model's name indicates the training sequence of each dataset.

	QA - Retrieval					
	Accuracy@1	**Precision@1**	**Recall@1**	**Accuracy@1**	**Precision@1**	**Recall@1**
	Train			**Test**		
STS+NLI	21.64%	21.64%	19.74%	29.88%	29.88%	27.21%
QQP	-	-	-	-	-	-
QQP+WikiQA	82.06%	82.06%	78.53%	28.63%	28.63%	27.07%
QQP+WikiQA+YQA	53.81%	53.81%	50.53%	46.47%	46.47%	43.36%
YQA+WikiQA+QQP	60.92%	60.92%	57.48%	37.76%	37.76%	34.85%
QQP+YQA+WikiQA	85.17%	85.17%	81.28%	46.06%	46.06%	42.36%

shown in table B.1, with the same datasets, the model trained with QQP as the last outperforms the model trained QQP first around 10.16%, from 80.13% to 90.29%. Besides, the cosine similarity threshold increases from 0.825 to 0.875 means that the questions with similar meanings are closer to each other in vector space.

Possible remedies to this effect will be the object of a separate study, as they are not relevant to the problem we are addressing in this paper.

In QA, we fine-tuned S-BERT on the classification task and evaluated it on both classification and retrieval tasks. Similar to QE, the best threshold contains more datasets from different tasks. In the classification task, the original S-BERT trained on STS+NLI outperforms other models. However, this is due to the bias of WikiQA dataset. 94.89% of the question-answer pairs in the training set of WikiQA are labelled as 0, and 95.24% in the test. S-BERT does not manage to identify the correct answer, and it only uses a large threshold to ensure that all the question-answer pairs are categorised as negative. WikiQA only contains question-answer pairs in the dataset. In order to evaluate the performance on retrieval task, a knowledge base that includes all the candidate sentences in WikiQA dataset was generated. And during evaluation, we leveraged only questions with a correct answer in the answer base. As shown in Table B.2, in QA retrieval, training with extra YQA data dramatically increase the accuracy of the WikiQA test set.

According to QE and QA performance in Table B.1 and Table B.2, we noticed that the best models for each specific task might have poorer accuracy on another. Besides, both of the models with the best performance are over-fitted on the training set. Therefore, we used the model following the training sequence of QQP, WikiQA, YQA, which has a more balanced performance on all three tasks, as our multi-tasking generalist QBERT. We applied this generalist QBERT to our task of the content analysis of the WTC corpus.

APPENDIX C

APPENDIX C WTC QUESTIONS ANSWERED BY QBERT-RR

The questions in this section originated from the WTC corpus. The answers are sentences retrieved from Wikipedia Summary. [1]

[1]The data from the WTC corpus can only be used for research purposes and cannot be used without permission. To access the data, please contact "We the Curious" Science centre: information@wethecurious.org

	Question	**Answer**
0	What does animals eat	They eat insects, frogs, freshwater fish, and small mammals.
1	How much seats are in a stadium	The figures generally represent the licensed capacity of the venue, which is usually far higher than the number of seats in the stands.
2	How long does it take to make a game?	One playthrough of the game takes about 5 to 10 hours.
3	What is 'time'?	Time is the indefinite continued progress of existence and events that occur in apparently irreversible succession from the past through the present to the future.
4	How does your heart pump blood around your body	Coordinated contractions of cardiac muscle cells in the heart pump blood out of the atria and ventricles to the blood vessels of the left/body/systemic and right/lungs/pulmonary circulatory systems.
5	How do clouds float without coming down	On Earth, clouds are formed as a result of saturation of the air when it is cooled to its dew point, or when it gains sufficient moisture (usually in the form of water vapor) from an adjacent source to raise the dew point to the ambient temperature.

	Question	**Answer**
6	How do clouds float?	On Earth, clouds are formed as a result of saturation of the air when it is cooled to its dew point, or when it gains sufficient moisture (usually in the form of water vapor) from an adjacent source to raise the dew point to the ambient temperature.
7	What is a nuclear war?	A Nuclear War is a war in which nuclear weapons are used.
8	Why is the sea blue?	This is the main reason why the ocean is blue.
9	What is matter	Matter is the substance of which objects are made.
10	Whats 10 +10	9,Äö 10).
11	How did you do that	I did this...
12	What does antidisestablishmentarianism mean?	Antiestablishmentarianism (or anti-establishmentarianism) is a political philosophy that views a nation's or society's power structure as corrupt, repressive, exploitative, or unjust.
13	What time is it supposed to get to	It clocks at 3:35.
14	How old is Oscar tarryt	He was born in 1977 and died in June 1995 at the age of 18.
15	circuses r Cl!	.
16	How does it work	It doesn't really work.
17	What is adhd	Attention deficit hyperactivity disorder (ADHD) is a mental disorder of the neurodevelopmental type.
18	Who am I.	I Am I.
19	Who am I	I'm me.
20	How does it work.	It doesn't work like that.
21	What is density	Density and dense usually refer to a measure of how much of some entity is within a fixed amount of space.
22	How does it work	It doesn't really work.
23	What is an atom made of	They are typically composed of thousands of atoms or more.
24	What is an atom made of	They are typically composed of thousands of atoms or more.
25	How is a baby born	In mammals, pregnancy is the period of reproduction during which a female carries one or more live offspring from implantation in the uterus through gestation.
26	What time is it	26:20).
27	What are atoms made of	They are typically composed of thousands of atoms or more.

	Question	**Answer**
28	What is the meaning of life	(The) Meaning of Life may also refer to: Monty Python's The Meaning of Life, a 1983 film by Monty Python Monty Python's The Meaning of Life (album), album by Monty Python in conjunction with film of same name, also contains a song titled "The Meaning of Life" The Meaning of Life (TV series), Irish television series presented by Gay Byrne The Meaning of Life (animated film), animated short film written and directed by Don Hertzfeldt The Meaning of Life (Tankard album), album by Tankard, also contains a song by the same name "The Meaning of Life" (The Offspring song), song by The Offspring from the 1997 album Ixnay on the Hombre M.O.L., acronym for "Meaning of Life", a DVD by Disturbed "Meaning of Life", a song by Disturbed from The Sickness Meaning of Life (album), by Kelly Clarkson Authors of books titled The Meaning of Life or similar include: Alfred Ayer (The Meaning of Life and Other Essays) Bradley Trevor Greive (The Meaning of Life) C.E.M.
29	How many animals are in the world	The world population is thought to be around 100,000 animals.
30	What is the capital of Australia	Canberra (or) is the capital city of Australia.
31	How do you do it	There's a bunch of different ways to do it.
32	It was really amazing	It was amazing.
33	How do LEDs work.	LED (Light-emitting diode) to LED communication, also known as ‚Äö Mobile Device Light Communications‚Äö are sundry techniques which repurpose mobile computer devices, including mobile telephones, to communicate with other devices, using visual or infrared light.
34	What is the planetorium	Terrestrial planet, a planet that is composed primarily of silicate rocks or metals.
35	How old is earth	It is approximately a billion years old and is located about 5,400 light years (ly) from the Solar System and 1,100 ly above the plane of the Milky Way galaxy.
36	What is gravity	Gravity is defined as the resultant of gravitation and the centrifugal force caused by the Earth's rotation.
37	How do you calculate quantum gravity	Quantum gravity (QG) is a field of theoretical physics that seeks to describe gravity according to the principles of quantum mechanics, and where quantum effects cannot be ignored.
38	What is the colour of the white horse	It is one of the most common horse coat colors, seen in almost every breed of horse.

	Question	**Answer**
39	How big is the Earth	It is 450 m long and 150 m wide and has a surface area of 45,5 decares.
40	What is a black hole	A stellar black hole (or stellar-mass black hole) is a black hole formed by the gravitational collapse of a massive star.
41	How heavy is gold	The collection contains 132 pieces of gold and has a total weight of more than 1,660 troy ounces (52 kg).
42	Why is the sky blue	The same elastic scattering processes cause the sky to be blue.
43	When was earth really created	It first appeared about 205 million years ago.
44	What is the longest time humans can live	The maximum life span is 2 to 2.5 years.
45	Why is the ocean blue?	This is the main reason why the ocean is blue.
46	I love this place	I love the place.
47	I don't know	I don't know.
48	How do aroplanes fly	The aircraft flies at high speed (approximately 100 mph) just above the surface of a lake or reservoir, scooping up copious amounts of water into its belly.
49	How old is the sun	It appears to be roughly the same age as the Sun; around four billion years.
50	What is a distance	Meters) or topological distance (e.g.
51	How was the earth made	It was created by tectonic movements of the Earth's crust.
52	How many strands of dna are in a average human	This genome is approximately 25,000 base pairs long and organised into twelve segments.
53	How big is the moon in millimetres	(Modern estimates are more conservative, giving the apparent size as one-half to two-thirds the diameter of the full moon).
54	How fast do rockets go	Rockets can be fired out to a range of 25 kilometres.
55	How far is mars from earth	It is indeed so close that it orbits Mars much faster than Mars rotates, and completes an orbit in just 7 hours and 39 minutes.
56	How many stars are in the space	The most voluminous modern catalogues list on the order of a billion stars, out of an estimated total of 200 to 400 billion in the Milky Way.
57	How long do fish live	Its lifespan is up to 3 years, but most fish do not exceed two.
58	What is the periodic table	The Periodic table is the periodic table of chemical elements.
59	How many stars are there	It consists of about 20 stars.
60	What is a hedgehog habitat.	The habitat consists of waste grounds, woodland fringes and hedgerows.

	Question	**Answer**
61	How is glass made	It is produced by casting glass onto a table and then subsequently grinding and polishing the glass.
62	When is the world going to end	NO one knows when the world will end.
63	When was stone age	It is thought to date from around 3000 BC.
64	How does Saturn have rings	The rings of Saturn are made up of objects ranging in size from microscopic to moonlets hundreds of meters across, each in its own orbit around Saturn.
65	Are you single?	I'm single.
66	who invented the	It was invented by Drs.
67	How are clouds made	On Earth, clouds are formed as a result of saturation of the air when it is cooled to its dew point, or when it gains sufficient moisture (usually in the form of water vapor) from an adjacent source to raise the dew point to the ambient temperature.
68	How long is the Milky Way	It spans about 35' on the sky which translates to a true radius of 7 light years.
69	When is the end of the world going to happen	NO one knows when the world will end.
70	What day is it	24 Mar.
71	How hot is the sun	It is radiating (after allowance for ultraviolet radiation) 1,340 times the Sun's luminosity from its photosphere at an effective temperature of 12,120 K.
72	Where is gravity	Its center of gravity lies in the equatorial plane.
73	How big is the moon	It is 603 kilometers in diameter, overlapping the near and far sides of the Moon.
74	How small are atoms	Atoms are very small; typical sizes are around 100 picometers (a ten-billionth of a meter, in the short scale).
75	Why is this here	:
76	How many earthquakes has the earth made.	About three thousand earthquakes occur on average each year with magnitude up to eight points.
77	How old is Saturn 5	It is relatively young for a star, with an estimated age of 32 million years.
78	Who was the founder of We the curious	It was developed by Dean William S. Taylor in the early 1920s.
79	How old is Saturn 5	It is relatively young for a star, with an estimated age of 32 million years.
80	How did the Big Bang theory come about	It was originally suggested as a phase of the cyclic model or oscillatory universe interpretation of the Big Bang, where the first cosmological event was the result of the collapse of a previous universe.
81	How was gravity made	In its original concept, gravity was a force between point masses.

	Question	Answer
82	How much is it	200,000.00.
83	Who is the oldest person in the world	The oldest verified man is Jiroemon Kimura, also from Japan, who died 12 June 2013, aged 116 years, 54 days.
84	How fast does the speed of light travel.	nearly a quarter the speed of light).
85	How many stars are there in our galaxy.	Circa 1,000 stars are supposed to exist within the galaxy.
86	What is the biggest dinosaurs	It includes some of the largest and heaviest dinosaurs to ever walk the earth.
87	Hello how are you.	How are you!
88	Rerffdbbfffggghhjhfhyuuijk5fgfhjjkuk ntyrtythngffgfghghgggcgfbgjhgh yfjgfdyjkuft cewghjdl j,hfhjrwjhrjhjerb&bw guhfr,ehhrwef struisdyieifuydehhedwdhjhgggggggggggggggyyhuihyufjhyrtre thjutttryjttnumhkhfytygnghhdrtgjhmhffthmyj,frsethtmfjf,drynyftmuyjf,yufmfyu yfufyu	O O OIWEQPOISDFBKJFOIWEQPOISDFBKJFOIWEQPOISDFBKJFOIWEQPOISDFBKJF EDGHOUIEROUIYWEVDGHOXUIEROIYWEVDGHEOXUIEOIYWEVDGHEOXUIEOIYWE KJBSVDBOIWERTBAKJBSVEDBOIWRTBAKJBSOVEDBOWRTBAKJBSOVEDBOWRTBA SFDHNWECTBYUVRGSFDHNYWECTBUVRGSFDHCNYWECBUVRGSFDHCNYWECBUVRG HNOWFHLSFDGWVRGHNOWFGHLSFDWVRGHNOWSFGHLSDWVRGHNLOWSFGLSDWVRG YPOWVXTNWFECHRGYPOWVEXTNWFCHRGYPOWNVEXTNFCHRGYPWOWNVETNFCHRG SVYUWXRGTWVETUISVYUWVXRGTWVETUISVYUWVXRGWVETUISVYUWVXRGWVETU WVERBYOIAWEYUIVWVERBEYOIAWEYUIVWVERBEYOIWEYUIVWLVERBEOIWEYUI EUIOETOUINWEBYOEUIOEWTOUINWEBYOEUIOEWTOUNWEBYOETUIOEWOUNWEBY WFVEWVETN9PUW4TWFVEWPVETN9UW4TWFVETWPVET NOUWQERFECHIBYWNOUWQXERFECIBYWNOUWFQXERFCIBYWNOFUWFQXRFCIBYW VEHWETUQECRFVE[VEHWERTUQECFVE[VEHWQERTUQCFVE[VEOHWQE UIWTUIRTWUYWQCRUIWTUYIRTWUWQCRUIWTXUYIRTUWQCRUIBWTXUYRTUWQCR IYPOWOXNPWTHIECIYPOWTOXNPWHIECIYPONWTOXNWHIECIYLPONWTXNWHIEC R9UHWVETPUNRQYBR9UHWVETPUNRQYBR9UHWVETPU Figure 4: A stereogram signature.

	Question	Answer
89	What is the meaning of life.	(The) Meaning of Life may also refer to: Monty Python's The Meaning of Life, a 1983 film by Monty Python Monty Python's The Meaning of Life (album), album by Monty Python in conjunction with film of same name, also contains a song titled "The Meaning of Life" The Meaning of Life (TV series), Irish television series presented by Gay Byrne The Meaning of Life (animated film), animated short film written and directed by Don Hertzfeldt The Meaning of Life (Tankard album), album by Tankard, also contains a song by the same name "The Meaning of Life" (The Offspring song), song by The Offspring from the 1997 album Ixnay on the Hombre M.O.L., acronym for "Meaning of Life", a DVD by Disturbed "Meaning of Life", a song by Disturbed from The Sickness Meaning of Life (album), by Kelly Clarkson Authors of books titled The Meaning of Life or similar include: Alfred Ayer (The Meaning of Life and Other Essays) Bradley Trevor Greive (The Meaning of Life) C.E.M.
90	How many people are in the world	The total number of 1 million people.
91	How are we here	We are here.
92	When will the world end	NO one knows when the world will end.
93	Hello is a invented word but what does it mean	Hello is a greeting in the English language.
94	How did the Earth start	Earth formed about 4.54 billion years ago by accretion from the solar nebula, a disk-shaped mass of dust and gas left over from the formation of the Sun, which also created the rest of the Solar System.
95	How many hours are there on Saturn	It takes 3.6 Earth years to complete an orbit around Saturn.
96	How long has the earth existed.	It was present from ~11.1 to 5.3 Ma, existing for approximately 6.2 million years.
97	How big is the sun	It has around 3.5 times the size of the Sun with five times the Sun's radius.
98	When Is the world going to end	NO one knows when the world will end.
99	What are the names of the dwarf planets.	For a list of named planets, dwarf planets and stars, see: List of gravitationally rounded objects in the Solar System For a list of named minor planets see: List of named minor planets.
100	How old can you live to	Maximum known age is eight years.
101	Meaning of life	Meaning of life generally refers to the possible purpose and significance that may be attributed to human existence and one's personal life.

	Question	**Answer**
102	How do batteries work	most automotive batteries) are designed to deliver short, high-current bursts for cranking the engine, thus frequently discharging only a small part of their capacity.
103	How many people are there in the world	The total number of 1 million people.
104	How old is the earth	It is estimated to be 200 million years old, and measures 35 kilometres (22 mi) in diameter.
105	What happened before the Big Bang	It started approximately 10'6 seconds after the Big Bang, when the temperature of the universe had fallen sufficiently to allow the quarks from the preceding quark epoch to bind together into hadrons.
106	How do birds fly	Flight is direct with rapid and buzzy wingbeats.
107	Are you dumb	You Must Be Joking!
108	How much does the world weigh	It weighs a little less than 15 million pounds.
109	Potatoes or Potatoes	potato skins).
110	How do the northern lights occur	North light is natural light coming from the north (in the Northern Hemisphere).
111	Who invented PCs	It was developed by Chips and Technologies.
112	When was the earth made	It is believed to have formed between 7,300 and 25,000 years ago.
113	Where is the earth	It lies in the southern hemisphere, to the south-southeast of the walled plain Jules Verne.
114	What was the first planet to be discovered.	It was the first planet discovered orbiting a giant star.
115	What is a planet	A planet, in astronomy, is one of a class of celestial bodies that orbit stars.
116	How many stars in the solar system	Of the approximately 6,000 stars visible to the naked eye under optimal conditions, the selected stars are among the brightest and span thirty-eight constellations of the celestial sphere from the declination of 70°south to 89°north.
117	What will happen to the sun in 100,000,000 years	During the next four billion years, the luminosity of the Sun will steadily increase, resulting in a rise in the solar radiation reaching the Earth.
118	What is liquid?	Liquid is a phase of matter.
119	What comes first chicken or egg?	Egg!
120	Whats your question?	And .
121	Will we ever be able to communicate with other planets?	Some day I hope we make contact with beings from other planets.
122	What are planets	Planets is a collection of compositions representing the solar system and beyond.
123	Can you explain the offside rule?	It is similar to offside except when it occurs, the play is not allowed to begin.

	Question	**Answer**
124	What is the meaning of life has it got anything to do with monty python?	(The) Meaning of Life may also refer to: Monty Python's The Meaning of Life, a 1983 film by Monty Python Monty Python's The Meaning of Life (album), album by Monty Python in conjunction with film of same name, also contains a song titled "The Meaning of Life" The Meaning of Life (TV series), Irish television series presented by Gay Byrne The Meaning of Life (animated film), animated short film written and directed by Don Hertzfeldt The Meaning of Life (Tankard album), album by Tankard, also contains a song by the same name "The Meaning of Life" (The Offspring song), song by The Offspring from the 1997 album Ixnay on the Hombre M.O.L., acronym for "Meaning of Life", a DVD by Disturbed "Meaning of Life", a song by Disturbed from The Sickness Meaning of Life (album), by Kelly Clarkson Authors of books titled The Meaning of Life or similar include: Alfred Ayer (The Meaning of Life and Other Essays) Bradley Trevor Greive (The Meaning of Life) C.E.M.
125	What does curious mean?	Being curious means being inquisitive and tending to investigate or explore, in the passive sense as strange, surprising, odd, or as a euphemism for erotic as in 'curious art'.
126	What is a blackhole?	A black hole is a region of spacetime exhibiting such strong gravitational effects that nothing,Äö not even particles and electromagnetic radiation such as light,Äö can escape from inside it.
127	What is a rainbow?	A rainbow is a meteorological phenomenon that appears as a multicolored arc that forms with the sunlight.
128	How is paper made?	Many kinds of paper are made from wood with nothing else mixed into them.
129	How many stars are there	It consists of about 20 stars.
130	What really is deja vu?	Déjà vu from French, literally "already seen", is the phenomenon of having the feeling that the situation currently being experienced has already been experienced in the past.
131	What is blue	It is the blue often associated with blue jeans.
132	How are you doing	How are you!
133	How many people live in Bristol	It has a population of approximately 22,636, stretching from Whitecraigs to Mearnskirk.
134	Love you	Love Me!
135	Ho ho ho	H≈ç-H≈ç!

	Question	**Answer**
136	How was the moon formed	The Moon is thought to have formed about 4.51 billion years ago, not long after Earth.
137	How are black holes formed	A stellar black hole (or stellar-mass black hole) is a black hole formed by the gravitational collapse of a massive star.
138	What colour is the sky.	It is the color of a cloud-covered sky, of ash and of lead.
139	What is the meaning of life	(The) Meaning of Life may also refer to: Monty Python's The Meaning of Life, a 1983 film by Monty Python Monty Python's The Meaning of Life (album), album by Monty Python in conjunction with film of same name, also contains a song titled "The Meaning of Life" The Meaning of Life (TV series), Irish television series presented by Gay Byrne The Meaning of Life (animated film), animated short film written and directed by Don Hertzfeldt The Meaning of Life (Tankard album), album by Tankard, also contains a song by the same name "The Meaning of Life" (The Offspring song), song by The Offspring from the 1997 album Ixnay on the Hombre M.O.L., acronym for "Meaning of Life", a DVD by Disturbed "Meaning of Life", a song by Disturbed from The Sickness Meaning of Life (album), by Kelly Clarkson Authors of books titled The Meaning of Life or similar include: Alfred Ayer (The Meaning of Life and Other Essays) Bradley Trevor Greive (The Meaning of Life) C.E.M.
140	to shape our future decisions	to make informed decisions.
141	What is the universe	The Universe is all of space and time (spacetime) and its contents, which includes planets, moons, minor planets, stars, galaxies, the contents of intergalactic space and all matter and energy.
142	How does a plane work	Takeoff is the phase of flight in which an aerospace vehicle or animal goes from the ground to flying in the air.
143	How long is the River Avon	It is a tributary of the Bristol Avon and is some 12 miles (19 km) long.
144	What time is it on the moon	Current local time is 22:04; the sun rises at 08:57 and sets at 21:04 local time (Asia/Chongqing UTC+8).
145	How many people our on earth	3300 people.
146	When is the world going to end.	NO one knows when the world will end.
147	Why are you asking me this.	I can't answer that question.
148	Who is the creator of museum	It was founded in 1961 by the Danish artist Asger Jorn, Peter Glob and Werner Jacobsen from the National Museum of Denmark and Holger Arbman of the University of Lund, Sweden.

	Question	**Answer**
149	How many moons on Venus	Jupiter has at least 69 moons, including the four large Galilean moons discovered by Galileo Galilei in 1610.
150	How big is the universe	is hundreds of times larger than the material universe .
151	Type your question here	Here what I'm saying.
152	Koechi la nie	Nyan Koi!
153	I love this	Dig This!
154	How big is the universe?	is hundreds of times larger than the material universe .
155	Why do certain people like certain types of music?	We like music that transcends genre.
156	How do batteries run themselves	most automotive batteries) are designed to deliver short, high-current bursts for cranking the engine, thus frequently discharging only a small part of their capacity.
157	How does science work	Science is a body of empirical, theoretical, and practical knowledge about the natural world, produced by scientists who emphasize the observation, explanation, and prediction of real world phenomena.
158	Why do we have a solar eclipse	A solar eclipse is an astronomical phenomenon that occurs when the Moon passes between Earth and the Sun, thereby totally or partly obscuring the image of the Sun for a viewer on Earth.
159	How does an animal talk	It communicates by using a variety of vocalizations and physical interactions.
160	States of matter	In physics, a state of matter is one of the distinct forms in which matter can exist.
161	What age is the earth	It is 9 km in diameter and the age is estimated to be 46 $\pm$ 3 million years old (Eocene).
162	Why does the earth rotate	The Earth's orbital motions (inclination of the earth's axis on its orbit with respect to the sun, gyroscopic precession of the earth's axis every 26,000 years; free precession every 440 days, precession of earth orbit and orbital variations such as perihelion precession every 19,000 and 23,000 years) leave traces visible in the geological record.

	Question	**Answer**
163	Why does the earth rotate	The Earth's orbital motions (inclination of the earth's axis on its orbit with respect to the sun, gyroscopic precession of the earth's axis every 26,000 years; free precession every 440 days, precession of earth orbit and orbital variations such as perihelion precession every 19,000 and 23,000 years) leave traces visible in the geological record.
164	How did the Big Bang happen.	It was originally suggested as a phase of the cyclic model or oscillatory universe interpretation of the Big Bang, where the first cosmological event was the result of the collapse of a previous universe.
165	Why is it so expensive	It's expensive.
166	D viens ti	Je suis venu calme orphelin.
167	How old is space	It is approximately a billion years old and is located about 5,400 light years (ly) from the Solar System and 1,100 ly above the plane of the Milky Way galaxy.
168	How many galaxies are there	It contains 29,418 galaxies and 9,134 galaxy clusters.
169	What is gravity	Gravity is defined as the resultant of gravitation and the centrifugal force caused by the Earth's rotation.
170	Who is olga buzova	Milka Tcherneva Bliznakov (1927-2010) was a Bulgarian architect and architectural historian.
171	Who invented earth	It was discovered by Rob.
172	How heavy is the earth	It weighs a little less than 15 million pounds.
173	Can we fly..	People Can Fly Sp.
174	Who made the first computer	It was originally developed at Bell Labs in 1972 by Marc Rochkind for an IBM System/370 computer running OS/360.
175	What made the Big Bang	It was originally suggested as a phase of the cyclic model or oscillatory universe interpretation of the Big Bang, where the first cosmological event was the result of the collapse of a previous universe.
176	What are black holes	A stellar black hole (or stellar-mass black hole) is a black hole formed by the gravitational collapse of a massive star.
177	How old is the universe.	Detailed measurements of the expansion rate of the universe place the Big Bang at around 13.8 billion years ago, which is thus considered the age of the universe.
178	When were dinosaurs made	It was described by Johnson and Coucot in 1967, and existed from the Silurian to the Devonian of Australia, Austria, Canada, China, the Czech Republic, Italy, Tajikistan, and the United States.
179	Will there be	They said 'No, no, there won't'.

	Question	Answer
180	how was earth made	Earth formed about 4.54 billion years ago by accretion from the solar nebula, a disk-shaped mass of dust and gas left over from the formation of the Sun, which also created the rest of the Solar System.
181	How does the moon provide gravity	There is a mutual attraction between the Moon and the solid Earth, which can be considered to act on its centre of mass.
182	Why is there gravity in the world	Gravity is a fundamental force on earth that influences all biological systems at a molecular level.
183	What is death	Death is the termination of the biological functions that sustain a living organism.
184	What is this block called	The Block may refer to:
185	Tyugxdffdglzlp h cvbhjjjy	Wq.
186	How does Mould form	The process is essentially that of a mould.
187	How does electricity work	Electricity is a type of energy caused by the presence and flow of electric charge.
188	How old is earth	It is approximately a billion years old and is located about 5,400 light years (ly) from the Solar System and 1,100 ly above the plane of the Milky Way galaxy.
189	How does a fan work	A fan is a machine used to create flow within a fluid, typically a gas such as air.
190	How big is the galaxy	It is one of the largest structures found in the universe; covering about 25x20 degrees of the sky.
191	What is the everage age of orangutans	The maximum age reported for this species is 4 years.
192	Yann or Jade	Jade.
193	Ferrari or Lamborghini.	BMW.
194	Lamborghini or Ferrari	BMW.
195	Who discovered gravity	It was discovered by Schelte J.
196	How big is the earth	It is 450 m long and 150 m wide and has a surface area of 45,5 decares.
197	How high is the sky	440 above the sea level.
198	Who won the premier league	It was won by Geylang United, which was their second league title.
199	How old is lil pump	He was born in 1977 and died in June 1995 at the age of 18.
200	What did they do	This they did.
201	Why is the sky blue	The same elastic scattering processes cause the sky to be blue.
202	How many stars are in the galaxy	The most voluminous modern catalogues list on the order of a billion stars, out of an estimated total of 200 to 400 billion in the Milky Way.
203	Where did the dinosaurs die	Its remains have been found in Africa, Asia, and Europe.

	Question	**Answer**
204	When was at Bristol made	Marlborough) was a British car manufactured in a garage at Brooklands, Weybridge, Surrey between 1923 and 1924.
205	How old is the universe	Detailed measurements of the expansion rate of the universe place the Big Bang at around 13.8 billion years ago, which is thus considered the age of the universe.
206	What is the cube	The Professor's Cube is a mechanical puzzle, a 5x5x5 version of the Rubik's Cube.
207	What is mud?	Mud is a liquid or semi-liquid mixture of water and soil.
208	What is cheese	Cheese is a milk-based food that is produced in wide-ranging flavors, textures, and forms.
209	How many people are on the Earth	It contains 18,365 people.
210	How far is the closest black hole	It lies at a distance of about 50 kiloparsecs (163,000 light-years) in a neighbouring galaxy known as the Large Magellanic Cloud.
211	How old is the earth	It is estimated to be 200 million years old, and measures 35 kilometres (22 mi) in diameter.
212	Daniela and Marisol	Maria and Ian Veneracion.
213	How many hours are there in a year in	It has a length of 365 days in an ordinary year, with 8,760 hours, 525,600 minutes, and 31,536,000 seconds; but 366 days in a leap year, with 8,784 hours, 527,040 minutes, and 31,622,400 seconds.
214	What is g forces	The g-force (with g from gravitational) is a measurement of the type of acceleration that causes a perception of weight.
215	What was the first planet created in the solar system	It was the first planet discovered orbiting a giant star.
216	What is the weight of the earth.	In other words, an object with a mass of 1.0 kilogram will weigh approximately 9.81 newtons on the surface of the Earth (its mass multiplied by the gravitational field strength), since the newton is a unit of force, while the kilogram is a unit of mass.
217	What is dark fluid.	It is a viscous transparent liquid at room temperature.
218	How dense are black holes	This makes it one of the most massive black holes known in the universe, 5,000 times the mass of the black hole at the center of the Milky Way.
219	How did the Big Bang happen	It was originally suggested as a phase of the cyclic model or oscillatory universe interpretation of the Big Bang, where the first cosmological event was the result of the collapse of a previous universe.

	Question	**Answer**
220	What is the meaning of life the universe and everything	The meaning of life as we perceive it is derived from philosophical and religious contemplation of, and scientific inquiries about existence, social ties, consciousness, and happiness.
221	What's the meaning of life	(The) Meaning of Life may also refer to: Monty Python's The Meaning of Life, a 1983 film by Monty Python Monty Python's The Meaning of Life (album), album by Monty Python in conjunction with film of same name, also contains a song titled "The Meaning of Life" The Meaning of Life (TV series), Irish television series presented by Gay Byrne The Meaning of Life (animated film), animated short film written and directed by Don Hertzfeldt The Meaning of Life (Tankard album), album by Tankard, also contains a song by the same name "The Meaning of Life" (The Offspring song), song by The Offspring from the 1997 album Ixnay on the Hombre M.O.L., acronym for "Meaning of Life", a DVD by Disturbed "Meaning of Life", a song by Disturbed from The Sickness Meaning of Life (album), by Kelly Clarkson Authors of books titled The Meaning of Life or similar include: Alfred Ayer (The Meaning of Life and Other Essays) Bradley Trevor Greive (The Meaning of Life) C.E.M.
222	What is string theory	String theory is a branch of theoretical physics String theory may also refer to: concatenation theory, a topic in symbolic logic dealing with strings of characters String Theory (band), an American electronic music band String Theory (novels), a trilogy of novels set in the Star Trek universe "String Theory" (Heroes), retitled "Five Years Gone", an episode of the TV series Heroes "String Theory" (The Shield), an episode of the TV series The Shield String Theory, a webcomic graphic novel based on the TV series Heroes
223	How big is the biggest penis	It is commonly cited as having an average penis length of 2.4 metres (7 ft 10 in) to 3 metres (9.8 ft) and a diameter of 30 centimetres (12 in) to 36 centimetres (14 in).
224	Vous avez bien aimer ?	Tout De M,àö,Ñ¢me, Je Vous Aime!
225	How do you make bubbles	Bubbles on top of the water, less ambiguously known as a foam bath (see photo), can be obtained by adding a product containing foaming surfactants to water and temporarily aerating it by agitation (often merely by the fall of water filling the tub).

	Question	Answer
226	What is the aim of our life	We make things with purpose, to make life feel more comfortable, more possible, and more luxurious.
227	Who wants to know	I Should Know."
228	Who is god	The most powerful god around the world with immediate effect.
229	Jordan was the	Jordan.
230	What is the density of a black hole.	It contains five times the density compared to the standard cosmological density of the universe.
231	When was the earth created	It is approximately 4,500 million years old and was formed at the same time as the Earth and the solar system.
232	How many dinosaurs were there	In total, their large efforts resulted in when 142 species of dinosaurs being discovered.
233	What day is it on the moon	It is generally celebrated on 13 April or 14 April and traditionally begins at the sighting of the new moon.
234	How long were the dinosaurs alive	It lasted from 497 to 485.4 million years ago.
235	How does energy work	Common forms of energy include the kinetic energy of a moving object, the potential energy stored by an object's position in a force field (gravitational, electric or magnetic), the elastic energy stored by stretching solid objects, the chemical energy released when a fuel burns, the radiant energy carried by light, and the thermal energy due to an object's temperature.
236	How does energy work	Common forms of energy include the kinetic energy of a moving object, the potential energy stored by an object's position in a force field (gravitational, electric or magnetic), the elastic energy stored by stretching solid objects, the chemical energy released when a fuel burns, the radiant energy carried by light, and the thermal energy due to an object's temperature.
237	Who invented gravity	It was discovered and described by J.T.
238	What do you mean	what.
239	If each year starts and ends on midnight, why is there 365 and 1/4 days in a year	It occurs because leap years occur every 4 years and there are 7 possible days to start a leap year, making a 28-year sequence.
240	What year did the dinosaurs extinct	They first appeared about 400 million years ago in the Devonian, and became extinct just before the Mesozoic, about 225 million years ago.
241	Why was this place renamed	gave rise to the name of the place.
242	How long is fence	The fence is usually 1.5-2 metres tall.
243	How long does it last for	which sometimes lasts for hours.
244	Why is light considered as the colour white	The light used falls under the visible blue light spectrum.

	Question	**Answer**
245	What is the biggest planet in the universe	In the Solar System, it is the fourth-largest planet by diameter, the third-most-massive planet, and the densest giant planet.
246	Who is Thomas Crestey	Morgan Bulkeley Brainard (January 8, 1879 - August 28, 1957) was an American attorney, insurance executive, and book collector.
247	What is the approximate age of a human to live up to	The maximum known age is 5 years for males and 12 years for females.
248	How tall is the tallest skyscraper in t	It is 5561 feet high.
249	How do rockets really work	Rocket engines work by action and reaction and push rockets forward simply by expelling their exhaust in the opposite direction at high speed, and can therefore work in the vacuum of space.
250	How old is she	She is 23 years old.
251	Hi will your ol	Hey There!
252	When was the earth formed?	According to radiometric dating and other sources of evidence, Earth formed over 4 billion years ago.
253	Whats global warming historic and projected timelines for global effects?	Global Warming: What You Need to Know is a 2006 global warming (climate change) documentary, directed by Nicolas Brown, starring Tom Brokaw, James Hansen, Michael Oppenheimer, and Mark Serreze.
254	How many creatures in the mariahd trench	Over 4,000 living species are known.
255	How are clouds formed?	On Earth, clouds are formed as a result of saturation of the air when it is cooled to its dew point, or when it gains sufficient moisture (usually in the form of water vapor) from an adjacent source to raise the dew point to the ambient temperature.
256	Who scored the goal?	Kopeykin scored the goal.
257	What is poop?	Poop or pooped may refer to: Feces, waste product from an animal's digestive tract expelled through the anus.
258	Does everyone see the same colours?	Not everyone sees the same colors.
259	Why does icecream melt?	It melts more quickly than solid ice.
260	How many people are there	There are currently 168 people listed.
261	How do clouds form?	On Earth, clouds are formed as a result of saturation of the air when it is cooled to its dew point, or when it gains sufficient moisture (usually in the form of water vapor) from an adjacent source to raise the dew point to the ambient temperature.
262	What is the longest word?	The longest word found in a major dictionary is "pneumonoultramicroscopicsilicovolcanoconiosis", but this is a technical term that was coined specifically to be the longest word.
263	What is it made from?	It is usually made from flint or chert.
264	Email to contact you	Addresses E-mail: trueman1215@msn.com.

	Question	**Answer**
265	DRAGON, SO THEY CAN ROAR	Drag‚àö‚â•n.
266	What is dirt made of?	Dirt!
267	What is lightening	Radioluminescence is the phenomenon by which light is produced in a material by bombardment with ionizing radiation such as alpha particles, beta particles, or gamma rays.
268	What is noise	Noise is a variety of sound, usually meaning any unwanted sound.
269	How can the universe be infinite	It is still undetermined whether the Universe is infinite.
270	What is the fastest plane in the world	When it was first introduced, some aviation circles claimed that at 615 mph (990 km/h), it was the fastest jet transport in the world.
271	How is a black hole formed	A stellar black hole (or stellar-mass black hole) is a black hole formed by the gravitational collapse of a massive star.
272	Development of education from space in universities and schools For a better understanding of the broader world of students now And raising the peculiarity and kindness in the small community we live in.	The opportunity for Learners to make a difference now.
273	Cuz I'm me	Because I do.
274	how big is the universe	is hundreds of times larger than the material universe .
275	A person who seems very happy	One who is friendly and attracts friendship from near and far
276	What is deja vu?	Déjà vu from French, literally "already seen", is the phenomenon of having the feeling that the situation currently being experienced has already been experienced in the past.
277	Whats a atom	An atom is the smallest constituent unit of ordinary matter that has the properties of a chemical element.
278	Are you serious?	You Cannot Be Serious!
279	Who discovered pluto	It was discovered by a team of astronomers from the Institute for Astronomy of the University of Hawaii led by David Jewitt and Scott S. Sheppard and Jan Kleyna in 2001, and given the temporary designation S/2001 J 3.
280	Who discovered electronic devices	It was invented by American physicist Robert J.
281	What is antimatter and what is in the space between atoms?	In modern physics, antimatter is defined as a material composed of the antiparticle (or "partners") to the corresponding particles of ordinary matter.

	Question	Answer
282	How did the big bang theory happen	It was originally suggested as a phase of the cyclic model or oscillatory universe interpretation of the Big Bang, where the first cosmological event was the result of the collapse of a previous universe.
283	How many books are on the planet	So far, 26 books have been released.
284	is the earth flat or round	Flat Earth may refer to:
285	how many galaxies are there in the universe	Recent estimates of the number of galaxies in the observable universe range from 200 billion ($2x10^{11}$) to 2 trillion ($2x10^{12}$) or more, containing more stars than all the grains of sand on planet Earth.
286	How are rivers formed	Normally, water that has accrued in a drainage basin eventually flows out through rivers or streams on the Earth's surface or by underground diffusion through permeable rock, ultimately ending up in the oceans.
287	time dilation do to relativity	For the concept in physics, see time dilation.
288	How far is the moon	It is currently in a polar orbit around Mars with an altitude of about 3,800 km or 2,400 miles.
289	How does the world rotate	The Earth's orbital motions (inclination of the earth's axis on its orbit with respect to the sun, gyroscopic precession of the earth's axis every 26,000 years; free precession every 440 days, precession of earth orbit and orbital variations such as perihelion precession every 19,000 and 23,000 years) leave traces visible in the geological record.
290	How is chocolate made	It is usually made from dairy products, such as milk and cream, and often combined with fruits or other ingredients and flavors.
291	Magic is simply someththing that the observer doesn't understand.	Magic is broken into two types.
292	How old is the oldest tree.	It is estimated to be between 600 and 1,000 years old, and is possibly the oldest tree in Germany.
293	Was Mars originally like Earth	It has long been hypothesized that life may have existed on Mars due to the similarity of environmental and tectonic conditions during the Archean time.
294	Mountains , hills , forests	Woods - natural forest with walking paths.
295	thank you for your	Thank you!
296	what is the training process?	Training is done in two stages.

Bibliography

[1] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg, *Fine-grained analysis of sentence embeddings using auxiliary prediction tasks*, arXiv preprint arXiv:1608.04207, (2016).

[2] E. Agirre, C. Banea, C. Cardie, D. M. Cer, M. T. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, *Semeval-2014 task 10: Multilingual semantic textual similarity.*, in SemEval@ COLING, 2014, pp. 81–91.

[3] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez Agirre, R. Mihalcea, G. Rigau Claramunt, and J. Wiebe, *Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation*, in SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511., ACL (Association for Computational Linguistics), 2016.

[4] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, *Semeval-2012 task 6: A pilot on semantic textual similarity*, in * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), 2012, pp. 385–393.

[5] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, ** sem 2013 shared task: Semantic textual similarity*, in Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity, 2013, pp. 32–43.

[6] J. Andreas, *Measuring compositionality in representation learning*, arXiv preprint arXiv:1902.07181, (2019).

[7] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, *Invariant risk minimization*, arXiv preprint arXiv:1907.02893, (2019).

[8] S. Arora, Y. Liang, and T. Ma, *A simple but tough-to-beat baseline for sentence embeddings*, (2016).

[9] M. Artetxe and H. Schwenk, *Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond*, arXiv preprint arXiv:1812.10464, (2018).

[10] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, (2014).

[11] Y. Bao, H. Zhou, S. Huang, L. Li, L. Mou, O. Vechtomova, X. Dai, and J. Chen, *Generating sentences from disentangled syntactic and semantic spaces*, arXiv preprint arXiv:1907.05789, (2019).

[12] P. Baudiš and J. Šedivỳ, *Modeling of the question answering task in the yodaqa system*, in International Conference of the cross-language evaluation Forum for European languages, Springer, 2015, pp. 222–228.

[13] M. Belkin, D. Hsu, S. Ma, and S. Mandal, *Reconciling modern machine-learning practice and the classical bias-variance trade-off*, Proceedings of the National Academy of Sciences, 116 (2019), pp. 15849–15854.

[14] Y. Bengio, A. Courville, and P. Vincent, *Representation learning: A review and new perspectives*, IEEE transactions on pattern analysis and machine intelligence, 35 (2013), pp. 1798–1828.

[15] Y. Bengio, R. Ducharme, and P. Vincent, *A neural probabilistic language model*, Advances in neural information processing systems, 13 (2000).

[16] Y. Bengio, P. Simard, P. Frasconi, et al., *Learning long-term dependencies with gradient descent is difficult*, IEEE transactions on neural networks, 5 (1994), pp. 157–166.

[17] H. Bhathena, A. Willis, and N. Dass, *Evaluating compositionality of sentence representation models*, in Proceedings of the 5th Workshop on Representation Learning for NLP, 2020, pp. 185–193.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent dirichlet allocation*, Journal of machine Learning research, 3 (2003), pp. 993–1022.

[19] B. S. Bloom, *Taxonomy of educational objectives: The classification of educational goals*, Cognitive domain, (1956).

[20] D. G. Bobrow, *Natural language input for a computer problem solving system*, (1964).

[21] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching word vectors with subword information*, arXiv preprint arXiv:1607.04606, (2016).

[22] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, *A large annotated corpus for learning natural language inference*, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 632–642.

[23] E. Brill, S. Dumais, and M. Banko, *An analysis of the askmsr question-answering system*, in Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), 2002, pp. 257–264.

[24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., *Language models are few-shot learners*, Advances in neural information processing systems, 33 (2020), pp. 1877–1901.

[25] R. Caruana, *Multitask learning*, Machine learning, 28 (1997), pp. 41–75.

[26] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, *Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation*, in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 1–14.

[27] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al., *Universal sentence encoder*, arXiv preprint arXiv:1803.11175, (2018).

[28] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, *Universal sentence encoder for English*, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, Nov. 2018, Association for Computational Linguistics, pp. 169–174.

[29] D. Chandrasekaran and V. Mago, *Evolution of semantic similarity,Äîa survey*, ACM Computing Surveys (CSUR), 54 (2021), pp. 1–37.

[30] D. Chen, A. Fisch, J. Weston, and A. Bordes, *Reading wikipedia to answer open-domain questions*, arXiv preprint arXiv:1704.00051, (2017).

[31] C. Chin and D. E. Brown, *Learning in science: A comparison of deep and surface approaches*, Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching, 37 (2000), pp. 109–138.

[32] C. Chin and J. Osborne, *Students' questions: a potential resource for teaching and learning science*, Studies in science education, 44 (2008), pp. 1–39.

[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, arXiv preprint arXiv:1412.3555, (2014).

[34] A. CLAUSET, M. E. NEWMAN, AND C. MOORE, *Finding community structure in very large networks*, Physical review E, 70 (2004), p. 066111.

[35] R. COLLOBERT, J. WESTON, L. BOTTOU, M. KARLEN, K. KAVUKCUOGLU, AND P. KUKSA, *Natural language processing (almost) from scratch*, Journal of machine learning research, 12 (2011), pp. 2493–2537.

[36] A. CONNEAU, D. KIELA, H. SCHWENK, L. BARRAULT, AND A. BORDES, *Supervised learning of universal sentence representations from natural language inference data*, arXiv preprint arXiv:1705.02364, (2017).

[37] A. CONNEAU, G. KRUSZEWSKI, G. LAMPLE, L. BARRAULT, AND M. BARONI, *What you can cram into a single vector: Probing sentence embeddings for linguistic properties*, arXiv preprint arXiv:1805.01070, (2018).

[38] K. CSERNAI, *Quora question pairs*, 2017.

[39] S. CUCCIO-SCHIRRIPA AND H. E. STEINER, *Enhancement and analysis of science question level for middle school students*, Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching, 37 (2000), pp. 210–224.

[40] A. M. DAI AND Q. V. LE, *Semi-supervised sequence learning*, in Advances in neural information processing systems, 2015, pp. 3079–3087.

[41] I. DASGUPTA, D. GUO, A. STUHLMÜLLER, S. J. GERSHMAN, AND N. D. GOODMAN, *Evaluating compositionality in sentence embeddings*, arXiv preprint arXiv:1802.04302, (2018).

[42] M. DEHGHANI, S. GOUWS, O. VINYALS, J. USZKOREIT, AND Ł. KAISER, *Universal transformers*, arXiv preprint arXiv:1807.03819, (2018).

[43] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2018.
arXiv preprint arXiv:1810.04805.

[44] Y. J. DORI AND O. HERSCOVITZ, *Question-posing capability as an alternative evaluation method: Analysis of an environmental case study*, Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching, 36 (1999), pp. 411–430.

[45] M. DUNN, L. SAGUN, M. HIGGINS, V. U. GUNEY, V. CIRIK, AND K. CHO, *Searchqa: A new q&a dataset augmented with context from a search engine*, arXiv preprint arXiv:1704.05179, (2017).

[46] M. EBERSBACH, M. FEIERABEND, AND K. B. B. NAZARI, *Comparing the effects of generating questions, testing, and restudying on students' long-term recall in university learning*, Applied Cognitive Psychology, 34 (2020), pp. 724–736.

[47] J. ELSTGEEST, *The right question at the right time*, Primary science: Taking the plunge, (1985), pp. 36–46.

[48] D. ERHAN, A. COURVILLE, Y. BENGIO, AND P. VINCENT, *Why does unsupervised pre-training help deep learning?*, in Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[49] A. ETTINGER, A. ELGOHARY, C. PHILLIPS, AND P. RESNIK, *Assessing composition in sentence vector representations*, in Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1790–1801.

[50] A. ETTINGER, A. ELGOHARY, AND P. RESNIK, *Probing for semantic evidence of composition by means of simple classification tasks*, in Proceedings of the 1st workshop on evaluating vector-space representations for nlp, 2016, pp. 134–139.

[51] R. S. J. J. Y. FAN, T. H. C. T.-S. CHUA, AND M.-Y. KAN, *Using syntactic and semantic relation analysis in question answering*, in Proceedings of the 14th Text REtrieval Conference (TREC), Gaithersburg, MD, USA, 2005, pp. 15–18.

[52] D. A. FERRUCCI, *Introduction to ‚Äúthis is watson‚Äù*, IBM Journal of Research and Development, 56 (2012), pp. 1–1.

[53] J. A. FODOR AND E. LEPORE, *The compositionality papers*, Oxford University Press, 2002.

[54] I. GOODFELLOW, Y. BENGIO, AND A. COURVILLE, *Deep Learning*, MIT Press, 2016. `http://www.deeplearningbook.org`.

[55] B. F. GREEN JR, A. K. WOLF, C. CHOMSKY, AND K. LAUGHERY, *Baseball: an automatic question-answerer*, in Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference, 1961, pp. 219–224.

[56] D. GUTHRIE, B. ALLISON, W. LIU, L. GUTHRIE, AND Y. WILKS, *A closer look at skip-gram modelling.*, in LREC, 2006, pp. 1222–1225.

[57] K. GUU, K. LEE, Z. TUNG, P. PASUPAT, AND M.-W. CHANG, *Realm: Retrieval-augmented language model pre-training*, arXiv preprint arXiv:2002.08909, (2020).

[58] R. HADSELL, S. CHOPRA, AND Y. LECUN, *Dimensionality reduction by learning an invariant mapping*, in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, IEEE, 2006, pp. 1735–1742.

[59] A. Hagberg, P. Swart, and D. S Chult, *Exploring network structure, dynamics, and function using networkx*, tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[60] M. Hajjem and C. Latiri, *Combining ir and lda topic modeling for filtering microblogs*, Procedia Computer Science, 112 (2017), pp. 761–770.

[61] S. M. Harabagiu, D. I. Moldovan, C. Clark, M. Bowden, J. Williams, and J. Bensley, *Answer mining by combining extraction techniques with abductive reasoning.*, in TREC, 2003, pp. 375–382.

[62] Z. S. Harris, *Distributional structure*, Word, 10 (1954), pp. 146–162.

[63] M. Henderson, R. Al-Rfou, B. Strope, Y.-H. Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil, *Efficient natural language response suggestion for smart reply*, arXiv preprint arXiv:1705.00652, (2017).

[64] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, arXiv preprint arXiv:1503.02531, (2015).

[65] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[66] M. Hu and B. Liu, *Mining and summarizing customer reviews*, in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 168–177.

[67] R. Huddleston, R. Huddleston, G. Pullum, P. K, L. Bauer, B. Birner, T. Briscoe, P. Collins, D. Denison, D. Lee, et al., *The Cambridge Grammar of the English Language*, The Cambridge Grammar of the English Language, Cambridge University Press, 2002.

[68] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, *On using very large target vocabulary for neural machine translation*, arXiv preprint arXiv:1412.2007, (2014).

[69] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*, in European conference on machine learning, Springer, 1998, pp. 137–142.

[70] J. Johnson, M. Douze, and H. Jégou, *Billion-scale similarity search with gpus*, IEEE Transactions on Big Data, (2019).

[71] K. S. Jones, *A statistical interpretation of term specificity and its application in retrieval*, Journal of documentation, (1972).

[72] M. JOSHI, E. CHOI, D. S. WELD, AND L. ZETTLEMOYER, *Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension*, arXiv preprint arXiv:1705.03551, (2017).

[73] D. JURAFSKY, *Speech & language processing*, Pearson Education India, 2000.

[74] D. JURAFSKY AND J. H. MARTIN, *Speech and language processing. vol. 3*, US: Prentice Hall, (2014).

[75] V. KARPUKHIN, B. OĞUZ, S. MIN, P. LEWIS, L. WU, S. EDUNOV, D. CHEN, AND W.-T. YIH, *Dense passage retrieval for open-domain question answering*, arXiv preprint arXiv:2004.04906, (2020).

[76] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, 2015.
In ICLR.

[77] R. KIROS, Y. ZHU, R. R. SALAKHUTDINOV, R. ZEMEL, R. URTASUN, A. TORRALBA, AND S. FIDLER, *Skip-thought vectors*, in Advances in neural information processing systems, 2015, pp. 3294–3302.

[78] R. A. KIRSCH, *Computer interpretation of english text and picture patterns*, IEEE Transactions on Electronic Computers, (1964), pp. 363–376.

[79] A. KRATZER AND I. HEIM, *Semantics in generative grammar*, vol. 1185, Blackwell Oxford, 1998.

[80] J. KUPIEC, *Murax: A robust linguistic approach for question answering using an on-line encyclopedia*, in Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, 1993, pp. 181–190.

[81] C. C. KWOK, O. ETZIONI, AND D. S. WELD, *Scaling question answering to the web*, in Proceedings of the 10th international conference on World Wide Web, 2001, pp. 150–161.

[82] Z. LAN, M. CHEN, S. GOODMAN, K. GIMPEL, P. SHARMA, AND R. SORICUT, *Albert: A lite bert for self-supervised learning of language representations*, arXiv preprint arXiv:1909.11942, (2019).

[83] J. LEE, M. SEO, H. HAJISHIRZI, AND J. KANG, *Contextualized sparse representations for real-time open-domain question answering*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 912–919.

[84] J. LEE, W. YOON, S. KIM, D. KIM, S. KIM, C. H. SO, AND J. KANG, *Biobert: a pre-trained biomedical language representation model for biomedical text mining*, Bioinformatics, 36 (2020), pp. 1234–1240.

[85] K. LEE, M.-W. CHANG, AND K. TOUTANOVA, *Latent retrieval for weakly supervised open domain question answering*, arXiv preprint arXiv:1906.00300, (2019).

[86] S. LEE, S. CHO, AND S. IM, *Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 15252–15261.

[87] W. G. LEHNERT, *A conceptual theory of question answering*, in Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1, 1977, pp. 158–164.

[88] R. LI, X. ZHAO, AND M.-F. MOENS, *A brief overview of universal sentence representation methods: A linguistic view*, ACM Computing Surveys (CSUR), 55 (2022), pp. 1–42.

[89] X. LI AND D. ROTH, *Learning question classifiers*, in COLING 2002: The 19th International Conference on Computational Linguistics, 2002.

[90] Y. LIN, H. JI, Z. LIU, AND M. SUN, *Denoising distantly supervised open-domain question answering*, in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1736–1745.

[91] B. LIQUET, S. MOKA, AND Y. NAZARATHY, *The mathematical engineering of deep learning*, 2022.

[92] X. LIU, P. HE, W. CHEN, AND J. GAO, *Multi-task deep neural networks for natural language understanding*, 2019.
In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4487-4496.

[93] Y. LIU, M. OTT, N. GOYAL, J. DU, M. JOSHI, D. CHEN, O. LEVY, M. LEWIS, L. ZETTLEMOYER, AND V. STOYANOV, *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692, (2019).

[94] L. LOGESWARAN AND H. LEE, *An efficient framework for learning sentence representations*, arXiv preprint arXiv:1803.02893, (2018).

[95] H. P. LUHN, *A statistical approach to mechanized encoding and searching of literary information*, IBM Journal of research and development, 1 (1957), pp. 309–317.

[96] J. MA, P. CUI, K. KUANG, X. WANG, AND W. ZHU, *Disentangled graph convolutional networks*, in International conference on machine learning, PMLR, 2019, pp. 4212–4221.

[97] M. MARELLI, S. MENINI, M. BARONI, L. BENTIVOGLI, R. BERNARDI, AND R. ZAMPARELLI, *A sick cure for the evaluation of compositional distributional semantic models*, in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), 2014, pp. 216–223.

[98] B. McCann, J. Bradbury, C. Xiong, and R. Socher, *Learned in translation: Contextualized word vectors*, Advances in neural information processing systems, 30 (2017).

[99] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, *The natural language decathlon: Multitask learning as question answering*, 2018.
arXiv preprint arXiv:1806.08730.

[100] D. Metzler and W. B. Croft, *Analysis of statistical question classification for fact-based questions*, Information Retrieval, 8 (2005), pp. 481–504.

[101] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, (2013).

[102] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, in Advances in neural information processing systems, 2013, pp. 3111–3119.

[103] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, *Key-value memory networks for directly reading documents*, arXiv preprint arXiv:1606.03126, (2016).

[104] G. A. Miller, *Wordnet: a lexical database for english*, Communications of the ACM, 38 (1995), pp. 39–41.

[105] S. M. Mohammad and G. Hirst, *Distributional measures of semantic distance: A survey*, arXiv preprint arXiv:1203.1858, (2012).

[106] A. Mohasseb, M. Bader-El-Den, and M. Cocea, *Question categorization and classification using grammar based approach*, Information Processing & Management, 54 (2018), pp. 1228–1243.

[107] D. Mollá, M. Van Zaanen, and D. Smith, *Named entity recognition for question answering*, in Proceedings of the Australasian language technology workshop 2006, 2006, pp. 51–58.

[108] B. Pang and L. Lee, *A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts*, in Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), 2004, pp. 271–278.

[109] ——, *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*, in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL,Äô05), 2005, pp. 115–124.

[110] M. Paşca, *Open-domain question answering from large text collections*, 2003.

[111] J. PEARL, *Causal and counterfactual inference*, The Handbook of Rationality, (2018), pp. 1–41.

[112] J. PENNINGTON, R. SOCHER, AND C. MANNING, *GloVe: Global vectors for word representation*, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, Oct. 2014, Association for Computational Linguistics, pp. 1532–1543.

[113] J. PENNINGTON, R. SOCHER, AND C. D. MANNING, *Glove: Global vectors for word representation*, in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[114] M. E. PETERS, M. NEUMANN, M. IYYER, M. GARDNER, C. CLARK, K. LEE, AND L. ZETTLEMOYER, *Deep contextualized word representations*, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, June 2018, Association for Computational Linguistics, pp. 2227–2237.

[115] ——, *Deep contextualized word representations*, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, June 2018, Association for Computational Linguistics, pp. 2227–2237.

[116] F. PETRONI, T. ROCKTÄSCHEL, P. LEWIS, A. BAKHTIN, Y. WU, A. H. MILLER, AND S. RIEDEL, *Language models as knowledge bases?*, arXiv preprint arXiv:1909.01066, (2019).

[117] R. PUDUPPULLY, L. DONG, AND M. LAPATA, *Data-to-text generation with content selection and planning*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 6908–6915.

[118] A. RADFORD, K. NARASIMHAN, T. SALIMANS, AND I. SUTSKEVER, *Improving language understanding by generative pre-training*, (2018).

[119] A. RADFORD, J. WU, R. CHILD, D. LUAN, D. AMODEI, I. SUTSKEVER, ET AL., *Language models are unsupervised multitask learners*, OpenAI blog, 1 (2019), p. 9.

[120] C. RAFFEL, N. SHAZEER, A. ROBERTS, K. LEE, S. NARANG, M. MATENA, Y. ZHOU, W. LI, AND P. J. LIU, *Exploring the limits of transfer learning with a unified text-to-text transformer*, Journal of Machine Learning Research, 21 (2020), pp. 1–67.

[121] P. RAJPURKAR, J. ZHANG, K. LOPYREV, AND P. LIANG, *Squad: 100,000+ questions for machine comprehension of text*, in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 2383–2392.

[122] P. RAMACHANDRAN, P. J. LIU, AND Q. LE, *Unsupervised pretraining for sequence to sequence learning*, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 383–391.

[123] I. REDKO, E. MORVANT, A. HABRARD, M. SEBBAN, AND Y. BENNANI, *Advances in domain adaptation theory*, Elsevier, 2019.

[124] S. REED, K. ZOLNA, E. PARISOTTO, S. G. COLMENAREJO, A. NOVIKOV, G. BARTH-MARON, M. GIMENEZ, Y. SULSKY, J. KAY, J. T. SPRINGENBERG, ET AL., *A generalist agent*, arXiv preprint arXiv:2205.06175, (2022).

[125] N. REIMERS AND I. GUREVYCH, *Sentence-bert: Sentence embeddings using siamese bert-networks*, 2019.
Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

[126] B. B. RIEGER, *On distributed representation in word semantics*, International Computer Science Institute Berkeley, CA, 1991.

[127] A. ROBERTS, C. RAFFEL, AND N. SHAZEER, *How much knowledge can you pack into the parameters of a language model?*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 5418–5426.

[128] S. RUDER, *An overview of multi-task learning in deep neural networks*, arXiv preprint arXiv:1706.05098, (2017).

[129] D. E. RUMELHART, G. E. HINTON, R. J. WILLIAMS, ET AL., *Learning representations by back-propagating errors*, Cognitive modeling, 5, p. 1.

[130] D. SÁNCHEZ, M. BATET, D. ISERN, AND A. VALLS, *Ontology-based semantic similarity: A new feature-based approach*, Expert systems with applications, 39 (2012), pp. 7718–7728.

[131] V. SANH, L. DEBUT, J. CHAUMOND, AND T. WOLF, *Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter*, arXiv preprint arXiv:1910.01108, (2019).

[132] M. SCARDAMALIA AND C. BEREITER, *Text-based and knowledge based questioning by children*, Cognition and instruction, 9 (1992), pp. 177–199.

[133] T. SCHEEPERS, *Improving the compositionality of word embeddings*, Master's thesis, Universiteit van Amsterdam, Science Park 904, Amsterdam, Netherlands, 11 2017.

[134] M. SEO, J. LEE, T. KWIATKOWSKI, A. P. PARIKH, A. FARHADI, AND H. HAJISHIRZI, *Real-time open-domain question answering with dense-sparse phrase index*, arXiv preprint arXiv:1906.05807, (2019).

[135] P. H. SEO, A. LEHRMANN, B. HAN, AND L. SIGAL, *Visual reference resolution using attention memory for visual dialog*, Advances in neural information processing systems, 30 (2017).

[136] L. SHANG, Z. LU, AND H. LI, *Neural responding machine for short-text conversation*, arXiv preprint arXiv:1503.02364, (2015).

[137] V. SHWARTZ AND I. DAGAN, *Still a Pain in the Neck: Evaluating Text Representations on Lexical Composition*, Transactions of the Association for Computational Linguistics, 7 (2019), pp. 403–419.

[138] R. F. SIMMONS, *Answering english questions by computer: a survey*, Communications of the ACM, 8 (1965), pp. 53–70.

[139] R. F. SIMMONS, S. KLEIN, AND K. MCCONLOGUE, *Indexing and dependency logic for answering english questions*, American Documentation, 15 (1964), pp. 196–204.

[140] R. SOCHER, A. PERELYGIN, J. WU, J. CHUANG, C. D. MANNING, A. Y. NG, AND C. POTTS, *Recursive deep models for semantic compositionality over a sentiment treebank*, in Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1631–1642.

[141] D. SONG, *Student-generated questioning and quality questions: A*, Research Journal of Educational Studies and Review, 2 (2016), pp. 58–70.

[142] P. SOUCY AND G. W. MINEAU, *A simple knn algorithm for text categorization*, in Proceedings 2001 IEEE international conference on data mining, IEEE, 2001, pp. 647–648.

[143] S. SUBRAMANIAN, A. TRISCHLER, Y. BENGIO, AND C. J. PAL, *Learning general purpose distributed sentence representations via large scale multi-task learning*, arXiv preprint arXiv:1804.00079, (2018).

[144] C. SUN, X. QIU, Y. XU, AND X. HUANG, *How to fine-tune bert for text classification?*, in China National Conference on Chinese Computational Linguistics, Springer, 2019, pp. 194–206.

[145] I. SUTSKEVER, O. VINYALS, AND Q. V. LE, *Sequence to sequence learning with neural networks*, in Advances in neural information processing systems, 2014, pp. 3104–3112.

[146] S. THRUN AND L. PRATT, *Learning to learn: Introduction and overview*, in Learning to learn, Springer, 1998, pp. 3–17.

[147] M. UNIVERSITY, *Four levels of questions*.
https://libguides.mcmaster.ca/c.php?g=718529&p=5130856, 2022.
Accessed: 2022-12-17.

[148] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, 2017.
In Advances in neural information processing systems, pp. 5998-6008. 2017.

[149] P. VINCENT, H. LAROCHELLE, Y. BENGIO, AND P.-A. MANZAGOL, *Extracting and composing robust features with denoising autoencoders*, in Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.

[150] A. WANG, Y. PRUKSACHATKUN, N. NANGIA, A. SINGH, J. MICHAEL, F. HILL, O. LEVY, AND S. BOWMAN, *Superglue: A stickier benchmark for general-purpose language understanding systems*, Advances in neural information processing systems, 32 (2019).

[151] A. WANG, A. SINGH, J. MICHAEL, F. HILL, O. LEVY, AND S. BOWMAN, *Glue: A multi-task benchmark and analysis platform for natural language understanding*, in Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2018, pp. 353–355.

[152] L. WANG, N. YANG, X. HUANG, B. JIAO, L. YANG, D. JIANG, R. MAJUMDER, AND F. WEI, *Text embeddings by weakly-supervised contrastive pre-training*, arXiv preprint arXiv:2212.03533, (2022).

[153] S. WANG, M. YU, X. GUO, Z. WANG, T. KLINGER, W. ZHANG, S. CHANG, G. TESAURO, B. ZHOU, AND J. JIANG, *R 3: Reinforced ranker-reader for open-domain question answering*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[154] S. WANG, M. YU, J. JIANG, W. ZHANG, X. GUO, S. CHANG, Z. WANG, T. KLINGER, G. TESAURO, AND M. CAMPBELL, *Evidence aggregation for answer re-ranking in open-domain question answering*, arXiv preprint arXiv:1711.05116, (2017).

[155] M. WATTS, S. ALSOP, G. GOULD, AND A. WALSH, *Prompting teachers‚Äô constructive reflection: Pupils‚Äô questions as critical incidents*, International Journal of Science Education, 19 (1997), pp. 1025–1037.

[156] R. WHITE AND R. GUNSTONE, *Probing understanding*, Routledge, 2014.

[157] J. WIEBE, T. WILSON, AND C. CARDIE, *Annotating expressions of opinions and emotions in language*, Language resources and evaluation, 39 (2005), pp. 165–210.

[158] J. WIETING, M. BANSAL, K. GIMPEL, AND K. LIVESCU, *Towards universal paraphrastic sentence embeddings*, 2016.

[159] A. Williams, N. Nangia, and S. Bowman, *A broad-coverage challenge corpus for sentence understanding through inference*, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 1112–1122.

[160] J. Worsham and J. Kalita, *Multi-task learning for natural language processing in the 2020s: where are we going?*, Pattern Recognition Letters, 136 (2020), pp. 120–126.

[161] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., *Google's neural machine translation system: Bridging the gap between human and machine translation*, arXiv preprint arXiv:1609.08144, (2016).

[162] Z. Xu and N. Cristianini, *Qbert: Generalist model for processing questions*, in Advances in Intelligent Data Analysis XXI: 21st International Symposium on Intelligent Data Analysis, IDA 2023, Louvain-la-Neuve, Belgium, April 12–14, 2023, Proceedings, Springer, 2023, pp. 472–483.

[163] Z. Xu, Z. Guo, and N. Cristianini, *On compositionality in data embedding*, in Advances in Intelligent Data Analysis XXI: 21st International Symposium on Intelligent Data Analysis, IDA 2023, Louvain-la-Neuve, Belgium, April 12–14, 2023, Proceedings, Springer, 2023, pp. 484–496.

[164] Z. Xu, A. Howarth, N. Briggs, N. Cristianini, et al., *What makes us curious? analysis of a corpus of open-domain questions*, in CS & IT Conference Proceedings, vol. 11, CS & IT Conference Proceedings, 2021.

[165] P. Yang, H. Fang, and J. Lin, *Anserini: Enabling the use of lucene for information retrieval research*, in Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, 2017, pp. 1253–1256.

[166] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin, *End-to-end open-domain question answering with bertserini*, arXiv preprint arXiv:1902.01718, (2019).

[167] Y. Yang, W.-t. Yih, and C. Meek, *Wikiqa: A challenge dataset for open-domain question answering*, in Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 2013–2018.

[168] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, *Xlnet: Generalized autoregressive pretraining for language understanding*, 2019.
In Advances in neural information processing systems 32.

[169] X. Zhang, J. Zhao, and Y. LeCun, *Character-level convolutional networks for text classification*, 2015.
In Advances in neural information processing systems.

[170] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 19–27.

[171] C. Zirn, M. Niepert, H. Stuckenschmidt, and M. Strube, *Fine-grained sentiment analysis with structural features*, in Proceedings of 5th International Joint Conference on Natural Language Processing, 2011, pp. 336–344.