Survivorship Bias in Sparsely-Labeled Information Retrieval Datasets

Prashansa Gupta

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

13 December 2021

Abstract

Survivorship bias, also often referred to as survival bias, is the tendency to concentrate on the positive outcomes of a selection process and overlook the results that generate negative outcomes. Survivorship bias is commonly observed in several real-life instances wherein overlooking the negative outcomes can result in erroneous judgments. This dissertation focuses on analysing the extent to which survivorship bias persists in the MS MARCO Question Answering dataset. MS MARCO is a sparsely-labelled dataset in which each query has a few relevant documents. 45% of the queries in the MS MARCO question answering development dataset are unanswered, i.e., these queries have no relevant documents. There are three factors that can result in a query being unanswered. Firstly, the type of query can affect the answer. The queries in MS MARCO are automatically annotated by the ”type” information using a classifier. The queries of type ”DESCRIPTION” and ”NUMERIC” have the highest amount of unanswered queries. Secondly, the number of top retrieved documents can affect whether a query is answered. 99.98% of answered queries have one or more relevant documents in the top 10 retrieved documents. If the amount of top documents retrieved is lessened, the percentage of answered queries in the dataset decrease. Thirdly, the ranking of documents in the corpus has a major impact, specifically, the reranked documents that appear in the top 10. To check the impact of the rank threshold on the dataset, subsets of the development set are extracted on the basis of the rank threshold and these subsets are tested against passage reranking results of rankers such as BM25, MonoT5, ANCE, and COLBERT. It is found that MRR@10 for the development set increases when the rank threshold of 10 is imposed on the queries.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: Prashansa Gupta Signature: Prashansa

Acknowledgements

I would like to thank my supervisor Mr. Sean MacAvaney for his constant support during this dissertation. Mr. MacAvaney has guided me through the process and his inputs have immensely helped me. I would also like to extend my gratitude to my family for encouraging and inspiring me.

Contents

Chapter 1

Introduction

1.1 Motivation

Survivorship bias is when one focuses more on the favorable outcomes rather than outcomes with poor conclusions [4]. This stems from the cognitive ability of the human brain to remember the successes rather than the failures. Survivorship bias can cause a lapse in judgement as it distorts the decision making process by blurring out the negative consequences.

A popular example of survivorship bias is the World War II planes. The United States armed forces observed that the returning planes had the maximum bullet marks along the body, wings and tail of the plane, causing them to believe that these were the parts of the plane which needed to be reinforced with armour. However, statistician Abraham Wald pointed out that it was the other parts that needed reinforcement, as the bullets to those areas have proved fatal for the plane [9]. Thus, by studying the patterns in data that did not pass the selection process, helpful conclusions can be made. Another popular example of survivorship bias is that rich and successful people have dropped out of college. This is a distorted inclination because not everyone dropping out of college will be successful [13].

Survivorship bias can be observed easily across several streams ranging from commercial to medical, and biases of such kind can give a skewed view of reality. It even affects the decision making and opinions of people when it comes to music, architecture and even career choices [4].

Survivorship bias is a cognitive bias on data, which is why the existence of such bias can be significant in data science. It has become increasingly common in IR evaluation to test on datasets with thousands queries but only a few known relevant documents per query (sparse datasets), rather than the traditional paradigm of using dozens of queries with many known relevant documents per query (dense datasets). Relatively little attention has been brought to various biases that sparse datasets exhibit. One possible bias is a survivorship bias; namely, that queries that are challenging to label are simply discarded. For sparsely-labelled dataset, labelling a query depends on the position of the positive relevance labels. Queries that have positive relevance at the top ranks are easier to label, as compared to queries with sparse positive relevance at the greater ranks.

The motivation of this dissertation is to combine the concepts of survivorship bias with the sparsely labelled MS MARCO - a popular dataset used for training and evaluating neural IR systems.

1.2 Aim and Objective

This dissertation focuses on two main objectives:

Objective 1: Investigate the MS MARCO dataset and derive patterns in answered and unanswered queries.

Objective 2: Investigate the extent to which survivorship bias is present in a representative sparse dataset (MS MARCO) and whether this bias can impact other studies.

1.3 Outline

This report is primarily divided into 6 chapters. The current section is the Chapter 1 of the report which highlights the motivation for the dissertation, the aim and objective and the outline of the report. Chapter 2 delves deeper into the theoretical background relating to this dissertation. In that section, the principal dataset MS MARCO is discussed. Additionally, the ranking models like BM25, MonoT5, ANCE and COLBERT are studied. This chapter also explores passage reranking in the MS MARCO dataset. Chapter 3 includes in-depth analysis of the MS MARCO dataset to derive the patterns in the answered and unanswered queries. This chapter is subdivided into 6 sections that showcase the experiments, and observations that are found during the analysis. Chapter 4 pursues the main objective of this dissertation and sets up the experimental design. It is subdivided into two sections that demonstrate the different stages of the experiment. Chapter 5 is focused around calculating and studying the findings of chapter 4 and finally, Chapter 6 comprises the conclusions that are drawn from the experiments, suggestions for future work and personal reflection.

Chapter 2

Background survey

The aim of this chapter is to review the theoretical background related to this dissertation, with the goal of introducing the dataset and explaining the concepts and models that are key to understanding the remainder of this dissertation. The domain of this dissertation revolves around machine reading comprehension and information retrieval, along with exploratory data analysis. In this chapter, the MS MARCO dataset is discussed comprehensively, followed by the ranking models that will be employed in subsequent chapters. Further, the passage reranking of the MS MARCO dataset is explored.

2.1 MS MARCO

MS MARCO stands for **Microsoft MAchine Reading COmprehension** and it is a machine reading comprehension dataset that consists of anonymized real-time queries sampled from Bing and Cortana, with an emphasis on queries that seem like a question. This dataset focuses on machine reading comprehension, along with question answering and passage reranking. MS MARCO provides a more ”natural” distribution of queries because it is sampled from real queries submitted from real users, while the majority of machine reading comprehension datasets involve queries that are synthetic or curated by crowd-source workers on the basis of provided documents[2].

MS MARCO combines the domain of information retrieval with human interaction, because the passages that are retrieved corresponding to a query are presented to a human editor. The task of the editor is to analyse the retrieved passages and provide relevance judgements by rating the passage as 1 if relevant or 0 if irrelevant. The queries with a 0 relevance across the retrieved passages is deemed as unanswered. The editors are not expected to annotate all the relevant passages for the query. For the queries that have relevant passages, the editors are then asked to create answers which are manually composed. There are also ’well-formed answers’ which furthers the annotator-generated answers by removing grammatical errors and restructuring the answer to remove text overlap and contextual errors [2].

Example of an answered query from the MS MARCO dataset:

```
query id:  663771
query text:  what geological features are shared by all terrestrial planets
query type:  ENTITY
query answer:  Each are composed primarily of silicate rock and metal.
```

Figure 2.1: Example of an answered query

MS MARCO dataset tackles real-world text queries, thus it entails the messiness of real text. The queries in the dataset could have grammatical errors, spelling mistakes or even incomplete/unclear formulation of the question. By saving the unanswered queries in the machine reading comprehension dataset, these queries can be further analysed to recognise insufficient information. MS MARCO provides a diversity of real-time queries, thus making it robust to incomplete and noisy inputs.

Three different machine learning tasks are proposed by the creators of the MS MARCO dataset. The novice task, which determines whether a query can be answered solely based on the passages. The system should return the correct answer if the retrieved passages contain the answer, or else return ”No Answer Present”. The intermediate task is an extension of the novice task but the answer should be well-formed, i.e. the answer should make sense if it is read out loud. The passage reranking task involves reranking the retrieved passages using BM25 [2].

2.2 Ranking models

This section discusses the ranking models of information retrieval that are essential for this dissertation. These models are used in the scope of this dissertation to re-rank the MS MARCO passage dataset with the intent of finding out whether unanswered queries can be answered when reranking of the top passages is done. The ranking models BM25, MonoT5, ANCE and COLBERT are used because these models are representative of several paradigms: lexical models, neural re-ranking, single-representation dense retrieval, multi-representation dense retrieval.

2.2.1 BM25

BM25 [18] is a popular ranking algorithm that estimates the relevance between a query and a document. BM25 model is built around the probabilistic retrieval framework, which rather than estimating relevance as a Boolean concept, uses probabilities to gauge if a document is relevant for a query. BM25 ranks documents on basis of query terms being present in the document irrespective of the proximity of query words with each other, thus following a bag of words retrieval approach. One limitation of BM25 is that it depends mainly on lexical matches.

2.2.2 MonoT5

MonoT5 [7] is a point-wise re-ranker that works on the concept of relevance classification. Relevance classification essentially converts the task of ranking the text into a classification problem, and the ranking of the text is done by sorting the probabilities. This approach is a variation of the ”Probability ranking principle” in which document-ranking is done by sorting probabilities in

decreasing order. [17]. More precisely, by training a classifier, the provided text is essentially classified as relevant and irrelevant texts with certain probabilities. These probabilities are then sorted at the time of ranking to acquire the ranked texts in order of relevance.

The MonoT5 ranker calculates a score by measuring how relevant a given document is for a given query:

$$P(Relevant = 1|document, query)$$

MonoT5 works on the T5 transformer model. Given the text of query and document, it converts each task into a text-to-text format. [16]. The model produces the tokens "true" or "false" as target tokens. Further, the softmax function is used specifically on the logits of these tokens and the documents are reranked according to the probabilities [15].

2.2.3 ANCE

ANCE [19] stands for Approximate Nearest-Neighbor Negative Contrastive Estimation and it is the learning procedure that selects hard-negatives by making use of the approximate nearest neighbour index. The model recognises the positive documents in the collection and separates them from the negative ones. The nearest neighbours in the collection are the hardest negatives for the model as they were chosen by the model itself.

The query and the passage are encoded by utilising BERT to produce embeddings. Later, the dot product of these embeddings are taken and the pair that has the maximum value is the positive match. Approximate Nearest Neighbor index updates asynchronously and selects the embeddings that are close to the positive match.

The ANCE learning mechanism can be showcased as:

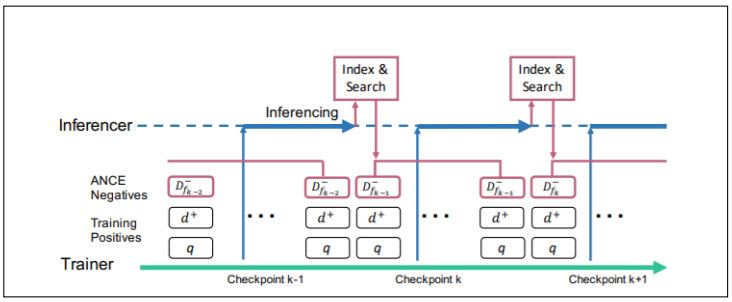


Figure 2.2: Learning mechanism of ANCE
[19]

2.2.4 COLBERT

COlBERT [5] stands for Contextualized Late Interaction over BERT and it is a ranking model centered around deep language models like BERT. COLBERT uses "late interaction" to gauge the

relevance between a document and a query. In this approach, the contextual embedding of the documents and the query are pre-computed separately. The relevance between these sets are further evaluated using pruning-friendly computations.

The below figure showcases the general architecture of ColBERT:

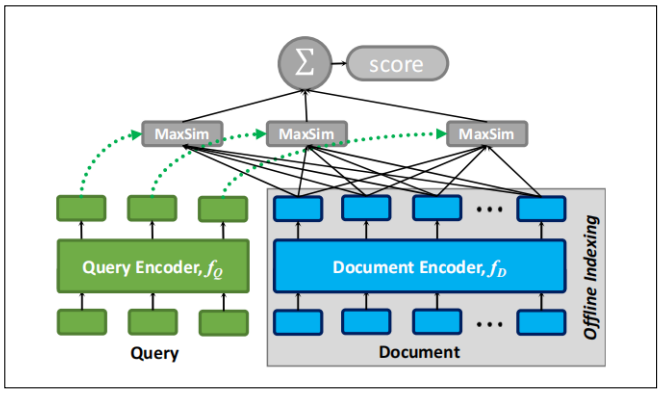


Figure 2.3: General architecture of colbert
[5]

ColBERT can also be utilized to perform top-k reranking on the output of other retrieval models. The queries are represented by the bag of contextual embeddings and the documents are represented as 3D tensors which contain k document matrices. The batch dot product is then calculated between the query and documents over several mini-batches. The output of the dot product is saved into a tensor, which is essentially a set of matrices between a query and each document. These matrices are then max-pooled and a summation is performed over the query terms to compute the score. These scores are then utilised to sort the k documents.

ColBERT provides cheaper computation as compared to the other neural rankers. In contrast to BERT, ColBERT provides BERT with a sequence of the query length as opposed to the added length of both query and document in BERT. Not only is ColBERT economical, but it also fares well as the values of k are changed.

2.3 Passage reranking in the MS MARCO dataset

Passage reranking is the third task proposed by the creators of MS MARCO dataset [2]. The passage reranking task involves pooling the 8.8 million unique passages of the MS MARCO passage dataset and reranking the passages for each query, which gives 1000 passages for each query. These 1000 passages are independent of the top 10 passages given to the annotators and there is a possibility that the retrieved 1000 passages do not contain passages that are marked as relevant by the annotators/editors. The success of the passage reranking task is determined by judging how highly the relevant passage is placed after re-ranking [12].

The results retrieved by BM25 can further be reranked by other ranking models involving advanced computations. This can be achieved by calculating a score that depicts the probability of relevance of a query to the document and then sorting these scores to obtain the top retrieved documents corresponding to the query [14].

Chapter 3

Data analysis

The scope of this dissertation focuses on two information retrieval datasets: msmarco-passage and msmarco-qna [2]. The ”msmarco-qna” is the question answering dataset which is used to study the queries and the retrieved documents for each query. This dataset is essential for the analysis of queries to find out the nature of the queries that are answered or unanswered. The ”msmarco-passage” is a benchmark dataset that is essential for reranking the passages that are retrieved for each query. The terms ”documents” and ”passages” are interchangeable for the scope of this dissertation.

In this chapter, the MS MARCO Question Answering dataset is examined to extract information and statistics from the distribution of the data. Various aspects of the msmarco-qna dataset will be explored to increase the understanding of this data. Firstly, **msmarco-qna/dev** is examined to check the basic structure of the dataset and the proportion between answered and unanswered queries. The second section focuses on the distribution of queries based on the attribute ”type”. The third section of the chapter involves analysis of the answered queries of the development set. The fourth section focuses on the manual analysis of the queries of the development and the training set, outlining the findings done by this analysis. In the fifth section, this manual analysis is furthered by reranking the passages of the newer MS MARCO passage dataset to check whether 50 selected queries can be answered. The final chapter focuses on the shortcomings of the MS MARCO dataset which were observed during the earlier section.

3.1 Structure of the msmarco-qna dataset and distribution of queries

In this section, firstly the structure and elements of the MS MARCO dataset are discussed and then the distribution of queries in the development set is observed. The MS MARCO dataset consists of training, development, and evaluation sets. The development and training sets of MS-MARCO contain the relevance rankings, while the rankings for the evaluation set are kept hidden [2].

ms-marco/qna	Number of queries
training set	808731
development set	101093
evaluation set	101092

The "**msmarco-qna/train**" consists of 808731 queries whereas the "**msmarco-qna/dev**" set consists of 101093 queries. For the scope of this disseration, the analysis is done on the msmarco-qna/dev set to manage the total number of queries.Thus, the findings and experiments are focused mainly on the development set rather than the training set.

The structure of the MS MARCO Question and answering development set can be described through the below diagram:

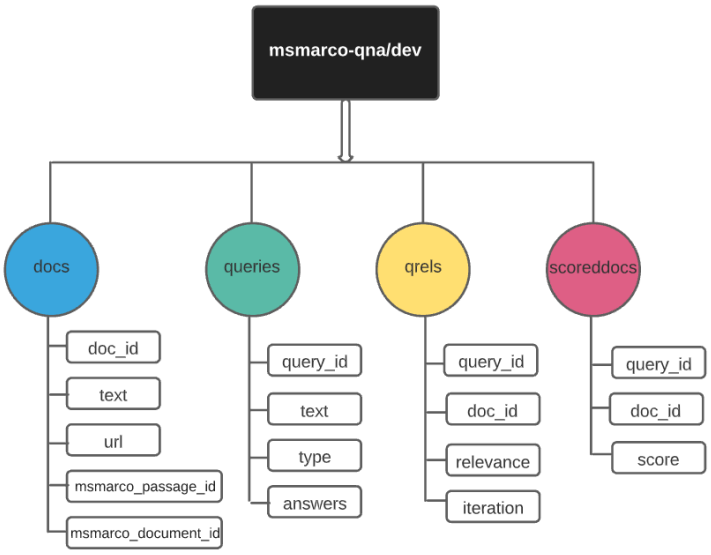


Figure 3.1: msmarco-qna/dev

The **docs** element of the dataset provides information about the document being ranked, such as doc_id, text, the url of the document and the msmarco_passage_id and msmarco_id_document_id. The **queries** element of the dataset provides information about the query such as query_id, text, type and answers. The **scoreddocs** element maps the query id with the 10 documents that were given to the editor and the score denotes the order in which these queries were presented. Lastly, the **qrels** element maps a query with the corresponding documents and the relevance attribute specifies whether a document is deemed relevant to answer the query. The value 1 signifies that a document is useful in order to answer a query. The relevance of 0 across all documents for a query means that the query is unanswered.

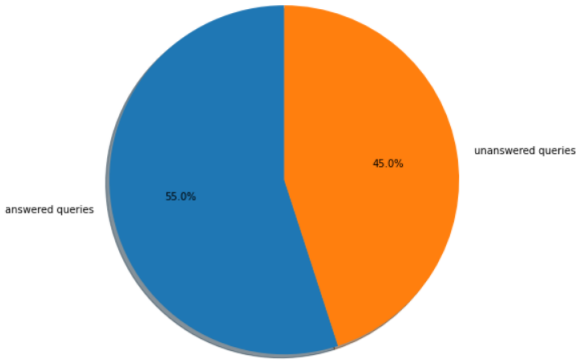


Figure 3.2: Distribution of queries in msmarco-qna/dev dataset

The decision of whether a query is answered or not depends on the relevance judgements given by the editors/annotators who judge the query text and the retrieved texts to decide the answer for the query.

Upon analysis of the attribute ”relevance”, it can be inferred that 45% of the total queries in the development dataset are unanswered, i.e. none of the top documents retrieved for the query could be deemed relevant to answer the query by the annotator. Given that the ”msmarco-qna” dataset is focused on finding answers for query, 45% of the queries being unanswered can affect the studies that use MS MARCO for training and evaluation.

3.2 Distribution of the type of queries in the MS MARCO development set

The section 3.1 highlights that 45% of the total queries in the msmarco-qna/dev are unanswered. To further understand the nature of queries that are left unanswered, segment information from the queries can be utilised.

The queries in the MS MARCO dataset are divided into 5 categories: DESCRIPTION (when the expected answer to the query is descriptive), NUMERIC(when the expected answer of the query is numerical), ENTITY(when the expected answer to the query is a single entity), PERSON(when the query's expected answer is a person) or LOCATION(when the expected answer to the query is a location) [2].

This annotation is done automatically with the help of a classifier. The candidate passages shown to the annotators do not have information about the type of query, yet this attribute can help in discerning the answerability of a query.

Upon analysis of the attribute ”type”, the distribution of queries of various type can be seen below:

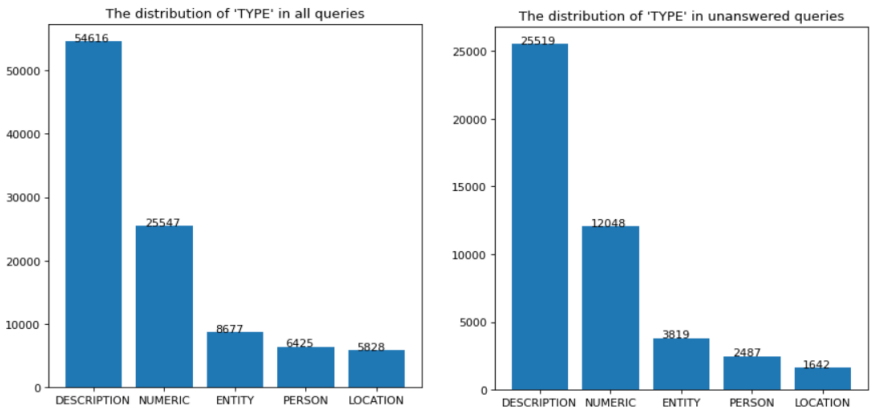


Figure 3.3: Segment Distribution

From the above figures, it can be inferred that the ”DESCRIPTION” queries constitute the biggest part of the distribution, followed by ”NUMERIC”. This congruence is maintained in the unanswered queries wherein ”DESCRIPTION” and ”NUMERIC” are the segments with highest number.

The below table extends the statistics of figure 3.3 to further determine what proportion of the each ”TYPE” of query is unanswered.

TYPE	Total queries	Unanswered queries	%Unanswered
DESCRIPTION	54616	25519	46.72
NUMERIC	25547	12048	47.16
ENTITY	8677	3819	44.01
PERSON	6425	2487	38.71
LOCATION	5828	1642	28.17

Table 3.1: Segment Distribution of unanswered queries

From the above table, it can be inferred that queries of type ”PERSON” and ”LOCATION” are more likely to be answered whereas queries of ”NUMERIC” and ”DESCRIPTION” type are unanswered at a higher rate. Thus, the analysis of type distribution gives more insight about the dataset. These findings are inspected in deeper detail in 3.4 where in the queries of different types are analysed to see why certain queries are answered more frequently than others.

3.3 Analysis of answered queries

After measuring the distribution of the type of queries, this section focuses on the queries with relevant documents to find information about the number of queries that are answered at each rank. From the table 3.1, it can be inferred that the queries of type ”LOCATION” and ”PERSON” are mostly answered in the development set.

The development set consists of queries that have relevance rankings ranging from document at rank 1 to rank 21, i.e. the top 21 documents. 55568(99.982%) of the 55578 queries have relevance ranking of 1 within the top 10 documents, i.e. most of the queries are answered within the top 10 documents.

The distribution of queries answered at a specific rank position can be seen below:

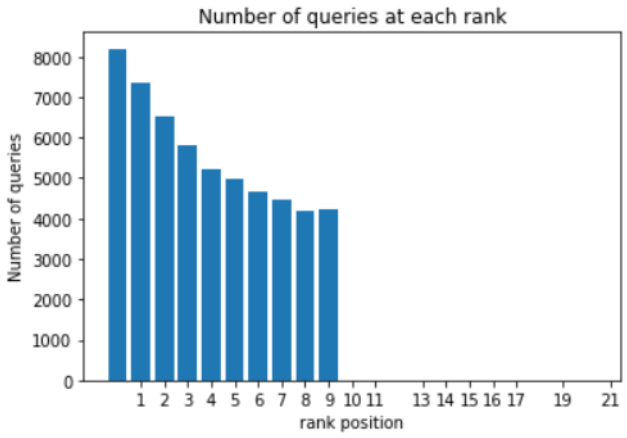


Figure 3.4: Number of queries at each rank

From the above figure it can be inferred that 8197 of 55578(**14.7%**) queries are answered by the document at rank 1, meaning that the topmost document was marked as the relevant document by the annotator. To further analyse the documents which are at rank 1, the ”type” distribution of these rank 1 queries are calculated.

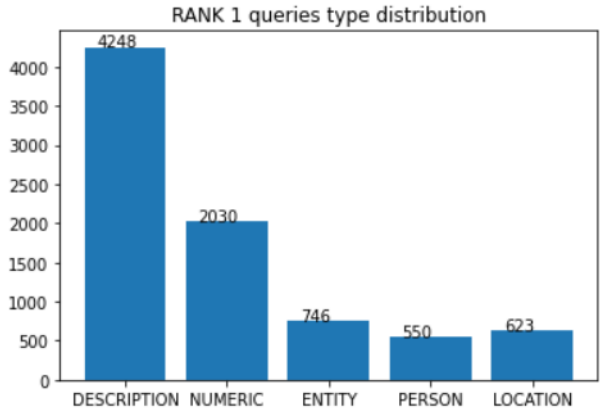


Figure 3.5: Number of queries at rank 1

The above figure outlines that the majority of the queries being answered at rank 1 were of ”DISTRIBUTION” type, but this behaviour is consistent with the original distribution of the queries, i.e. ”DESCRIPTION” queries constitute the major portion of the development set which is why the most rank 1 queries belong to this type.

To draw out further information from the figure 3.5, other approach can be to check how many of the total ”DESCRIPTION” queries were ranked at rank 1, rank 2 and so on. This analysis would give an estimate about whether queries of a certain type are generally ranked higher or lower.

The below table showcases the distribution of queries across different types for different ranks:

RANK	Total queries	%DESCRIPTION	%NUMERIC	%ENTITY	%PERSON	%LOCATION
1	8197	14.6%	15%	15.4%	14%	14.9%
2	7346	13.1%	13.4%	13.1%	14.2%	13%
3	6517	11.6%	12.2%	11.2%	12.6%	11.4%
4	5798	10.6%	9.7%	10.7%	10.2%	11.1%
5	5204	9.4%	9.3%	9.9%	9.1%	9%
6	4962	8.9%	8.8%	9.6%	8.6%	8.7%
7	4645	8.5%	8.4%	8%	7.4%	8.2%
8	4477	8%	8%	7.9%	8.1%	8.3%
9	4198	7.4%	7.6%	7.1%	8.6%	8.1%
10	4224	7.8%	7.5%	7.1%	7.2%	7.3%

According to the figure 3.4, it was found that 14.7% of the total queries are ranked at 1 in the development set. The in-depth analysis of queries at each rank from the above table outlines the percentage distribution of queries at each rank.

It could be inferred that the queries at rank 1 have almost similar proportions of each type, although type ”NUMERIC” and ”ENTITY” have a slightly higher percentage, meaning that comparatively higher number of queries that are of type ”ENTITY” can be answered by the top retrieved document. The distribution is consistent at the other ranks.

Thus, by analysing the distribution of type in the answered queries, it can be inferred that the type of query does not have much impact on the possibility of it being answered at a certain rank.

3.4 Manual analysis of unanswered queries

The objective of this section is to monitor unanswered queries from the MS MARCO Question Answering dataset with the aim of deriving more information around the findings of section 3.1. From the figure 3.2, it is evident that 45% of the total queries from the development set are unanswered. Given that these queries are derived from the search logs of Bing and Cortana, this statistic implies that almost half of the queries submitted by the users go unanswered, which can be problematic as the prime motive of the search engine is to derive answers for users' queries.

To dissect the unanswered queries further, 100 unanswered queries from the development set and 50 unanswered queries from the training set were selected and then the top retrieved passages for each of these queries were examined and annotated. The annotation information is summarised below:

Annotation	Description
0	If the retrieved passages do not answer the query.
1	If the retrieved passages answer the query and it was mistakenly marked as unanswered.
2	If the retrieved passages partially answer the query, and the query could have been answered if it was more specific.
3	If the query is non-answerable or invalid, i.e. the information being asked for is not clear or non-existent.

After annotation, the development set and training set queries showed the below distribution:

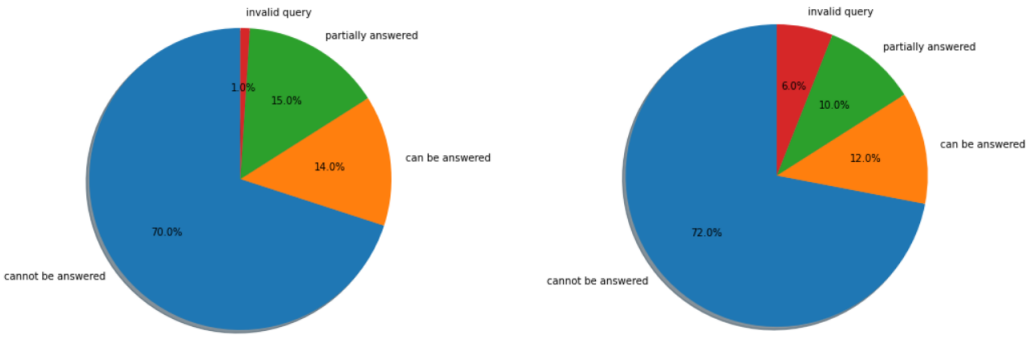


Figure 3.6: Annotation distribution on dev and train set

The following observations were made:

1. Out of the 150 queries considered from the development and training sets, 71% of the queries could not be answered by the retrieved passages.
2. 13.3% of the total queries could be answered by the retrieved passages, yet the query was marked as unanswered. This could be attributed to the fact that the answerability of a query

depends very highly on the judgement of the annotator. These queries may not be answered either due to the documents retrieved not being relevant enough for the query according to the annotator or could be annotator error.

3. 12% of the total queries were unanswered as the retrieved passages only partially answered the query. This could be due to two reasons. Firstly, if the query text is not specific enough and the retrieved passages are specific. Secondly, if the query requires more details than the passages. For instance, a query which is instructional in nature are not answered as the passages contain only certain parts of the required answer.

4. 4% of the queries were unanswered as it is difficult to determine the intent or expected answer for a query from the query text. This happens in two cases. Firstly, when the query is not a question, rather a phrase which makes it harder for to determine the relevant passages. Secondly, when the query is wrongly worded and the information being asked for doesn't exist. An example of this is the following query:

```
query id:  648473
query text:  what does stack the board mean
query type:  DESCRIPTION
query answers:  ()
```

   The correct idiom is "stack the deck". The query is not correctly worded, causing the retrieved passages to be irrelevant to what is actually required.

5. In several queries, a repetition is observed in the retrieved passages, which could be a contributing factor towards the queries being unanswered. If repeating passages are filtered out, that can facilitate unique relevant passages to be included in the top 10 retrieved documents. This could help in the answering the query.

The following observations were made with respect to the "type" of queries:

1. The queries that are **"NUMERIC"** in nature are mainly calculative, so the retrieved documents that have instructions for calculation or formulae are disregarded as the annotators look for numeric answers to the solution.

2. The queries that are of type **"DESCRIPTION"** are more complex as the relevance of a particular document could differ based on the annotator's judgement of relevance. For instance, a query about food recipes could have varying relevance judgements across different annotators. Another reason for the description queries to have no relevant documents could be the reason that the retrieved passage does not entirely answer the query, rather have incomplete information about the query.

3. The queries that are of type **"LOCATION"** are more distinct in nature, often needing specific answers for the query. This minimises the ambiguity around the query as was observed in the "DESCRIPTION" queries.

4. The queries that are of type **"ENTITY"** and **"PERSON"** are simpler in structure as compared to the **"DESCRIPTION"** queries. Upon observation it is observed that these are generally unanswered mainly due to the retrieved passages.

Thus, the manual analysis of queries in the section provides insights into the details and the issues associated with the unanswered queries. As the analysis on this section was based on manual inspection of queries, the findings are according to the 150 queries selected.

In 71% of the total unanswered queries, the retrieved passages could not answer the question, not even partially, which implies that either the corpus does not have the relevant passages or the passages retrieved were not ranked properly. The answerability of a query relies on the rank at which the relevant passage is present. To substantiate this, the analysis is furthered by re-ranking the unanswered queries to find out whether they can be answered.

3.5 Reranking retrieved passages in the unanswered queries

This aim of this section is to find out whether reranking the retrieved passages for a given query can affect the answerability. From the previous section 3.4, it was perceived that 71% of the unanswered queries did not contain a relevant passage in the top 10 retrieved passages. The goal is to perform an experiment which finds out whether the queries could be answered after reranking.

For the passage reranking task, 8.8 million unique passages are pooled together and the ranking model BM25 is run, which retrieves top 1000 passages corresponding to a query [12]. The BM25 results are reranked further by ranking models like MonoT5, ANCE and COLBERT and the top 10 passages retrieved for each of these rankers for each query are analysed. One addition in the experiment is the usage of the **version 2 of the MS MARCO passage dataset** to incorporate the updated corpus of 138 million passages. In the version 2 of the passage dataset, duplicate entries are removed by the authors [10].

For this experiment, 50 queries(10 of each type) were picked up from the MS MARCO question answering development set; these queries were preferably the ones that may be answered by reranking. For the selected queries, the passages from the MS MARCO passage dataset are reranked and then, top 10 reranked passages are analysed. The queries are then annotated using the rank at which the first relevant passage exists, and '-1' is assigned to the queries that could not be answered by the reranked passages.

Note that these queries are the ones which do not have a relevance judgement, i.e. no qrels. Hence, the relevant document for each of these queries is decided based on the passages retrieved and general knowledge.

The findings of these experiment are noted below:

Ranking model	**Total queries**	**%Queries answered at rank 1**	**%Queries answered in the top 5 ranks**	**%Queries unanswered**
BM25	50	14%	32%	**52%**
MonoT5	50	50%	68%	**30%**
ANCE	50	50%	66%	**26%**
ColBERT	50	46%	66%	**32%**

From the above table, it can be observed that the ranking model ANCE performs the best with only 26% unanswered queries and about 66% of the total queries being answered in the top 5 ranks. To

further analyse the queries ranked at Rank 1 for the ANCE model, the segment information of these queries is plotted and compared against the queries unanswered.

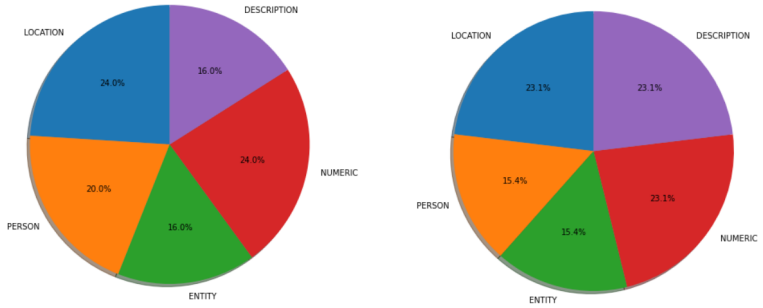


Figure 3.7: i) ANCE Rank 1 queries ii) ANCE unanswered queries

The queries of the type ”LOCATION” and ”NUMERIC” are majorly the ones that were answered at rank 1, while ”DESCRIPTION” queries have the least percentage in the rank 1 queries. Interestingly, when the unanswered queries are checked, ”LOCATION”, ”NUMERIC” and ”DESCRIPTION” are the types which are mostly unanswered.

The aim of this small experiment was to determine whether the unanswered queries of the MS MARCO development set can be answered if reranked passages are considered. Out of the total 50 queries considered, at least 48% (with BM25) could be answered. The use of neural rerankers in conjunction with the version 2 of the MS MARCO passage dataset is the reason why the selected queries could be answered upon manual inspection.

This finding implies that the number of unanswered queries in the MS MARCO dataset is heavily dependent on two factors : **the reranking of the passages** and **the rank threshold** till which the passages are considered. From the table 3.5, it can be observed that out of the total answered queries, about 66% of the queries could be answered in the top 5 retrieved documents(ANCE and COLBERT). One perspective here is that, the amount of answered queries would be much lesser if instead of top 10, only top 5 documents were being considered.

3.6 Limitations of the MS MARCO dataset

This section sheds light on the shortcomings of the MS MARCO dataset that were observed during the earlier sections of data analysis.

The following were the issues observed in the MS MARCO dataset:

1. **The retrieved set of ranked documents may not always contain the best answer.**

   The MS MARCO Question Answering dataset shows annotators the top retrieved documents for a particular query, and they judge the documents to see which ones(could be one or more documents) answer the query [2].

   The Question Answering dataset aims at finding answers for a query, rather than ranking the documents that are relevant to the query. This may be conflicting as the retrieved documents stand correct on search engine relevance but these may not necessarily answer the query in

the best sense. In fact, it has been found that the top passages retrieved after reranking the documents can be superior to the qrels [1].

2. **The MS MARCO dataset carries queries and the corresponding documents according to 2016. Upon manual analysis, it is found that most unanswered queries can be answered now.**

   While performing the analysis of the unanswered queries in the section 3.4, it was noticed that many of these queries could be answered by the Bing search engine now, which could be attributed to advances in the information retrieval of the Bing engine, and more documents being added to the internet relating to the query since 2016. For instance, consider the below query from the MS MARCO QNA training set:

```
query id: 27022
query text: artin chicken mcdonalds calories
query type: NUMERIC
answers: ()
```

Figure 3.8: Query id: 27022

The same query when searched on Bing can be answered despite the spelling mistake.

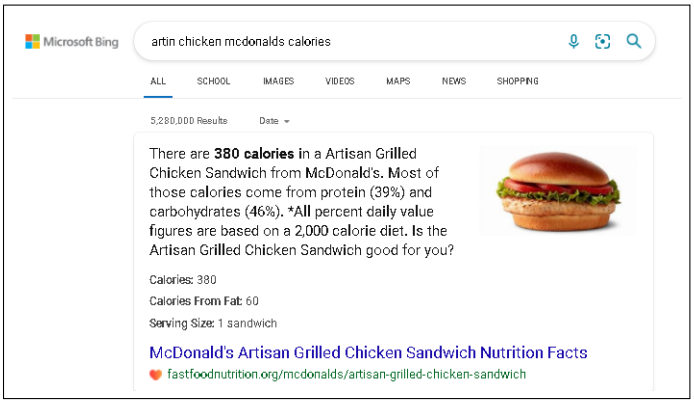


Figure 3.9: Query id: 27022

From the above observation, it can be inferred that as more documents are added on the internet, it is easier not only to derive answers for previously unanswered questions, but also to derive ”better” answers for the already answered ones.As observed in the section 3.5, many queries can be answered by using version 2 of the MS MARCO passage dataset.

3. **The attribute ”type” of a query is not always correctly annotated by the classifier.**

   In the section 3.2, the development set is examined on the basis of the segment of the queries to derive more information about unanswered queries. While this evaluation provides useful insights about the MS MARCO dataset, it was discovered during the manual analysis that certain queries had incorrect type information. For instance, consider the below query:

```
query id: 1101280
query text: do owls eat in the day
query type: NUMERIC
answers: ()
```

Figure 3.10: Query id: 1101280

It can be observed that the above query doesn't require a NUMERIC answer, and the query type should rather be "DESCRIPTION". Although the candidate passages presented to the annotators do not have type information, implying that a wrong type distribution does not affect the answerability of the query, but it affects the distribution of queries, which could affect conclusions.

4. **The decision of whether a document is relevant is based highly on the judgement of the annotator, and different annotators can have different answers for the same question.**

   The MS MARCO dataset contains the queries extracted from the Bing's search engine query logs, implying that the queries consists of a diversity of types, ranging from simple queries that require short answers to queries that require more description to be answered. The "DESCRIPTION" type queries can be further subdivided to various divisions. For example, queries could be instructional or inquisitive or exploratory.

   For instance, consider the below query from the MS MARCO QNA training set:

```
query id: 112914
query text: crockpot broccoli chicken recipe
query type: DESCRIPTION
answers: ()
```

Figure 3.11: Query id: 112914

   This is a query that can have varying answers depending on traditional aspects and even personal preferences. Depending on the annotator, queries of such nature could have 1 qrel or 4 qrels as this essentially does not have only "one" answer. This sheds light on the possibility that many queries are not limited to one qrel, and the best answer depends hugely on the judgement of the annotator [6].

5. **The retrieved documents for a query are not gauged by authenticity, which means documents of less reliability can be picked up as relevant document.**

   Another observation made in the section 3.4 is that the annotators are presented with the top documents for a given query, there is no way for them to check the authenticity of the documents being presented. For instance, for a query such as "is panglao island safe?", the documents from verified sources hold more merit over other documents. This would add another dimension to the question answering dataset as the annotators could make a more informed decision while making judgements on documents. The analysis of the legitimacy of retrieved documents is outside the scope of this dissertation.

6. **The answerability of a query is heavily dependent on the rank threshold of the documents**

   The general process of the MS MARCO dataset involves showing top 10 passages for a given query to the annotators and asking them to annotate the passages which is relevant for the query [11]. There are two main considerations here. Firstly, only those queries would be answered for whom the relevant document appears in the top 10 positions, which calls into question the ranking of the passages. The section 3.5 demonstrated that using a reranker can help in acquiring answers for many queries. Secondly, the number of queries being answered would be lesser if the annotators were asked to consider only the top 5 documents, which can be discerned from the table 3.5. The existence of rank bias can be a reason why 45% of the total queries in the development set are unanswered.

Chapter 4

Experimental Design

As seen in the limitation 6 of the section 3.6, the answerability of a query is dependent highly on the rank threshold. To further investigate this, the queries can be divided into subsets by tightening the rank threshold till which the annotators look for relevant answers. This chapter focuses on the main objective of the dissertation, i.e., analysing the extent to which survivorship bias affects the MS MARCO dataset.

In section 3.1, it can be observed that the MS MARCO dataset contains relevance rankings for each query. These rankings play an essential role in determining which document or passage at which index is being used to answer the query. Note that the terms ”passage” and ”document” are used interchangeably in the dissertation. Below is a table that summarizes how relevance rankings can be used to extract information about a query:

Query ID	Relevance ranking	Comments
302090	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	Unanswered query as it contains no relevance rankings.
364154	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	The relevant document is at rank 10,so the query is answerable in the top 10 ranks.
64084	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]	The relevant document is at rank 1, so the query is answerable in the top 1 rank.
149257	[0, 0, 1, 1, 0, 0, 0, 0, 0, 0]	The relevant documents are at ranks 3 and 4, so the query is answerable in the top 3 and 4 ranks.

The ”surviving” queries in the subsets are the queries that positively surpass the selection criteria, the criteria here is the existence of a relevant document in the specified rank threshold. Thus, by controlling the rank, it is possible to regulate how survivorship bias is affecting the MS MARCO development dataset.

Figure 3.2 displays that about 45% of the queries in the MS MARCO development set are unanswered. The reason behind these non-survival cases can be that the annotators were not able to find the relevant document for the queries in the top 10 retrieved documents. This motivates this experiment where the number of retrieved documents that are checked by the annotators is decreased to find out the ”surviving” queries. These ”surviving” queries are tested against other possible thresholds.

The experiment is divided into 2 parts:

Step 1: To derive the subset of the MS MARCO development set on the basis of the rank of the documents used to answer the queries.

Step 2: To calculate the evaluation metric **MRR@10** for the subsets derived in **Step 1** against different reranked passages.

The experiment aims to observe how the value of the evaluation metric MRR@10 changes when instead of the complete dataset, only the ”surviving” queries are considered. The further sections of this chapter focus on acquiring the data needed to perform the experiment.

4.1 Extracting subsets of the development set

The first step of the experiment is to extract the subsets from the MS MARCO development set. To achieve this, the queries having one or more relevant documents in the top 10 retrieved documents are sub-grouped, secondly, the queries having relevance of 1 in the top 9 and so on, till the queries which contain a relevant document at rank 1.

The subsets are created using the function **”subset_function”**:

```
def subset_function(rank, qids_to_relevances):
  queries_returned = []
  for key, value in qids_to_relevances.items():
    if 1 in value[:rank]:
      queries_returned.append(key)
  return queries_returned
```

In the above function, **”rank”** specifies the rank till which the relevant documents are searched, i.e. rank=10 would look for queries that have a relevance ranking of ’1’ from rank 1 to rank 10. The attribute **”qids_to_relevances”** is the mapping of a query id to the corresponding relevance rankings as defined in qrels.

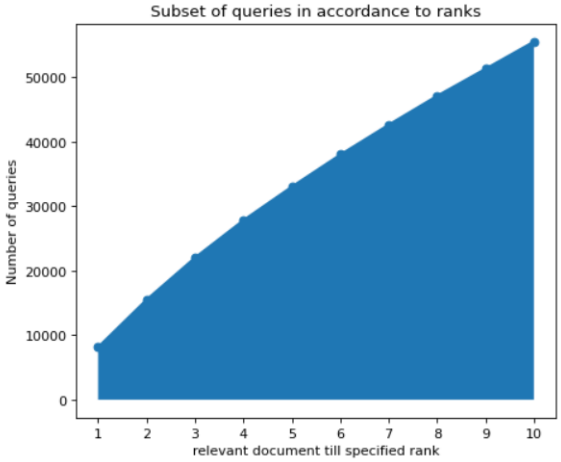


Figure 4.1: Subset of queries

As the ranks are decreased from 10 to 1, the size of the subsets decreases substantially. This is in congruence with the findings of figure 3.4 because as the rank is decreased, the queries answered only by the greater ranked documents are lost.

The retrieved subsets in this section would be used further for the step 2 of the experiment.

4.2 Passage reranking on the MS MARCO set using rerankers

This section explains how the reranked passages could be acquired to carry out the step 2 of the experiment. For this section, the rerankers discussed in the section 2.2 are being used. The passages from the MS MARCO passage dataset are retrieved using BM25 and the results are further reranked using the ranking models such as MonoT5, ANCE and COLBERT. These ranking models were chosen because they represent paradigms such as lexical models, neural re-ranking, single-representation dense retrieval, multi-representation dense retrieval.

These reranked passages can be retrieved using the following pyterrier [8] pipelines:

---

dataset = pt.get_dataset('irds:msmarco-passage/dev')

bm25 = pt.TerrierRetrieve.from_dataset('msmarco-passage', 'terrier_stemmed', wmodel='BM25')

monoT5 = bm25 >> pt.text.get_text(dataset, 'text') >> MonoT5ReRanker()

ance = bm25 >> pt.text.get_text(dataset, 'text') >> pyterrier_ance.ANCETextScorer("msmarco-firstp-checkpoint")

colbert = bm25 >> pt.text.get_text(dataset, 'text') >> pyterrier_colbert.ranking.ColBERTFactory("http://www.dcs.gla.ac.uk/ craigm/colbert.dnn.zip", None, None).text_scorer()

Upon executing the above pipelines, the reranked results for the aforementioned rerankers could be obtained.

Chapter 5

Results and discussion

5.1 Calculation of MRR scores for different rerankers

This section utilises the subsets of datasets created in the section 4.1 and the reranked results retrieved from the section 4.2 to calculate the evaluation metric MRR@10 for different rerankers.

The evaluation metric MRR@10 is the mean reciprocal rank at rank 10. It is calculated by averaging the reciprocal ranks of the relevant documents till rank 10. As it is essential for the relevant documents to be highly ranked, the rank 10 is chosen. The calculation of MRR@10 is done with the help of the pyterrier evaluate function **pt.Utils.evaluate()** which calculates the MRR score by accepting the reranked results from the rankers, the subsets of the queries from the development set, and the choice of evaluation metric(MRR@10) [8].

The findings are noted in the section A. A similar pattern is observed across all rankers as the size of subset is decreased. The value of MRR@10 is highest at rank 10 across all the ranking models.

5.2 Comparison of the MRR scores for different rerankers

In this section, the MRR scores calculated for the different ranking models in the section 5.1 are plotted together to gain an insight into the value of MRR across different rankers.

The findings of this experiment are listed below:

1. The MRR@10 score follows a very similar pattern across the various rankers as noted in A. The MRR@10 value is the least when all the queries of the development set are considered. This behavior can be ascribed to the presence of unanswered queries in the development set, which brings down the average MRR score across all the query.
2. The MRR scores spike when only the subset of queries with a relevant answer in the top 10 documents is considered. This increase in MRR denotes the extent to which survivorship bias impacts the dataset. The section 3.3 points that 99.982% queries in the development set

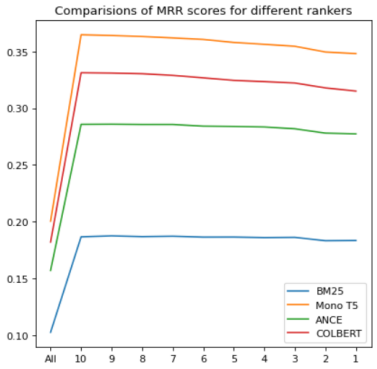


Figure 5.1: MRR scores with different rankers

are answered by the top 10 documents. This subset essentially contains only the answered queries, causing the MRR score to rise up.

3. The MRR scores across all rankers have a general pattern of decrease. One possible explanation of this behavior is that as the criteria for subset selection becomes more stringent, the queries with answers at the higher ranks are lost.

4. The queries with relevance level of 1 are the queries that are answered by the top retrieved document. Such queries are assumed to be simpler in nature and thus the relevant document for these queries are expected to be ranked higher when the reranked using various rankers. Figure 5.1 controverts this assumption as the MRR score does not increase as the subsets with lower ranks are checked against the reranked results.

5. Lastly, from the figure 5.1, it can be observed that the MonoT5 ranker performs the best amongst all the rankers while BM25 performs the worst. MonoT5 makes use of contextualized language models to identify complicated correlations between the query and passage, resulting in it's performance being the best amongst all rankers.

5.3 Implications of the results

An important point to note here is that section 5.1 calculates MRR by finding the ranks at which the qrels are being placed by the rerankers. But there are nuances to consider with qrels. The qrels marked by the annotator is not necessarily the best answer or the only answer for the query. Studies have shown that top passages retrieved upon reranking can sometimes give a superior answer for the query [1]. This finding is also seen in the section 3.5 where out of the 50 unanswered queries, 34 could be answered after reranking the passages using ANCE ranking model.

It can be inferred that ranking of the passages and the rank threshold can play a major role in the number of surviving queries. From the figure 5.1, it can be observed that the MRR score boosts up when only the queries that have relevant documents in the top 10 ranks are considered. The MRR scores for rank thresholds lower than 10 remains more or less the same.

The existence of rank bias can also affect the results of the studies that use MS MARCO for training and evaluation.

Chapter 6

Conclusion and suggestions for future work

6.1 Conclusion

This dissertation had two primary objectives: Firstly, to examine the representative sparse dataset MS MARCO and perform in-depth analysis to derive patterns in answered and unanswered queries. Secondly, to investigate the effect of survivorship bias on the MS MARCO dataset.

During data analysis, it is observed that 45% of the total queries MS MARCO development set are unanswered, which can be problematic for a dataset which aims on question-answering. Upon further analysis of the unanswered queries, it is found that queries of type ”DESCRIPTION” and ”NUMERICAL” have the highest chance of being unanswered. All answered queries in the development set have relevant document within the top 21 documents and 99.98% of the answered queries are answerable in the top 10. It is also observed that the type of query does not affect the rank at which the relevant document would be present.

Upon manual analysis of 150 selected queries from the training and the development set, it is seen that although some unanswered queries are a result of annotator mistake, 71% of the queries could not be answered even partially by the retrieved passages. To check whether reranking the passages would help in answering the query, 50 unanswered queries were taken and the passages were reranked using the ranking models like BM25, MonoT5, ANCE and COLBERT. After analysing the reranked passages, it is observed that 74% of the queries could be answered after reranking (ANCE). Two factors contribute to the answerability of a query : **the ranking of the passages** and **the rank threshold**. This finding is further extended by performing an experiment that finds out how the modulating the rank threshold can affect the MRR scores. For this experiment, subsets are extracted on the basis of the rank threshold and these subsets are tested against the reranked passages.

The results of the experiment display a similar trend in MRR@10 values across all rankers. The MRR score value is the least when the complete development set is considered, the value spikes as the subset of queries answerable in the top 10 documents are considered. As the relevance levels are decreased from 10 to 1, the MRR@10 score slightly decreases further. Out of all the rankers considered, MonoT5 performs the best with the highest value of MRR@10 across all the subsets.

6.2 Suggestions for future work

This dissertation explores the MS MARCO Question answering dataset in-depth to procure knowledge about the queries and their answerability. There are three suggestions for future work:

1. By training a classifier on the training set (focusing on the query text and the relevance judgements) and testing it against the development set. This classifier can be further gauged against the evaluation dataset(the dataset does not contain relevance judgements) to estimate how many queries of the evaluation set can be answered.
2. By employing a metric which measures the sentence complexity, more information can be drawn from the queries being ranked higher by the MS MARCO passage ranking dataset. This can be achieved by either computing the number of tokens in each sentences to get a general estimate about the queries, or by using measures such as **Type-Token Ration(TTR)** [3] to evaluate the lexical diversity of query text.
3. The rank filtered versions of the dataset that are derived in the section 4.1 can be used to further train NIR models. This can help in finding out the effect of rank-filtering of the dataset on the training process of NIR models.

6.3 Personal reflection

This dissertation has been a great learning experience. The topic of this dissertation falls under the domain of Information retrieval, and it provides deeper insights into the concepts of document retrieval, passage reranking, and query-document relevance. The earlier chapters of the dissertation are centred around analyzing the MS MARCO dataset, a dataset based on real-time queries of a search engine. Studying the patterns of this data has given intuitions about the real-world text and the issues encountered when dealing with it.

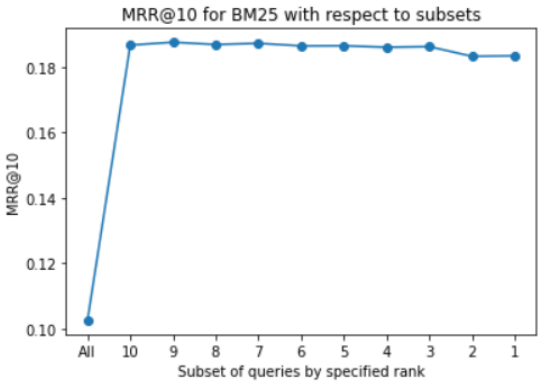
This dissertation has also facilitated a deeper understanding of biases in data science and how these biases can impact other studies. This study also helped in gaining a better grasp on Google Colab and dealing with technical issues when working with large datasets.

Appendix A

Calculation of MRR@10 for different rankers across various subsets

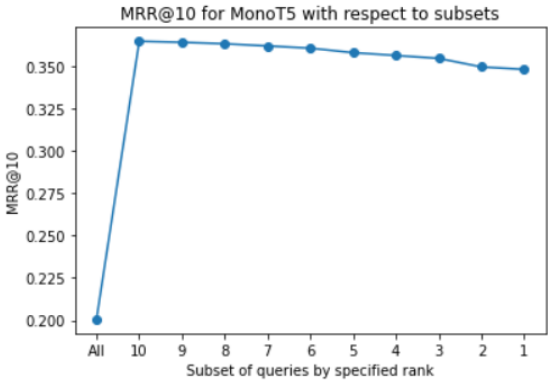
A.1 BM25

	ranking function	rank	MRR score	No. of queries
0	BM25	All	0.10263	101093
1	BM25	10	0.18670	55568
2	BM25	9	0.18758	51344
3	BM25	8	0.18688	47146
4	BM25	7	0.18725	42669
5	BM25	6	0.18646	38024
6	BM25	5	0.18651	33062
7	BM25	4	0.18605	27858
8	BM25	3	0.18625	22060
9	BM25	2	0.18327	15543
10	BM25	1	0.18344	8197



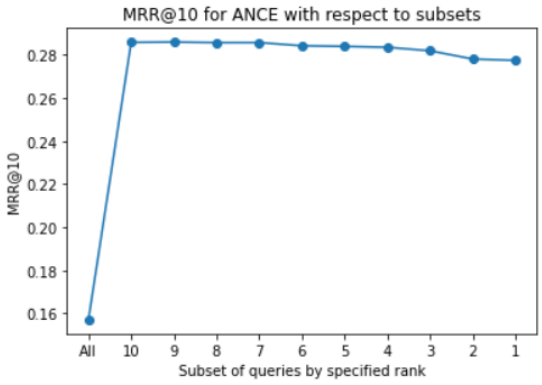
A.2 MonoT5

	ranking function	rank	MRR score	No. of queries
0	Mono T5	All	0.20049	101093
1	Mono T5	10	0.36467	55568
2	Mono T5	9	0.36404	51344
3	Mono T5	8	0.36316	47146
4	Mono T5	7	0.36189	42669
5	Mono T5	6	0.36058	38024
6	Mono T5	5	0.35792	33062
7	Mono T5	4	0.35627	27858
8	Mono T5	3	0.35454	22060
9	Mono T5	2	0.34951	15543
10	Mono T5	1	0.34806	8197



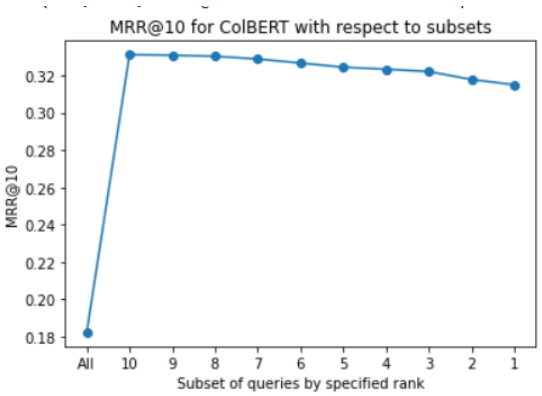
A.3 ANCE

	ranking function	rank	MRR score	No. of queries
0	ANCE	All	0.15711	101093
1	ANCE	10	0.28577	55568
2	ANCE	9	0.28591	51344
3	ANCE	8	0.28566	47146
4	ANCE	7	0.28567	42669
5	ANCE	6	0.28419	38024
6	ANCE	5	0.28391	33062
7	ANCE	4	0.28348	27858
8	ANCE	3	0.28186	22060
9	ANCE	2	0.27805	15543
10	ANCE	1	0.27741	8197



A.4 COLBERT

	ranking function	rank	MRR score	No. of queries
0	COLBERT	All	0.18210	101093
1	COLBERT	10	0.33124	55568
2	COLBERT	9	0.33095	51344
3	COLBERT	8	0.33035	47146
4	COLBERT	7	0.32892	42669
5	COLBERT	6	0.32674	38024
6	COLBERT	5	0.32452	33062
7	COLBERT	4	0.32340	27858
8	COLBERT	3	0.32220	22060
9	COLBERT	2	0.31789	15543
10	COLBERT	1	0.31509	8197



Bibliography

[1] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles L. A. Clarke. Shallow pooling for sparse labels, 2021.

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

[3] Gerasimos Fergadiotis, Heather Wright, and Samuel Green. Psychometric evaluation of lexical diversity indices: Assessing length effects. *Journal of speech, language, and hearing research : JSLHR*, 58, 03 2015.

[4] Patricia Katopol. Maybe best practices aren't: How survivorship bias skews information gathering and decision-making. *Library Leadership & Management*, 32(1).

[5] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert, 2020.

[6] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 08 2019.

[7] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond, 2021.

[8] Craig Macdonald and Nicola Tonellotto. Declarative experimentation in information retrieval using pyterrier. *CoRR*, abs/2007.14271, 2020.

[9] Marc Mangel and F.J. Samaniego. Abraham wald's work on aircraft survivability. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 79:259–267, 06 1984.

[10] Microsoft. Ms marco passage dataset version 2. `https://ir-datasets.com/msmarco-passage-v2.html`.

[11] Microsoft. Ms marco question answering dataset. `https://microsoft.github.io/MSMARCO-Question-Answering/`.

[12] Microsoft. Msmarco-passage-ranking. `https://github.com/microsoft/MSMARCO-Passage-Ranking`.

[13] Katy Milkman. The perils of "survivorship bias". https://www.scientificamerican.com/article/the-perils-of-survivorship-bias/#.

[14] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert, 2020.

[15] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models, 2021.

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[17] Stephen Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33:294–304, 12 1977.

[18] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389, 01 2009.

[19] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval, 2020.