Leipzig University
Institute of Computer Science
Degree Programme Data Science M.Sc.

Implicit Evaluation of Health Answers from Large Language Models

Master's Thesis

Jonas Probst
Born Sept 10, 1995 in Stuttgart

Matriculation Number 3466651

1. Referee: Prof. Dr. Martin Potthast
2. Referee: Dr. Harrisen Scells

Submission date: January 16, 2024

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, January 16, 2024

. . . .
Jonas Probst

Abstract

With the release of ChatGPT, open-ended generation of text became the biggest use case of Large Language Models (LLMs). Meanwhile, LLM evaluation focuses on classical NLP tasks like single-choice question answering or text classification, which do not represent the LLMs' capabilities in long-form question answering (LFQA). The lack of evaluation of open-ended questions is especially concerning in the medical domain, as answers that are misleading or wrong could have significant impact on the users personal health. Using human experts to compare generated answers is considered the gold standard in this space, but it leads to high costs and slower evaluation procedures, while also introducing subjectivity to the evaluations. In this thesis, we present a retrieval-based implicit evaluation method for LFQA, aiming to make the evaluation process faster, cheaper and more repeatable. Using a dataset of queries and associated documents, which were evaluated by human annotators, we first compare multiple retrieval methods for retrieving documents that are relevant, readable, and credible. We then use the most effective retrieval method to rank new answers generated by LLMs against the web answers from the dataset. Because the retrieval method is previously evaluated to produce rankings similar to the human evaluations, we assume that it ranks the generated LLM answer close to where a human would rank it. Our findings demonstrate that the proposed retrieval-based implicit evaluation ranks the effectiveness of various LLMs in a similar order as other benchmarks, underscoring the validity of the approach. Additionally, we show that the LLM ranking improves with model size and with more sophisticated prompting strategies, which aligns with trends observed in the literature. This work is a first step towards building a more automated evaluation framework for LFQA, which could decrease development costs and ensure comparability of different LLMs, even if they are not evaluated by the same research group. Because the most effective model in our research (ChatGPT) already ranks best on nearly every query, we encourage future research to produce a more challenging dataset, enabling the comparison of more advanced models.

Contents

Chapter 1

Related Work

The work presented in this thesis is based on two main areas of research: the evaluation of Large Language Models (LLMs) and Information Retrieval (IR). In this chapter, the current state of research in those areas is presented, as it relates to this thesis. We start with a short introduction to LLMs, then present the current evaluation methods for those models in the field of question answering, since the different versions of question-answering tasks, especially long-form question answering are most similar to our evaluation setup.

Afterwards, the field of IR is introduced. Different retrieval methods used in this thesis are presented and why they were chosen for this work. Additionally, the evaluation metrics used to compare those retrieval methods are introduced.

1.1 Evaluation of LLMs for Question Answering

Transformer-based language models are generally defined as systems that produce probability distributions over a set of tokens (which can be words, subwords, or characters) given the preceding or surrounding context. The rise of LLMs started with the introduction of the transformer architecture by Vaswani et al. (2017), followed by the release of models like BERT (Devlin et al. (2018)) and GPT-2 (Radford et al. (2018)). The transformer architecture allowed the models to process more context than previous models, such as LSTM-based methods like ELMo (Peters et al. (2018)), or static word embeddings based on statistical co-occurrences like GloVe (Pennington et al. (2014)). This led to improvements in many NLP tasks over earlier methods, as shown by Radford et al. (2018), who compare the base GPT performance against multiple then state-of-the-art models on different tasks, improving or performing at least competitively on all of them.

With the release of GPT-3 (Brown et al. (2020)), the size of datasets used to train LLMs, as well as the number of parameters in the models increased significantly. While GPT-2 in its largest version has a total of 1.5 billion parameters, GPT-3 has 175 billion parameters. As Wei et al. (2022) show, this scale not only leads to improvements over previously used benchmarks compared to smaller models but also to what they call *emergent abilities. Emergent abilities* are abilities that are not present in smaller models. Those include generating long coherent stories or poems, translating between languages, and answering long-form questions. None of those capabilities are present in smaller models, with e.g. the largest version of GPT-2 performing poorly on translation and summarization, as shown in the original paper (Radford et al. (2018)).

With the capabilities of LLMs expanding, the task of evaluating them becomes more challenging.

In this section, we focus on the current evaluation methods of question-answering (QA) capabilities of LLMs and how the task of LFQA is evaluated.

The field of QA in NLP contains multiple different tasks, which can be grouped into extractive QA, single/multiple choice QA, and long-form QA. Those tasks vary in their complexity, with earlier models like BERT and GPT-2 already being benchmarked on extractive QA and single-choice QA.

Extractive QA and single/multiple choice QA have in common that they are relatively easy to evaluate. When evaluating extractive QA the overlap of the predicted tokens with the answer span can be calculated, and for single-choice QA the predicted answer option can be compared to the ground truth answer.

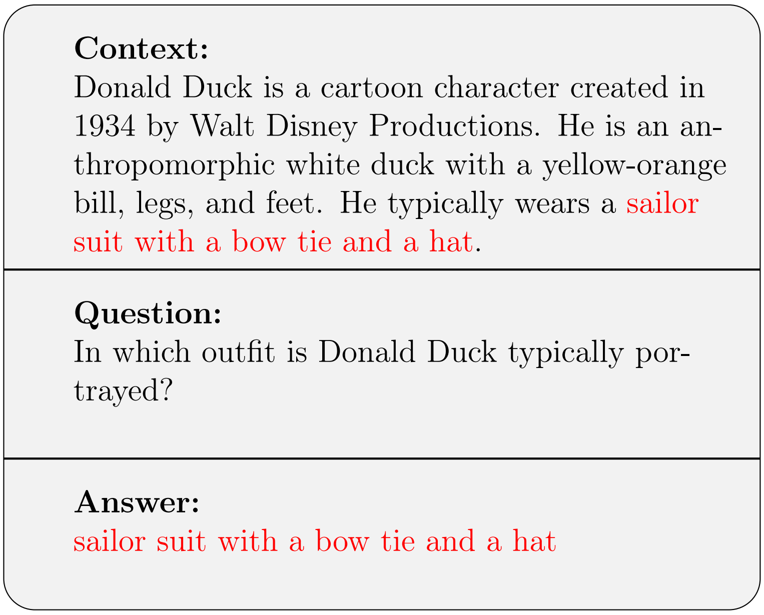
This changes when evaluating long-form QA. Here, models are evaluated on their ability to answer questions in a free-form manner, without constraining the length of the answer. Answers can get long and complex, branching out in different directions by including examples or other information. The evaluation of such answers is not as straightforward as for the other versions of QA, so human evaluation is the gold standard.

In the following, the different QA evaluation tasks are introduced, including popular datasets and evaluation metrics for them.

1.1.1 Extractive Question Answering

Extractive QA is the task of answering questions given a context containing the answer. In this context, which can be a short paragraph or an entire Wikipedia article, the correct answer span has to be selected by the model.

One of the most popular datasets for evaluating LLMs in this task is SQuAD (Rajpurkar et al. (2016)), and its successor SQuAD 2.0 (Rajpurkar et al. (2018)), which includes unanswerable questions. Adding unanswerable questions to the dataset is a way to test the model's ability to detect when a question can not be answered by the given context. Many other datasets like



Context:
Donald Duck is a cartoon character created in 1934 by Walt Disney Productions. He is an anthropomorphic white duck with a yellow-orange bill, legs, and feet. He typically wears a sailor suit with a bow tie and a hat.

Question:
In which outfit is Donald Duck typically portrayed?

Answer:
sailor suit with a bow tie and a hat

Figure 1.1: Example of a typical extractive question answering task: The answer span "sailor suit with a bow tie and a hat" is highlighted in red in the paragraph.

NarrativeQA (Kočiskỳ et al. (2018)), QuAC (Choi et al. (2018)) or Natural Questions (Kwiatkowski et al. (2019)) are based on the same principle. They consist of questions written by crowd workers or experts in the field, based on a Wikipedia article snippet or similar text passages. The exact constraints on the questions and the context vary between the datasets, but the general idea is the same. Figure 1.1 shows a generic example of a question from an extractive QA dataset.

The evaluation metrics for tasks of this category are based on the overlap between the predicted answer span and the ground truth answer span. Specifically, this would be the exact match (EM) score, which measures the percentage of exact matches between the predicted and the ground truth answer span. Alternatively, the F1 score as the harmonic mean of precision and recall can be used, with precision being defined as

$$\text{precision} = \frac{\text{number of correct tokens in prediction}}{\text{total number of tokens in the prediction}}$$

and recall as

$$\text{recall} = \frac{\text{number of correct tokens in prediction}}{\text{total number of tokens in the ground truth}}$$

with a correct token being a predicted token that overlaps with the ground truth answer. The F1 score is then calculated as

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

which is the harmonic mean of precision and recall.

Earlier LLMs like BERT have to be specifically fine-tuned for the task of extractive QA. For each token in the provided context, the model assigns a probability of the token being the starting or ending token of the answer span (Devlin et al. (2018)).

Later models like GPT-3 can directly generate the answer from the context and the questions without any fine-tuning, by using the zero-shot, single-shot, or multi-shot capabilities of the model (Brown et al. (2020)). In some settings of the datasets, the context can be completely omitted, forcing the model to directly answer the question. This means that the results of the two approaches are not directly comparable, because even though the models were evaluated on the same dataset, the approaches are fundamentally different.

1.1.2 Single and Multiple Choice Question Answering

For single and multiple choice question answering, the model has to select the correct answer from a set of possible answers, which can be done with or without context. While most datasets are single-choice datasets, some datasets (like MultiRC by Khashabi et al. (2018)) are multiple-choice so that the model has to check each answer for correctness, instead of just selecting the best fitting one. Questions for single and multiple choice QA often stem from official exams, like the MMLU dataset (Hendrycks et al. (2020)), which combines questions from many exams like the United States Medical Licensing Examination or the Examination for Professional Practice in Psychology. In other datasets, the questions are collected from crowd workers and verified by experts (Clark et al. (2018), Mihaylov et al. (2018)).

Figure 1.2 shows an example of a single-choice question with context. The LLM is provided with the context, the question and the answer options, and a “Answer: ” prefix. Based on this, the model has to select one of the given options.

Evaluation is straightforward in this case, the model’s generated answer option is compared to the ground truth answer, and accuracy over all questions is calculated.

1.1.3 Long Form Question Answering

Long-form QA refers to the task of answering open-ended questions, which can usually not be answered by simply providing one entity or number, but

Context:
Donald Duck is a cartoon character created in 1934 by Walt Disney Productions. He is an anthropomorphic white duck with a yellow-orange bill, legs, and feet. He typically wears a sailor suit with a bow tie and a hat.

Question:
What does Donald Duck typically wear?

(A) T-shirt and jeans
(B) Sailor suit with a bow tie and a hat
(C) Business suit
(D) Basketball jersey

Answer:
B

Figure 1.2: Example of a single-choice question with context: The correct option is highlighted in red. The blue B would be a possible answer generated by the LLM, after being primed with the context and the question.

require an in-depth answer. With LLMs being deployed in chatbots and as such expected to deliver answers mostly without context, evaluating the models on long-form QA is important.

So far, only a handful of datasets are available for this task, with the first dataset in this category being the ELI5 dataset (Fan et al. (2019)). It consists of questions and the corresponding highest-voted answer from the "Explain Like I'm Five" subreddit, where users ask questions about complex topics, which are then answered by other users. They are accompanied by support documents, which are retrieved from web sources by querying for the original question. This dataset was used by Nakano et al. (2021) to fine-tune GPT-3 for the task of long-form QA without using the context documents. The answers given by the fine-tuned model are evaluated by humans, by comparing them to the highest voted answer from the ELI5 dataset.

An additional dataset for this task is the MultiMedQA (Singhal et al. (2023)), which curates questions from multiple other datasets used previously. Answers generated by physicians are used as ground truth answers. Those are

Question:
Can you describe the evolution of Donald Duck's character design and attire over the years, focusing on any major changes and the reasons behind them?

Answer:
The evolution of Donald Duck's character design and attire has seen several notable changes since his creation by Walt Disney and Dick Lundy in 1934. Here is an overview of the major changes and the reasons behind them:

$$\left[\vdots\right]$$

Figure 1.3: Example of Long-Form QA without Context: The question prompts for a detailed answer about Donald Duck's character evolution and attire. The prompt given to the model is written in black, and the sample answer (generated by ChatGPT-3.5) is written in blue. The generated answer continues for multiple paragraphs.

then compared to the model-generated answers by other physicians, as well as by laypeople. Additionally, the answers were individually rated in different rubrics, introduced in a previous work (Singhal et al. (2022)).

None of the papers for current, main-stream LLMs like GPT-3 (Brown et al. (2020)), GPT-4 (OpenAI (2023)) or Llama 2 (Touvron et al. (2023)) include evaluations on common benchmarks for this category. This shows that long-form QA is still a relatively new task, lacking sufficient academic benchmarks.

1.1.4 Difficulties of Long Form Question Answering

Recent works have shown multiple challenges in the task of long-form question answering, independently of which model architecture is used to answer the questions. Since the answers are free-form text, and not just a multiple choice option, one number, or one entity, the quality of the model can't be measured using accuracy or similar metrics that require static ground truth information.

Multiple evaluation dimensions are of interest, for example:

- **Relevance:** Are the most important aspects of the query answered?
- **Readability:** How easy is the answer to read and understand?
- **Credibility:** Are there any references provided and are they of high quality?

This adds additional complexity to the evaluation process, compared to other QA tasks which only measure the correctness of the answer.

Xu et al. (2023) focus on the evaluation process of LFQA, comparing different automatic evaluation methods to human judgment. They differentiate between general-purpose generation evaluation metrics, which were originally designed for other NLP tasks like summarization or translation, and fine-tuned metrics, which are fine-tuned to LFQA. The following types of metrics are considered general-purpose metrics, used in the same way in different NLP tasks:

- **Answer-reference metrics:** Include metrics like ROUGE or BERTScore. These metrics compare generated answers to reference answers, focusing on aspects like lexical overlap and semantic similarity.
- **Answer-only metrics:** Such as Self-BLEU which measures the fluency and diversity of generated text. These are intrinsic metrics of the generated text and do not need a reference answer for evaluation.
- **Question-answer metrics:** Score answers given the question in one of two ways: either by calculating the likelihood of questions given an answer, or by using an encoder model to score sequences given a prefix.
- **Answer-evidence metrics:** Judge the given answer by the evidence documents used to generate it. This method indirectly assesses the answer's credibility and factual accuracy.

In addition to these metrics, Xu et al. (2023) evaluate two different versions of fine-tuned metrics. The first one is based on Longformer (Beltagy et al. (2020)), in which the model is fine-tuned to produce a score given a question and an answer, optionally combined with evidence documents. The second model is a fine-tuned version of GPT-3, which is trained to output either *Answer1* or *Answer2* given a question and two answer options. Both fine-tuned models are trained on the dataset generated by Nakano et al. (2021), which contains human preference labels for different answer pairs.

All automatic evaluation methods are evaluated on the task of choosing the preferable answer given two long-form answers to a question. The results are compared to previous human judgment on the same task. They find that one of their baseline models, choosing always the longest answer, performs almost as well as the fine-tuned GPT-3 model, which outperformed all other methods. Both variants are still outperformed by human annotations, which is the gold standard.

Krishna et al. (2021) investigates in more detail the problems of reference-based evaluation metrics like ROUGE-L. They highlight that those metrics are unable to capture answer components like examples if those examples are not present in the ground truth answers.

Furthermore, they note that even human evaluation is limited in judging long-form QA over different models. Some problems include the hiring process of experts, especially when datasets tackle multiple fields of expertise. Finding experts of similar education and background is challenging when doing evaluations of different models over time. Additionally, the evaluation process is more mentally demanding for the individual annotator the longer the answers get. Similar problems have previously been shown by Akoury et al. (2020) in the context of machine-generated stories. They find that crowd workers have low agreement for different evaluation metrics when evaluating the same stories. They tackle this problem by using gamification techniques to activate online users of a story-writing platform to evaluate and improve the generated stories. A similar approach is taken by Dugan et al. (2020), who implement a website where users try to differentiate machine-generated text from human-generated text.

This shows that there are still many obstacles to overcome in the task of evaluating long-form QA systems. We try to tackle some of those problems with the approach presented in this thesis. To incorporate multiple metrics, we first compare and evaluate multiple retrieval methods on the previously mentioned metrics of relevance, readability, and credibility. This ensures that the document ranking is optimized for multiple dimensions, instead of just one. Based on the findings by Xu et al. (2023) which shows that of the automated metrics the transformer-based models perform best, we compare multiple transformer-based models for the retrieval step.

Since our approach to tackling this problem uses concepts from the field of information retrieval, a short overview of relevant methods is given in the next section.

1.2 Retrieval Models

Since we want to use retrieval methods to evaluate the performance of LLMs, we will give some background on the field of Information Retrieval(IR), and

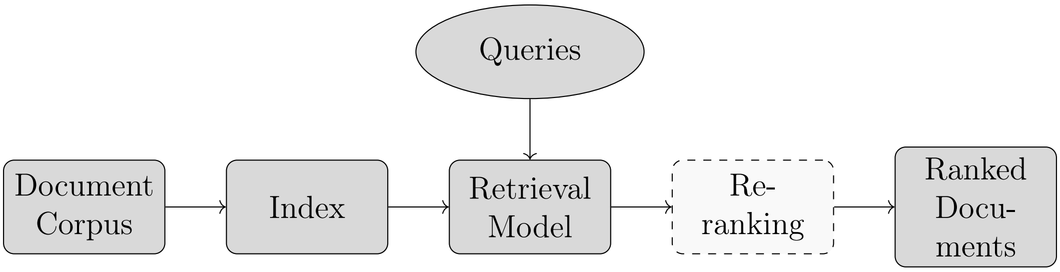


Figure 1.4: Depiction of a basic retrieval pipeline, with optional re-ranking step. The process begins with a document corpus, which is indexed to facilitate efficient retrieval. Given the information needed by a user in the form of a query, the retrieval model returns the most relevant documents. Optionally, the documents can be re-ranked using a re-ranking model which is usually more resource-intensive.

how it relates to long-form QA. IR is the process of retrieving relevant information based on an information need, from a collection of documents, usually in the form of whole documents, passages, or single sentences. A brief overview of a retrieval pipeline based on “An introduction to information retrieval” by Manning (2009) is provided now.

In most IR systems, the document corpus first has to be brought into a form that is more suitable for retrieval. This process usually starts with a step to reduce to total size of the vocabulary, using tokenization, stemming, stop word removal, and other techniques. This removes unnecessary information from the documents, which would otherwise increase the size of the index, which is created in the next step. Commonly, this is an inverted index, which maps each term in the corpus to the documents that contain it. Now, given a query, an IR system returns a ranked list of documents that are most relevant to the query. To achieve this, IR systems estimate a relevance score for each document in the collection with respect to the query. The documents are then ranked according to their relevance scores, with the most relevant documents appearing at the top of the list. Some pipelines include a re-ranking step, in which the most relevant documents are re-ranked after the first retrieval using a more expensive (and hopefully more effective) retrieval model. Figure 1.4 shows a basic retrieval pipeline, in which the re-ranking step is optional.

1.2.1 Baseline Retrieval Models

First, we will look at the basic retrieval models, which are used as baselines in this thesis. They have been chosen because they were shown to be effective on the used dataset in previous work (Goeuriot et al. (2021)). Both models use the implementation in the Terrier IR platform (Ounis et al. (2005)).

TF-IDF

Term Frequency-Inverse Document Frequency (tf-idf) is one of the most commonly used models in information retrieval. It measures the importance of a term in a document relative to a collection of documents or corpus. The central intuition is that terms that appear frequently in a document but not in many other documents in the corpus are significant and thus should be given higher weight.

Given a query q consisting of terms $t_1, t_2, \cdots, t_n$ and a document corpus D of size N we calculate the score of q given document d by first calculating the tf-idf of each term in q given d and then summing them up.

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t)$$

Where $\text{tf}(t, d)$ is the frequency of term t in document d and

$$\text{idf}(t) = \log\left(\frac{N}{1 + \text{df}(t)}\right)$$

where document frequency $\text{df}(t)$ is the number of documents containing term t. The final score of the query given the document is then calculated as

$$\text{score}(q, d) = \sum_{t \in q} \text{tf-idf}(t, d)$$

For retrieval given a query q tf-idf scores are calculated for all documents, which are then ranked according to their score.

In the Terrier IR platform, the implementation of tf-idf uses variants of the tf and idf components. For Term Frequency, Robertson's tf formulation (Robertson (2004)) is used, which incorporates an additional parameter that adds a saturation effect to the term frequency. For idf, the original formulation by Sparck Jones (Sparck Jones (1972)) is applied.

DPH

Divergence from Randomness (DFR) is a framework in IR that assigns term weights based on the divergence of the actual within-document term frequency distribution from a random term frequency distribution (Amati (2006)). The divergence from randomness using the hyper-geometric distribution model(DPH) is one of the models derived from the DFR framework.

The principle behind DFR is that terms that are informative in a document will have a distribution that deviates significantly from what would be expected if terms were distributed randomly. While TF-IDF emphasizes the importance

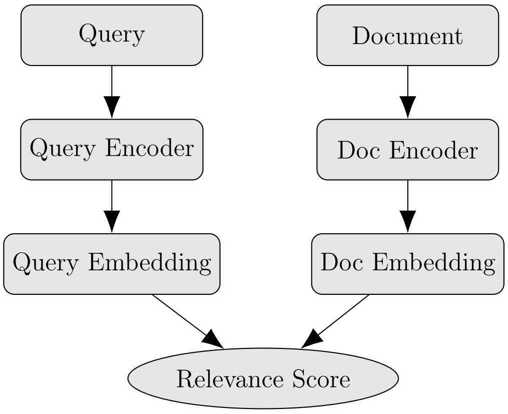


Figure 1.5: Bi-Encoder Architecture. Query and document are encoded separately, then the relevance is calculated based on similarity measures between the embeddings.

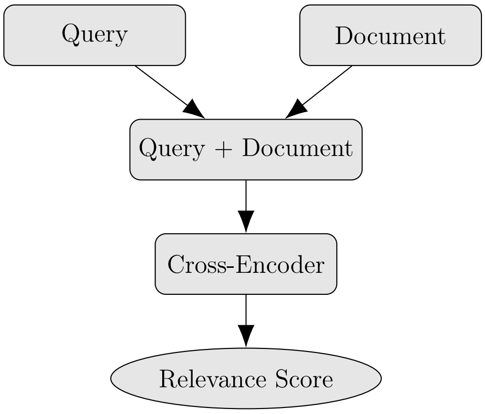


Figure 1.6: Cross-Encoder Architecture. Query and document are first combined and then fed to the Cross-Encoder, the relevance is directly calculated by the encoder.

of terms based on their frequency in a document and their inverse frequency in the corpus, DPH focuses on the divergence of a term's distribution from what would be expected under a random distribution. Specifically, DPH assesses the divergence using the hyper-geometric distribution. In essence, where TF-IDF weights terms based on their prominence and rarity, DPH weights them based on how much their occurrence pattern deviates from randomness. The implementation in the Terrier IR platform follows the original formulation by Amati (2006).

1.2.2 Transformer-based Retrieval Models

As mentioned in earlier sections, transformer-based models have been shown to be effective on a variety of tasks, including IR. There are different approaches to how transformers can be used for retrieval, a brief overview of methods used in this thesis is given here. All transformer models are a form of learning-to-rank, in which the model is trained on a dataset to predict which documents are relevant to a query. This differentiates them from our baseline models, which are rule-based and do not require training.

Two of the most common transformer-based architectures are cross-encoder and bi-encoder models. Bi-encoders independently embed queries and documents using transformer models like BERT (Devlin et al. (2018)) or T5 (Roberts et al. (2019)). For the document collection, this can be done offline since the embeddings are independent of the query. This separation allows them to efficiently process large datasets as the embeddings can be pre-computed and stored. It also enables rapid retrieval, since at inference time only the embedding for the current query has to be calculated and compared to the precomputed document embeddings. The documents are then ranked according to their embedding similarity to the query.

Cross-encoders, on the other hand, take a combined input sequence of both query and document and produce a scalar relevance score. This joint modeling enables them to better capture the interaction and nuances between a query and a document. Due to their fine-grained interaction modeling, cross-encoders outperform bi-encoders in terms of precision in retrieving relevant documents as shown by Thakur et al. (2020) and Rosa et al. (2022). However, the need to process each query-document pair individually makes them computationally demanding, especially for large datasets. As a result, they are typically only used in second-stage retrieval, where the objective is to re-rank the top results obtained from an initial retrieval method.

The following sections describe the different transformer-based retrieval models used in this thesis.

monoT5 + duoT5

MonoT5 and duoT5 (Roberts et al. (2019)), are based on previous work by Nogueira et al. (2019), who introduced monoBERT and duoBERT, first applying transformer architecture to the task of document ranking. The general idea is to first use a baseline retrieval model like BM25 to retrieve an initial set of relevant documents. Then, pairs of the query and each document in the initial set are concatenated and fed into the mono version of the transformer model, which produces a scalar score for each query-document pair. The documents with the highest scores are then fed into the duo model, which takes the query and each possible pair of documents as input, outputting a probability of one document being more relevant than the other. Given those scores, the documents are re-ranked another time, serving as the final output of the retrieval pipeline. The main difference between the BERT and T5 versions is that for BERT the [CLS] token from embedding the query and the document can be used as input to a single-layer neural network, to output a probability of the document being relevant. Since there is no [CLS] token for models in the T5 family, as they are sequence-to-sequence models, this part is done using an input template:

$$\text{Query: } q \text{ Document: } d \text{ Relevant:} \tag{1.1}$$

where the model is fine-tuned to produce the token *true* or *false* given query q and document d.

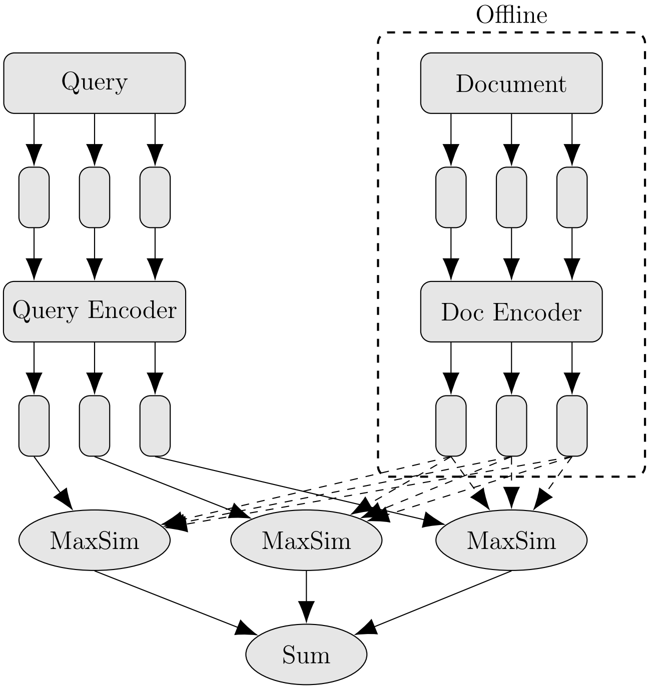


Figure 1.7: ColBERT Retrieval Model. The query and document are processed separately to generate token embeddings. The maximum cosine similarity between all token embeddings of the query and document is calculated and summed over all query tokens, which is the relevance score of the document given the query.

At inference, softmax is applied to the *true* and *false* tokens only, the scores are then calculated using the probability of the *true* token. This works analogously for the Duo version of both models, by just adding the second document. MonoT5 and duoT5 both belong to the family of cross-encoder models, sharing the general characteristics of those models.

This model architecture allows for precisely tuning retrieval efficiency vs. effectiveness, by parameterizing how many documents are filtered in each step, making the model suitable for different use cases.

ColBERT

ColBERT (Khattab and Zaharia (2020)) is another BERT-based model for document ranking. It shares similarities with the general bi-encoder architecture but introduces some modifications to improve effectiveness. Instead of representing each document and query as a single vector, ColBERT uses a set of vectors to represent contextualized embeddings for each token in the document or query. For the documents, this can again be done offline, so that at the retrieval stage, only the embeddings for the query have to be generated and compared to the documents. The relevance score for document d given query q is estimated using their late interaction model, in which maximum cosine similarity between all query term embeddings and document term embeddings is calculated, and then summed over for all query terms. This allows for a more fine-grained mapping between query and document terms, also allowing for more fine-grained meanings of different terms. An efficient computation of the similarity calculation allows ColBERT to scale much better, compared to feeding BERT the query and each document as input.

The ColBERT architecture can generally be used for re-ranking or end-to-end retrieval. In the second case, an additional step is added in which the complete collection is filtered for relevant documents using similarity search to find documents that contain similar terms as the query. In the second step, the remaining documents are re-ranked using the maximum similarity metrics.

ColBERTv2

ColBERTv2 (Santhanam et al., 2021) is directly based on the late-interaction architecture of ColBERT but adds improvements to the architecture and the training process.

To improve the training process, a new process of generating hard negatives is applied, in which a cross-encoder model is used to first rank passages given a query. From the ranked passages a highly ranked one is selected as the positive passage and a low ranked one as the negative passage.

Improvements to the architecture are made by incorporating a residual compression technique. While for ColBERTv1 each token embedding is saved separately in its original state, ColBERTv2 represents each token embedding as its nearest neighboring centroid embedding and a residual. The centroids are calculated using k-means clustering on the token embeddings of a sample of all passages at indexing time. This allows for a more compact representation of the token embeddings since only the centroid embedding has to be stored.

As shown by the authors, ColBERTv2 outperforms ColBERTv1 on all evaluated datasets, while also being more efficient

1.2.3 Evaluation of Retrieval Models

Evaluation metrics are important for the automatic assessment of retrieval model effectiveness. They provide a quantitative measure of how well a system retrieves relevant documents in response to a user's query. Compared to other tasks like classification or regression, the evaluation of retrieval models is more challenging, since the evaluation metrics are not as intuitive. Instead of a single correct answer, multiple documents could be considered relevant to a query, which can be retrieved in different permutations. To be able to judge different retrieval methods, relevance assessments for document-query pairs have to be defined by human assessors.

There are multiple metrics which can be used in information retrieval tasks:

- **Precision**: Measures the fraction of retrieved documents that are relevant.
- **Recall**: Captures the fraction of relevant documents that are retrieved.
- **F1 Score**: The harmonic mean of precision and recall.
- **Mean Average Precision (MAP)**: The arithmetic mean of each query's average precision at each document position.
- **Normalized Discounted Cumulative Gain (nDCG)**: Considers both the ranking and the relevance grade of retrieved documents, weighing highly relevant documents higher than less relevant ones.

Given the context of our dataset, which contains multiple queries with documents ranked for relevance between 0 and 3, and has relevance ratings for each document in the dataset, the nDCG metric is the most suitable one.

The formula for nDCG@k is as follows is based on the concept of cumulative gain (CG), which is the sum of the relevance ratings of all documents up until position k:

$$\text{CG@k} = \sum_{i=1}^{k} r(d_i, q)$$

where $r(d_i, q)$ is the relevance rating of document d_i for query q. Based on that, the discounted cumulative gain (DCG) is defined as

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{r(d_i, q)}{\log_2(i+1)}$$

where the relevance ratings are discounted by the logarithm of the rank of the document, to weight higher-ranked documents stronger than lower-ranked ones. The ideal DCG (IDCG) is the DCG of the ideal ranking, which is the ranking of documents sorted descending by their relevance rating. This produces the highest possible DCG for a set of retrieved documents. Finally, the normalized DCG is calculated as

$$\text{nDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}$$

which normalizes the DCG by the IDCG, producing a score between 0 and 1. The cutoff value of k is set depending on how many of the ranked documents should be considered for each query.

1.2.4 Using Retrieval Models for Evaluating NLP tasks

In the field of evaluating Natural Language Processing(NLP) tasks, IR techniques have been mainly used to evaluate machine translation systems. Since both question answering and machine translation are sequence-to-sequence tasks with multiple correct answers, there are similarities in the evaluation process.

For machine translation, a model is given a source sentence and has to produce a target sentence in another language. Those systems are usually evaluated given a ground truth reference sentence, which is compared to the predicted sentence. Traditionally, methods like BLEU (Papineni et al. (2002)), ROUGE (Lin (2004)), or METEOR (Banerjee and Lavie (2005)) are used to compare the predicted sentence to one or multiple reference sentences. There are multiple versions of those metrics with slightly different parameters, which mostly differ in what sequences of tokens are compared (1-gram, 2-gram, 3-gram, 4-gram, longest common sub-sequence), how precision and recall are weighted, and how the results are smoothed. In the end, all of those metrics generate a similarity score between the predicted and the reference sentence, which can be used to compare different models.

Alternative approaches incorporating ranking methods have been proposed as well. Duh (2008) argues that, considering the final goal is to compare different translation systems, a direct comparison between the systems is preferable to evaluating their quality individually. He shows that ranking methods like RankSVM (Joachims (2002)) and RankBoost (Freund et al. (2003)) can be applied to rank different candidate translations against the reference, generating similar scores to BLEU on the same feature set. When incorporating intra-set features which can not easily be incorporated into BLEU-like scores, the ranking methods achieve higher similarities to human rankings compared to BLEU and smoothed BLEU.

Another approach based on learning-to-rank methods is proposed by Li et al. (2013). Again, they compare multiple translation candidates to a reference sentence, but here they use listwise learning-to-rank methods to generate a ranking of the candidates. The objective of listwise ranking is to train a ranking function that minimizes the loss between the predicted ranking and the ground truth human-generated ranking on a training set. Similar to Duh (2008), they show that the ranking-based methods correlate stronger with human judgment, compared to BLEU-like metrics.

Guzmán et al. (2019) use neural methods to incorporate syntactic and semantic information into the evaluation process. They train a pairwise ranking model, which compares two candidate sentences given the reference and returns which one is the better translation. Even though they do not outperform other state-of-the-art methods, they deliver competitive results while staying closer to the human evaluation framework.

This overview shows that information retrieval methods have successfully been applied to the task of evaluating machine translation systems. However, those approaches are hard to transfer to the task of evaluating from question-answering systems, since the evaluation metrics are not directly applicable. All ranking-based machine translation evaluation methods mentioned here compare the set of candidate translations to a single reference translation, trying to rank the candidate translations amongst each other. Since the space of correct answers given a long-form question is generally much higher in comparison to translating a sentence, this evaluation setup can not be directly applied here. So, even though information retrieval methods are applied here as well, the evaluation approach is formulated differently.

1.3 Summary

In this chapter, we presented an overview of the current state of research in the field of evaluating large language models for question answering. We highlighted that the evaluation of long-form question answering is challenging since the answers are free-form text and not only a single entity, number, or multiple choice option. Additionally, we show that evaluating long-form answers in one single dimension of correctness is not sufficient, as other aspects like readability and credibility are important as well. In this thesis, we try to tackle both of the mentioned challenges.

Based on the assumption that if we develop a ranking model that can effectively rank human-generated documents in the dimensions of relevance, readability, and credibility in a way that is similar to human judgment, we can use it to evaluate LFQA systems. The evaluation of new documents generated by LLMs can be done by ranking them with the previously validated retrieval model and using the achieved rank of the new document as a proxy for the quality of the generated answer. We formalize this approach as follows:

1. **Dataset acquisition:** Collect a dataset of queries $q_1, q_2, \cdots, q_n$ with associated human-generated documents for each query $d_{i,1}, d_{i,2}, \cdots d_{i,j} \forall i \in \{1, \cdots, n\}$. Each document has a relevance rating $r_{rel}(d_{i,j}, q_i)$, a readability rating $r_{read}(d_{i,j}, q_i)$ and a credibility rating $r_{cred}(d_{i,j}, q_i)$ for the query.

2. **Retrieval Model Evaluation:** Evaluate a set of retrieval models $\mathcal{M}$ on the dataset, using the nDCG metric for relevance, readability and credibility.

3. **Generate LLM Answers:** Use a set LLMs $\mathcal{L}$ to generate answers $a_{l,i}$ for all queries q_i.

4. **Rank Answers:** Add the generated answers $a_{l,i}$ to the documents $d_{1,i}, d_{2,i}, \cdots, d_{n,i}$ for each query q_i and rank them using the best of the previously evaluated retrieval model $\mathcal{M}$.

This approach allows us to evaluate the capabilities of multiple LLMs in long-form QA, by comparing the ranks of the generated answers as a proxy for their quality. By evaluating the retrieval models on multiple evaluation criteria, we tackle the problem of evaluating the answers in multiple dimensions. Furthermore, the diversity of possible answers can be better captured because multiple different answers in the dataset can be ranked highly, instead of only having one reference answer. We hope that the retrieval models can capture this diversity in their ranking, especially when using transformer-based models.

In the following chapter, the experimental setup is described, with the results being presented in chapter **??**.

Bibliography

Akoury, N., Wang, S., Whiting, J., Hood, S., Peng, N., and Iyyer, M. (2020). Storium: A dataset and evaluation platform for machine-in-the-loop story generation. *arXiv preprint arXiv:2010.01717.*

Amati, G. (2006). Frequentist and bayesian approach to information retrieval. In *European Conference on Information Retrieval*, pages 13–24. Springer.

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

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150.*

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *CoRR*, abs/2005.14165.

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., Liang, P., and Zettlemoyer, L. (2018). Quac: Question answering in context. *arXiv preprint arXiv:1808.07036.*

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dugan, L., Ippolito, D., Kirubarajan, A., and Callison-Burch, C. (2020). Roft: A tool for evaluating human detection of machine-generated text. *arXiv preprint arXiv:2010.03070*.

Duh, K. (2008). Ranking vs. regression in machine translation evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 191–194.

Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., and Auli, M. (2019). Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969.

Goeuriot, L., Suominen, H., Pasi, G., Bassani, E., Brew-Sam, N., Sáez, G. N. G., Kelly, L., Mulhem, P., Seneviratne, S., Upadhyay, R., Viviani, M., and Xu, C. (2021). Consumer Health Search at CLEF eHealth 2021.

Guzmán, F., Joty, S., Màrquez, L., and Nakov, P. (2019). Pairwise neural machine translation evaluation. *arXiv preprint arXiv:1912.03135*.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., and Roth, D. (2018). Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Khattab, O. and Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Kočiskỳ, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. (2018). The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Krishna, K., Roy, A., and Iyyer, M. (2021). Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Li, M., Jiang, A., and Wang, M. (2013). Listwise approach to learning to rank for automatic evaluation of machine translation. In *Proceedings of Machine Translation Summit XIV: Papers*.

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

Manning, C. D. (2009). *An introduction to information retrieval*. Cambridge university press.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. (2021). Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Nogueira, R., Yang, W., Cho, K., and Lin, J. (2019). Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

OpenAI (2023). Gpt-4 technical report.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and Johnson, D. (2005). Terrier information retrieval platform. In *Proceedings of the 27th European Conference on Advances in Information Retrieval Research*, ECIR'05, page 517–519, Berlin, Heidelberg. Springer-Verlag.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). 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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Roberts, A., Raffel, C., Lee, K., Matena, M., Shazeer, N., Liu, P. J., Narang, S., Li, W., and Zhou, Y. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520.

Rosa, G., Bonifacio, L., Jeronymo, V., Abonizio, H., Fadaee, M., Lotufo, R., and Nogueira, R. (2022). In defense of cross-encoders for zero-shot retrieval. *arXiv preprint arXiv:2212.06121*.

Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., and Zaharia, M. (2021). Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A. K., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Schärli, N., Chowdhery, A., Mansfield, P. A., y Arcas, B. A., Webster, D. R., Corrado, G. S., Matias, Y., Chou, K., Gottweis, J., Tomasev, N., Liu, Y., Rajkomar, A., Barral, J. K., Semturs, C., Karthikesalingam, A., and Natarajan, V. (2022). Large language models encode clinical knowledge. *CoRR*, abs/2212.13138.

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., Clark, K., Pfohl, S., Cole-Lewis, H., Neal, D., et al. (2023). Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2020). Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

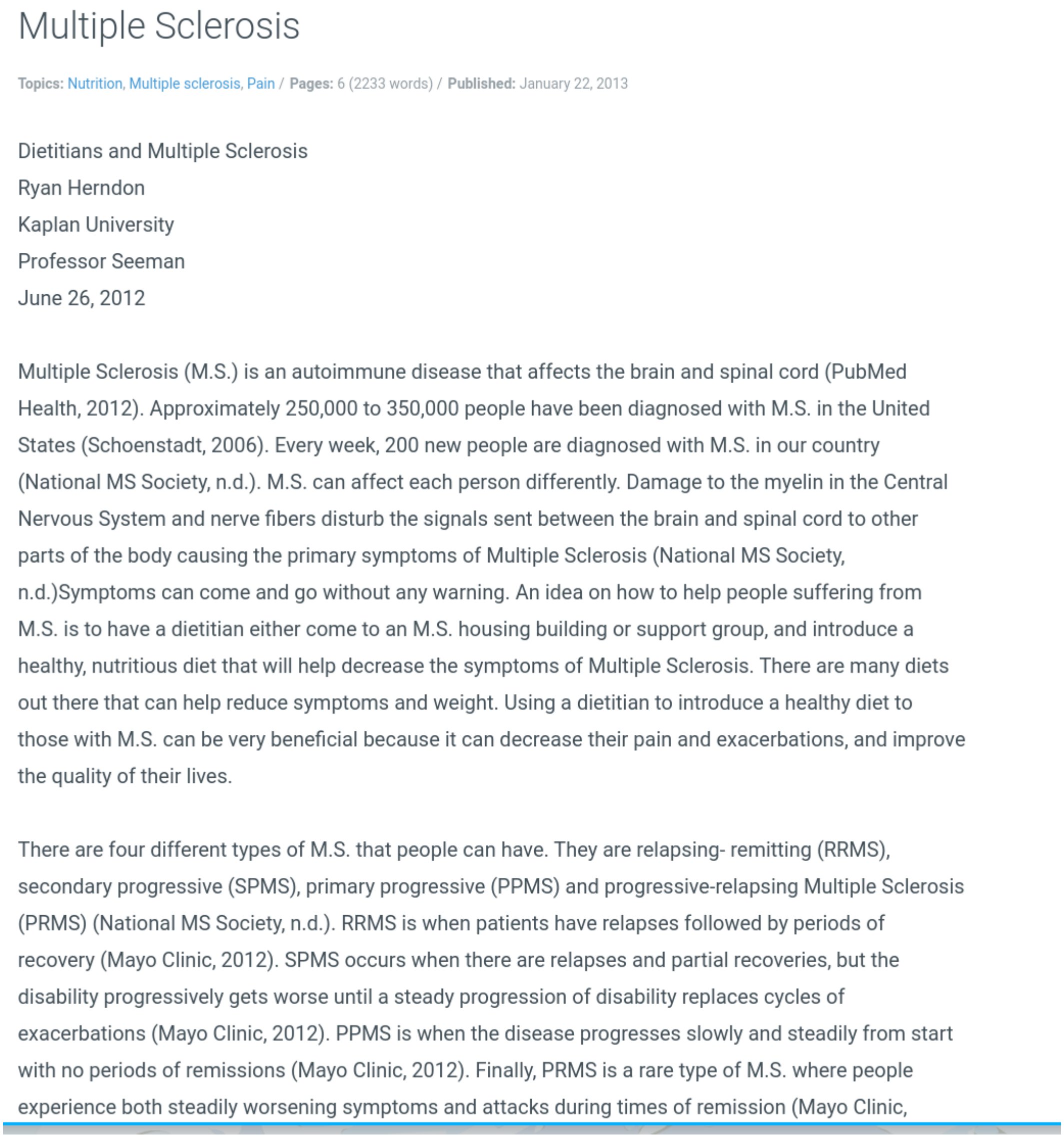
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Xu, F., Song, Y., Iyyer, M., and Choi, E. (2023). A critical evaluation of evaluations for long-form question answering. *arXiv preprint arXiv:2305.18201*.

Appendix A

Dataset



Multiple Sclerosis

Topics: Nutrition, Multiple sclerosis, Pain / Pages: 6 (2233 words) / Published: January 22, 2013

Dietitians and Multiple Sclerosis
Ryan Herndon
Kaplan University
Professor Seeman
June 26, 2012

Multiple Sclerosis (M.S.) is an autoimmune disease that affects the brain and spinal cord (PubMed Health, 2012). Approximately 250,000 to 350,000 people have been diagnosed with M.S. in the United States (Schoenstadt, 2006). Every week, 200 new people are diagnosed with M.S. in our country (National MS Society, n.d.). M.S. can affect each person differently. Damage to the myelin in the Central Nervous System and nerve fibers disturb the signals sent between the brain and spinal cord to other parts of the body causing the primary symptoms of Multiple Sclerosis (National MS Society, n.d.)Symptoms can come and go without any warning. An idea on how to help people suffering from M.S. is to have a dietitian either come to an M.S. housing building or support group, and introduce a healthy, nutritious diet that will help decrease the symptoms of Multiple Sclerosis. There are many diets out there that can help reduce symptoms and weight. Using a dietitian to introduce a healthy diet to those with M.S. can be very beneficial because it can decrease their pain and exacerbations, and improve the quality of their lives.

There are four different types of M.S. that people can have. They are relapsing- remitting (RRMS), secondary progressive (SPMS), primary progressive (PPMS) and progressive-relapsing Multiple Sclerosis (PRMS) (National MS Society, n.d.). RRMS is when patients have relapses followed by periods of recovery (Mayo Clinic, 2012). SPMS occurs when there are relapses and partial recoveries, but the disability progressively gets worse until a steady progression of disability replaces cycles of exacerbations (Mayo Clinic, 2012). PPMS is when the disease progresses slowly and steadily from start with no periods of remissions (Mayo Clinic, 2012). Finally, PRMS is a rare type of M.S. where people experience both steadily worsening symptoms and attacks during times of remission (Mayo Clinic,

Query	Rel	Cred	Read
Covid-19 vaccine & MS drugs	0	1	2
MS transmission to family	0	1	2
Reading issues & MS	0	1	2
Menopause & MS symptoms	0	3	2
Improvement timeline in MS	1	0	1
MS, sleep issues, & forgetfulness	1	1	1
Relapsing-remitting MS	1	1	1
MS development risk	1	1	2
MS fatigue causes	1	2	1
MS symptoms list	1	2	2
Working full-time with MS	1	2	2
Secondary progressive MS	1	3	1
Diagnosing MS relapse	2	1	1
MS impact on career	2	1	2
Managing MS	2	2	2

Figure A.1: Example of a document with multiple ratings for different queries. Queries are shortened for readability.