Utilizing Deep Natural Language Processing to Beat Jeopardy!

Yiqing (Melody) Wang
Department of Computational Biology
Carnegie Mellon University
Pittsburgh, PA 15213
yiqingwa@andrew.cmu.edu

Caroline Springer
Department of Computational Biology
Carnegie Mellon University
Pittsburgh, PA 15213
cjspring@andrew.cmu.edu

Kevin Elaba
Department of Computational Biology
Carnegie Mellon University
Pittsburgh, PA 15213
kelaba@andrew.cmu.edu

1 Introduction

Jeopardy! is a TV quiz show where three contestants compete against one another in a game that requires answering natural language questions over a broad domain of topics. A computer system that could compete at human champion levels at this game needs to produce answers with high precision and speed as well as having a reliable confidence in its answers, such that it could answer roughly 70 percent of the questions asked with greater than 80 percent precision in 3 seconds or less [1]. We aim to improve upon a current method implemented in Haystack, an open-source framework for question answering [2]. The inputs to our system are Jeopardy! questions and Wikipedia article. The the outputs are the answer to the Jeopardy! questions. The performance is analyzed by the F1 score and exact match score of the highest confidence answer to each question.

1.1 The Datasets

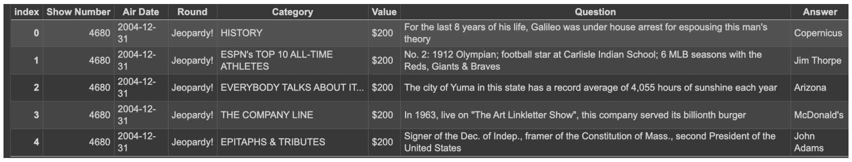
1.1.1 WikiMedia Dumps Dataset

The datasets are built from the Wikipedia dump [3] with one split per language. We initially used the first 50,000 out of about 6.5 million preprocessed English Wikipedia articles for our preliminary midway results. However, the dataset has been taken down recently and has not been recovered at the moment of writing, so we resorted to locally preprocessing the simple English dataset from Wikipedia dump on another date. We chose the simple English dataset for its smaller size (about 200,000 articles), so as to spend a reasonable amount of time on downloading and preprocessing. Each example contains the content of one full Wikipedia article cleaned to strip markdown and unwanted sections (references, etc.). The data fields are the same among all configurations, containing 'id', 'url', 'title', and 'text' in string format. Due to the size of the dataset, we only used a random set of 50,000 simple English articles for evaluation of our models. The same set is used to compare different methods.

1.1.2 J-Archive Question-Answer Dataset

A JSON file containing 216,930 Jeopardy! questions, answers and other data. The json file is an unordered list of questions where each data point has labels: 'category', 'value', 'question', 'answer', 'round', 'show number', 'air date' (Figure 1). Due to the size of the dataset, we only used a random set of 1000 questions to evaluate our models. The same set is used to compare different methods.

Preprint. Under review.

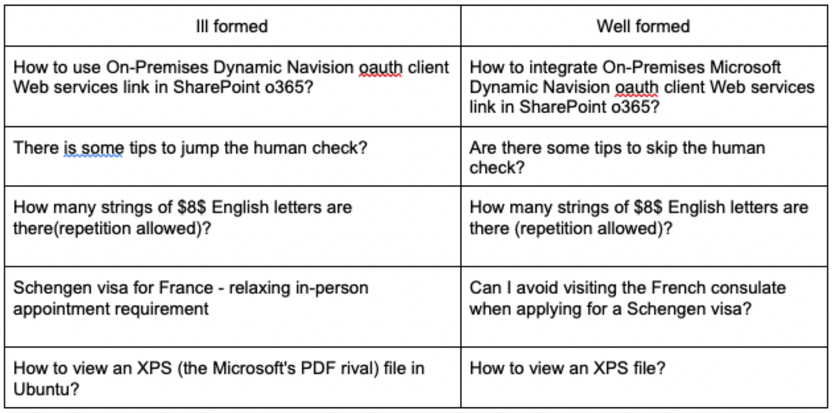


index	Show Number	Air Date	Round	Category	Value	Question	Answer
0	4680	2004-12-31	Jeopardy!	HISTORY	$200	For the last 8 years of his life, Galileo was under house arrest for espousing this man's theory	Copernicus
1	4680	2004-12-31	Jeopardy!	ESPN's TOP 10 ALL-TIME ATHLETES	$200	No. 2: 1912 Olympian; football star at Carlisle Indian School; 6 MLB seasons with the Reds, Giants & Braves	Jim Thorpe
2	4680	2004-12-31	Jeopardy!	EVERYBODY TALKS ABOUT IT...	$200	The city of Yuma in this state has a record average of 4,055 hours of sunshine each year	Arizona
3	4680	2004-12-31	Jeopardy!	THE COMPANY LINE	$200	In 1963, live on "The Art Linkletter Show", this company served its billionth burger	McDonald's
4	4680	2004-12-31	Jeopardy!	EPITAPHS & TRIBUTES	$200	Signer of the Dec. of Indep., framer of the Constitution of Mass., second President of the United States	John Adams

Figure 1: The J-Archive Dataset

1.1.3 MQR Ill-formed Well-formed Question Pairs

The MQR dataset is a multi-domain question rewriting dataset constructed from human contributed Stack Exchange question edit histories [4]. This dataset contains 423,494 entries of both an ’ill formed’ and a ’well formed’ question, as well as a ’category’ tag denoting what kind of question it is.



Ill formed	Well formed
How to use On-Premises Dynamic Navision oauth client Web services link in SharePoint o365?	How to integrate On-Premises Microsoft Dynamic Navision oauth client Web services link in SharePoint o365?
There is some tips to jump the human check?	Are there some tips to skip the human check?
How many strings of 8 English letters are there(repetition allowed)?	How many strings of 8 English letters are there (repetition allowed)?
Schengen visa for France - relaxing in-person appointment requirement	Can I avoid visiting the French consulate when applying for a Schengen visa?
How to view an XPS (the Microsoft's PDF rival) file in Ubuntu?	How to view an XPS file?

Figure 2: MQR Sample Questions

2 Background

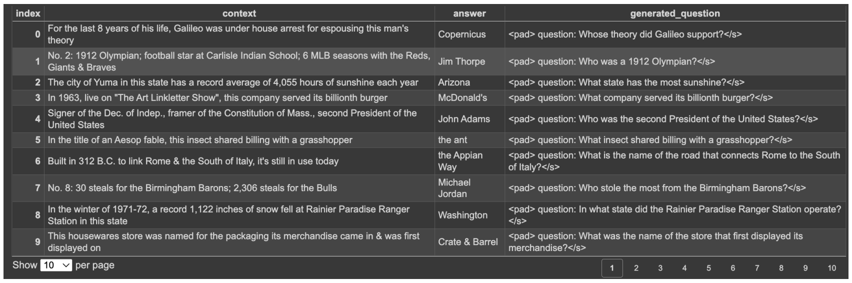
During our initial investigation, we used an extractive QA model from Haystack as a baseline. It uses a passage ranking model, BM25, to select the most relevant Wikipedia articles to provide as context for the QA model itself. Then, it uses the RoBERTa QA model trained with SQuAD 2.0, referred to as the Reader, to extract the answer from the selected Wikipedia article. [2]

2.1 Improper Question Construction

We found that improper question construction impinges on model performance. When asked the Jeopardy! question "For the last 8 years of his life, Galileo was under house arrest for espousing this man’s theory" the correct answer is "Copernicus." The pipeline answers "1642" with confidence score 0.40, "Earth rotating daily and revolving around the sun" with confidence score 0.37, and "Copernicus" with confidence score 0.30. However, when asked "Whose theory did Galileo espouse and was under house arrest as a result for the last 8 years of his life?", the pipeline answers "Copernicus" with confidence score 0.82, "Tycho Brahe" with confidence score 0.72, and "Nicolaus Copernicus" with confidence score 0.69.

To address the question reformulation challenge, we decided to use Google’s pre-trained T5 model as a baseline[5]. The model requires a context and answer. By providing the Jeopardy! clue and corresponding answer, the T5 model generates questions with varying degrees of relevance [Figure 2].

While the model parses relevant information from the clue, the model doesn’t use all details to generate a question. As such, some of the questions generated would not help in searching for an



index	context	answer	generated_question
0	For the last 8 years of his life, Galileo was under house arrest for espousing this man's theory	Copernicus	<pad> question: Whose theory did Galileo support?</s>
1	No. 2: 1912 Olympian; football star at Carlisle Indian School; 6 MLB seasons with the Reds, Giants & Braves	Jim Thorpe	<pad> question: Who was a 1912 Olympian?</s>
2	The city of Yuma in this state has a record average of 4,055 hours of sunshine each year	Arizona	<pad> question: What state has the most sunshine?</s>
3	In 1963, live on "The Art Linkletter Show", this company served its billionth burger	McDonald's	<pad> question: What company served its billionth burger?</s>
4	Signer of the Dec. of Indep., framer of the Constitution of Mass., second President of the United States	John Adams	<pad> question: Who was the second President of the United States?</s>
5	In the title of an Aesop fable, this insect shared billing with a grasshopper	the ant	<pad> question: What insect shared billing with a grasshopper?</s>
6	Built in 312 B.C. to link Rome & the South of Italy, it's still in use today	the Appian Way	<pad> question: What is the name of the road that connects Rome to the South of Italy?</s>
7	No. 8: 30 steals for the Birmingham Barons; 2,306 steals for the Bulls	Michael Jordan	<pad> question: Who stole the most from the Birmingham Barons?</s>
8	In the winter of 1971-72, a record 1,122 inches of snow fell at Rainier Paradise Ranger Station in this state	Washington	<pad> question: In what state did the Rainier Paradise Ranger Station operate?</s>
9	This housewares store was named for the packaging its merchandise came in & was first displayed on	Crate & Barrel	<pad> question: What was the name of the store that first displayed its merchandise?</s>

Figure 3: T5 generated questions from Jeopardy! Clues

answer via a QA model. After observing this behavior, we aimed to implement a modified question generation model that can take into account more relevant information given in a Jeopardy! clue, as well as integrate knowledge of the Jeopardy! category to help guide the model towards generating a more useful question that we can feed into a QA model.

2.2 Large Scale Dataset

We were already faced with many issues brought by the large datasets we are working with in the initial investigation phase, which is why we only picked 50,000 English Wikipedia articles. Even then, Haystack takes around 7 minutes to answer one original (i.e. ill-formed) Jeopardy! question. If it were to answer all 216,930 of them, it would take more than 1000 days! This prompt a more efficient method for relevant Wikipedia article retrieval when given the question. We decided to implement a Facebook AI Similarity Search (FAISS) metric to project the articles into a high-dimensional space and get the nearest neighbors (articles) of the question queried [6]. We also tried to implement DistilBERT as a smaller version of BERT to speed up the process [7].

3 Related Work

The original method for retrieving relevant documents is BM25, a TF-IDF based method [8], which requires scanning through all documents. In contrast, Facebook AI Similarity Search extracts a vector representation, (and in our case, sentence embeddings are already vectors), and relates different objects in a Euclidean space. The Euclidean distance between two objects suggests their similarities [6].

Some recent top performing NLP models relevant to our task are T5 [9] and multiple variations of BERT [10] [7][11]. T5 has a encoder-decoder structure which was designed for Seq2Seq translation and can be easily modified for downstream tasks. T5 formulates any language tasks into the text-to-text format, where a prefix description of the task is attached to each input, instructing the model to perform the task through text without having to vary much of the training pipeline. In the preliminary exploration, we investigated T5, DistilBERT, and RoBERTa. DistilBERT is a more efficient improvement upon BERT [7], which takes a question and the context as its input and outputs an answer [10]. The RoBERTa model is an extension of BERT with modified hyperparameters and pretraining tasks, a larger dataset, and longer training time [11].

4 Methods

4.1 Baseline

The baseline Jeopardy! bot is implemented using DeepSet's Haystack package, with BM25 document retrieval and RoBERTa question answering model (QA model). Because the original preprocessed Wikipedia datasets were taken down, we used 50,000 random simple English Wikipedia articles instead. To get the baseline performance, we randomly selected 1000 random Jeopardy! questions. For each question, we append the category to the question (separated by a semicolon and a space), termed Jeopardy! clue, to provide more information as to what is being asked. Then, given a Jeopardy! clue, the retriever gets the top 5 simple English Wikipedia articles related to the clue, and the QA model generates an answer for each of the top 5 articles as context, and reports the one with the highest confidence. We track the total time it takes to find the top 5 articles and generate the top 1 answer, the average F1 score of the answer, the average exact match (EM) score, and the average of the F1 and EM scores corrected for confidence score.

$$F1\_confidence\_corrected = F1/confidence\_score$$
$$EM\_confidence\_corrected = EM/confidence\_score$$

We have decided to invent such measure to not penalize the model harshly if it understands that the answer generated is not a good one.

4.2 Question Reformulation from Jeopardy! Clues

The baseline T5 model was trained on the SQuAD dataset, which contains context (usually several sentences long), a question, and an answer. Because the model expects more information to be given, it generates suboptimal or irrelevant questions when given only the Jeopardy! clue. In our project, we further finetuned the T5 model upon the MQR dataset. In data preprocessing, we also preprended the prefix "generate question: " and the category to the input sentence. We also excluded questions, such as how and why, that are uncommon in Jeopardy!. Due to computational resources, we fine-tuned the pretrained 'T5-small' model with a batch size of 64 for 900 steps, which took approximately 4 hours per iteration. We used regular cross-entropy loss as a criterion. We evaluated the fine-tuned model using BLEU score on the MRQ dataset. We then performed inference on 1000 random Jeopardy! questions to reformulate them into proper question forms for the QA model. After obtaining the reformulated questions, we ran our baseline model again with these properly formed questions.

4.3 Document Retrieval using FAISS

The FAISS index is added to the document embeddings of each Wikipedia article using huggingface's add_faiss_index() function. The document embeddings are retrieved from huggingface's model "sentence-transformers/multi-qa-mpnet-base-dot-v1", following [12] tutorial. After the FAISS indices are added to the text embeddings, each question from the 1000 Jeopardy! clues (note: not the reformulated questions but the category concatenated ones) are queried, and the top 5 simple English articles out of the 50,000 random ones are selected to be the context for the DistilBERT model. The time for retrieving the top 5 articles is averaged across the 1000 queries and recorded.

4.4 DistilBERT for QA

Due to a teammate's medical emergency, we are unable to report full results and detailed methods we have accomplished. However, we tried to get partial results using [13] implementation of the DistilBERT model. We have the average F1, EM, corrected F1 and EM, and running time of 511 out of the 1000 reformulated Jeopardy! questions.

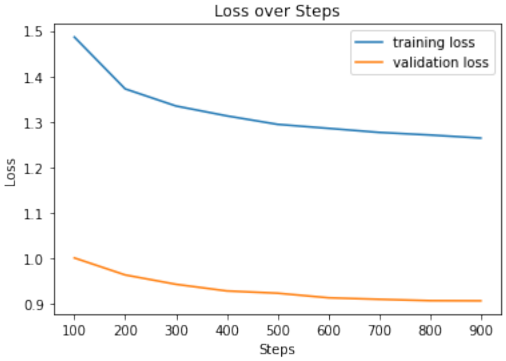
5 Results

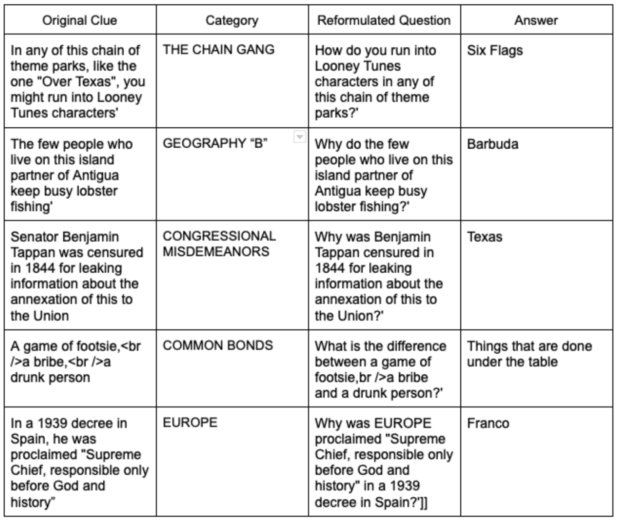
5.1 Question Reformulator

Some generated questions of the fine-tuned T5-small model can be seen in Figure 4. The reformulator achieved a BLEU score of 13.9766, which is lower than the base T5 model's score of 22.63. This result was expected however; further improvements to this model include additional data gathering for training and increased training time.

5.2 Overall Performance

The best performing pipeline is the baseline pipeline with the reformulated questions. After reformulating the questions, the confidence score also seem to increase, as is evident from lower corrected F1 and EM scores despite higher F1 and EM scores when compared to the baseline (Table 1). Despite the original hope that FAISS and DistilBERT may improve the runtime, they do not seem to behave as expected, showing much slower runtime than the baseline (Table 2).





Original Clue	Category	Reformulated Question	Answer
In any of this chain of theme parks, like the one "Over Texas", you might run into Looney Tunes characters'	THE CHAIN GANG	How do you run into Looney Tunes characters in any of this chain of theme parks?'	Six Flags
The few people who live on this island partner of Antigua keep busy lobster fishing'	GEOGRAPHY "B"	Why do the few people who live on this island partner of Antigua keep busy lobster fishing?'	Barbuda
Senator Benjamin Tappan was censured in 1844 for leaking information about the annexation of this to the Union	CONGRESSIONAL MISDEMEANORS	Why was Benjamin Tappan censured in 1844 for leaking information about the annexation of this to the Union?'	Texas
A game of footsie,<br />a bribe,<br />a drunk person	COMMON BONDS	What is the difference between a game of footsie,br />a bribe and a drunk person?'	Things that are done under the table
In a 1939 decree in Spain, he was proclaimed "Supreme Chief, responsible only before God and history"	EUROPE	Why was EUROPE proclaimed "Supreme Chief, responsible only before God and history" in a 1939 decree in Spain?']]	Franco

Figure 4: Reformulator Loss Plot and Generated Questions

	Baseline	Baseline+ReformulatedQs	FAISS+DistilBERT+ReformualtedQs (511/1000)
F1	0.04535	**0.0514**	0.03576
EM	0.03	**0.034**	0.01957
F1 corrected	0.76758	0.4841	0.1134
EM corrected	0.53434	0.2465	0.0414

Table 1: Jeopardy! bot performance across different pipelines.

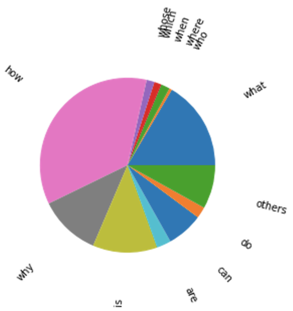
	Baseline Retriever	Baseline QA	FAISS	DistilBERT (511/1000)
time (s)	0.01307	7.1218	0.31085	17.3059

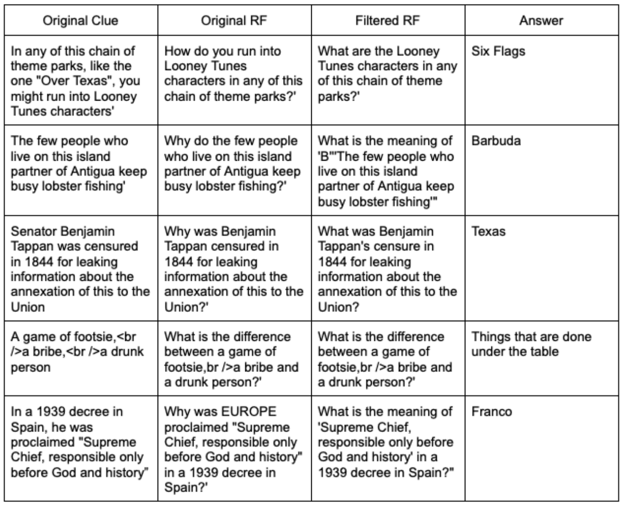
Table 2: Average run time per Jeopardy! question

6 Discussion

6.1 Question Reformulation

We noticed that the reformulator's questions in Figure 4 often started with "how" or "why"; these questions cannot be easily answered with one-word phrases, which are often the answers to Jeopardy! clues. We reasoned it was due to the distribution of questions in the dataset, and we found a majority of questions asked on the Internet to be "how". To further clean the dataset, we kept entries where the "well-formed" question started with "what", "where", "who", "when", "whose", and "which". This significantly reduced the dataset down to 90,000 entries, with the majority of questions starting with "what". The generated questions in Figure 5 also mirror that change. The BLEU score calculated on the MRQ test dataset for this new reformulator is 10.4205, which is lower than the original most likely due to the fact that the generated questions seem to default to prepending the phrase "what is" in front of the original Jeopardy! clue. However, this seemed to be a better strategy as searching some of these generated questions into Google would often lead to the correct answer, even without additional context via Wikipedia links. This result shows that the reformulator learned how to ask a question, but might not know what specific information to ask about. To improve upon this method, we propose pre-training the reformulator on additional paraphrasing datasets or translation datasets before fine-tuning it on MQA, as well as training for a longer period of time.





Original Clue	Original RF	Filtered RF	Answer
In any of this chain of theme parks, like the one "Over Texas", you might run into Looney Tunes characters'	How do you run into Looney Tunes characters in any of this chain of theme parks?'	What are the Looney Tunes characters in any of this chain of theme parks?'	Six Flags
The few people who live on this island partner of Antigua keep busy lobster fishing'	Why do the few people who live on this island partner of Antigua keep busy lobster fishing?'	What is the meaning of 'B"'The few people who live on this island partner of Antigua keep busy lobster fishing'"	Barbuda
Senator Benjamin Tappan was censured in 1844 for leaking information about the annexation of this to the Union	Why was Benjamin Tappan censured in 1844 for leaking information about the annexation of this to the Union?'	What was Benjamin Tappan's censure in 1844 for leaking information about the annexation of this to the Union?	Texas
A game of footsie,<br />a bribe,<br />a drunk person	What is the difference between a game of footsie,br />a bribe and a drunk person?'	What is the difference between a game of footsie,br />a bribe and a drunk person?'	Things that are done under the table
In a 1939 decree in Spain, he was proclaimed "Supreme Chief, responsible only before God and history"	Why was EUROPE proclaimed "Supreme Chief, responsible only before God and history" in a 1939 decree in Spain?'	What is the meaning of 'Supreme Chief, responsible only before God and history' in a 1939 decree in Spain?"	Franco

Figure 5: MQR Question Distribution

6.2 FAISS and DistilBERT, No Improvement on Runtime

The original hypothesis and the rationale for using FAISS and DistilBERT is such that the model could be faster. However, it does not seem to be the case that the original model is slow because of its algorithm inefficiency, but rather due to the shear size of the 50,000 English Wikipedia articles. Once we switched to 50,000 simple English Wikipedia articles, the run time shrunk from 7 minutes to around 7 seconds. FAISS and DistilBERT did not improve runtime at all. One reason could be that because the baseline is implemented as a coherent pipeline in the haystack package, whereas FAISS and DistilBERT are developed separately and not incorporated into a pipeline, the performance could have not been optimized. Alternatively, GPU resources may not have been efficiently utilized for FAISS and DistilBERT.

6.3 Overall Low F1 and EM Scores

We believe that the overall low scores can be attributed to the choice of Wikipedia articles. Simple English Wikipedia articles are much smaller, so it takes less time to load into our program, but that is precisely because there are fewer articles, and each article is significantly shorter. Therefore, critical information could be missing. Another observation we have made is that because the original preprocessed and cleaned Wikipedia article is not longer available, our locally processed articles contain special symbols that perhaps were once tables and some articles are thus littered with a lot of junk, which could have confused the QA model.

6.4 Conclusion

We show that a T5 model can be fine-tuned for question generation from shorter contexts, and having well-formed questions aid QA models answer Jeopardy! clues. We also demonstrate that the baseline (haystack pipeline of using BM25 and RoBERTa) is efficient and can be utilized in conjunction with question reformulation to build a Jeopardy! bot. FAISS and DistilBERT can be used as alternatives but are less efficient.

One important future direction is using the full English Wikipedia dataset to investigate potential improvement on pipeline performance. Other endeavors can focus on adding conversational context to both the question reformulator and the QA model. The question reformulator could potentially produce more informed reformulations that understands, as well as humans, what Jeopardy! questions are asking for. The QA model could potentially understand deeper meanings and connections between topics and better answer questions such as "what is the commonality between a game of footsie, a bribe, and a drunk person? (answer: things that are done under the table)".

References

[1] David Ferrucci and Eric Brown. “Building Watson: An Overview of the DeepQA Project”. In: *Association for the Advancement of Artificial Intelligence* 1 (2010), pp. 59–79.

[2] M. Pietsch et al. *Haystack (Version 0.5.0)*. 2020.

[3] Wikimedia Foundation. *Wikimedia Downloads*. URL: `https://dumps.wikimedia.org`.

[4] Zewei Chu et al. *How to Ask Better Questions? A Large-Scale Multi-Domain Dataset for Rewriting Ill-Formed Questions*. DOI: `10.48550/ARXIV.1911.09247`. URL: `https://arxiv.org/abs/1911.09247`.

[5] Manuel Romero. *T5 (base) fine-tuned on SQUAD for QG via AP*. `https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap`. 2021.

[6] Hervé Jegou, Matthijs Douze, and Jeff Johnson. *Faiss: A library for efficient similarity search*. June 2018. URL: `https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/`.

[7] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv preprint arXiv:1910.01108* (2019).

[8] Lan Chu. *Understanding Term-Based Retrieval Methods in Information Retrieval*. 2022.

[9] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer.” In: *J. Mach. Learn. Res.* 21.140 (2020), pp. 1–67.

[10] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *ArXiv* abs/1907.11692 (2019).

[12] *Semantic search with FAISS*. URL: `https://huggingface.co/course/chapter5/6?fw=tf#using-faiss-for-efficient-similarity-search`.

[13] the Hugging Face team. *DistilBERT base cased distilled SQuAD*. URL: `https://huggingface.co/distilbert-base-cased-distilled-squad`.