Question - Answering System in News Article

ANONYMOUS AUTHOR(S)*

Abstract here

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

ACM Reference Format:

Anonymous Author(s). 2022. Question - Answering System in News Article. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, April 30–May 6, 2022, New Orleans, LA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Understanding the information written in an article is a very important factor [8]. One can understand the importance of the article by the title. In our project, we will work on a news article that has missing information. We will retrieve the answer from an article and either append it to the heading or generate a new heading by summarizing the answer and heading. This not only facilitates normal users but also blind users, as it saves a user's time.

In [20] Wang et al. performed a word-by-word comparison of the passage and the question using a "match-LSTM." Finding a passage's sub-sequence that answers the query is the objective. They used the Pointer Network (Ptr-Net) model to create the replies using the input text's tokens. An easy technique to use Ptr-Net in this situation is to treat the answers as a series of tokens from the input passage while ignoring the fact that these tokens are sequential in the original passage. In particular, the answer is represented as a series of integers a = (a1, a2,...), where each ai is an integer designating a specific location in the passage. The phrase "sequence model" describes this. Alternately, if we want to guarantee consecutively, that is, if we want to be sure that we actually choose a sub-sequence from the passage as an answer, we can use the Ptr-Net to forecast only the beginning and end of a response. The Ptr-Net just needs to choose two tokens from the passage as input in this scenario, and any tokens between those two tokens are treated as the answer. In [21], Wang† et al. have demonstrated the reading comprehension-style question-and-answer system using gated self-matching networks. A bi-directional recurrent network here processes the question and passage independently. They then used gated attention-based recurrent networks to match the question and passage, resulting in a representation of the passage that takes into account the query. Additionally, they have used self-matching attention to refine the passage representation and accumulate evidence from the entire passage, which is input into the output layer (Pointer Network) to forecast the border of the answer span.

From the above articles, we can observe they have used LSTM as an approach for the problem statement. For this project, we'll employ the Transformer-based language model known as BERT (Bidirectional Encoder Representations from Transformers). BERT has been trained to utilize the Transformer Encoder architecture with the Next Sentence Prediction (NSP) pre-training objective using Masked Language Modelling (MLM). The transformer Encoder is used by BERT. Each encoder block in an encoder is made up of two neural network layers, and an encoder has a stack of encoder

Manuscript submitted to ACM

bloc0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 (where the output of one block is provided as the input to the next block). A self-attention layer comes first, followed by a straightforward feed-forward layer. MLM and NSP are used by BERT as their pre-training objectives. In order to achieve these goals, it makes use of a few unique tokens, including CLS, SEP, and MASK. To perform the QA task we add a new question-answering head on top of BERT. The purpose of this question-answering head is to find the start token and end token of an answer for a given paragraph. Everything that comes in between, including the start and end token, is considered an answer. Inside the question-answering head are two sets of weights, one for the start token and another for the end token, which has the same dimensions as the output embedding. The output embedding of all the tokens is fed to this head, and a dot product is calculated between them and the set of weights for the start and end tokens, separately.

Bidirectional Encoder Representations from Transformers is known as BERT [4]. It is among the most well-liked and frequently applied NLP models. By examining the words that follow before and after a word, BERT models can take into account its entire context. This feature is especially helpful for determining the question's intended purpose. Due to its bi-directionality, which gives it a greater understanding of linguistic context and flow, it is now used in many NLP tasks.

In Figure 1 we demonstrate our approach to the problem statement. The user search for a news article runs our extension, then our model extracts the heading of the article and converts it into a question, in the next step model process the entire article and looks for the answer to the question, and enhances user with the entire information about the article in synthesis format.

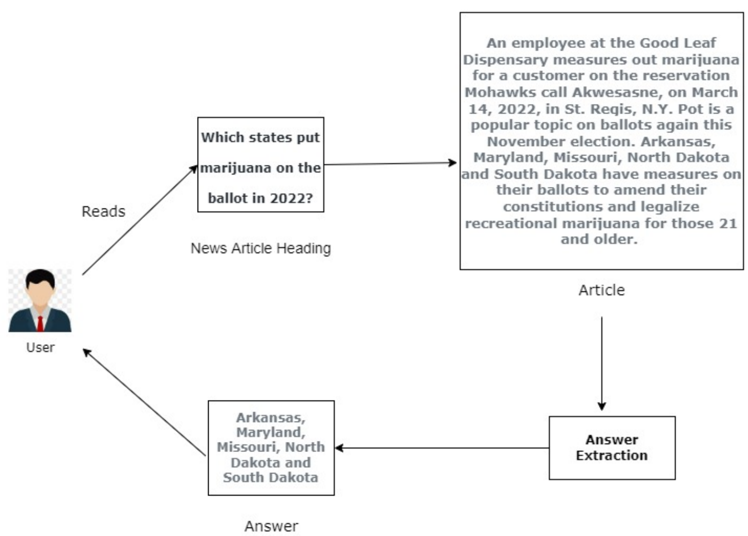

Fig. 1. Overall setup of Answering System in News Article

2 RELATED WORK

2.1 Web Accessibility

Throughout these years, many have implemented screen readers for visually impaired people for web accessibility [1, 2, 7, 13]. News articles have much information that is not relevant to read. Sighted users can shift through the information faster, however, blind people have to listen through the content narrated by the screen readers. To address this problem in this paper [1], they have proposed an underlying algorithm for automatic summarization that could be used for non-visual skimming. In this paper [13], they have analyzed blind user interaction with an information service (news) in order to understand the main accessibility and usability issues related to it when using a screen reader. Sometimes the way a web page is displayed might be inadequate for visually impaired people as they will not get the gist of information and thus it becomes necessary to rearrange web content. To resolve this issue, in this article [7] they have proposed an approach by combining simplification and summarization techniques to transform an entire website into a simpler and more easily navigable website that could be explored by persons with visual impairments. Moreover, there should be proper guidelines to design a website for better accessibility and this [2] does the same and presents requirements to improve the accessibility of websites for blind Web users.

Even though some of them did this on news articles or news websites, they did not address the problem of missing information in the news article heading that we have described in this paper. So by addressing this problem, visually impaired people do not have to go through an article, they will have the missing information appended to the article heading itself which in turn saves their time.

2.2 Question Answering Systems

Lots of research have been done on question answering system[10, 12, 14–16, 19, 23]. Among which few have done it on news articles [14, 19, 23]. News articles have been archived for many years and contain many events that happened in past years. A research [19] has been to have proposed a question-answering system that answers event-focused questions in these temporal collections of news articles by introducing a document retriever module to retrieve articles that have timestamp mentioned in the question and a time-aware re-ranking module to rank those articles. This article [23], explores the use of question-answering (QA) techniques to support personalized news video retrieval. And this paper [14] presents a new data set that provides the temporal context of both, the questions and the knowledge required to answer them, answers are from 14 years of time-stamped news articles.

There have been several rule-based models proposed for question-answering tasks. In [17] Lin et al. explained about JAVELIN Cross-Language Question Answering system. that generated answers for questions and information, they translate the model from to multi languages question answering systems such as Chinese and Japanese. In [6] Gusmita1 et al. explain about Question and answer system by combining two different architectures, one of the architectures uses appropriate documents, and another uses a rule-based method. this forms a new architecture, that was designed to collect relevant documents related to the keywords and then use a rule-based method to collect answer candidates. In [9] Jain et al. propose the architecture of a question-answering system for the medical domain and discuss rule-based question processing and answer retrieval. The paper also discusses rule formation for retrieving answers, the approach for the problem statement is first they processed the Question to comprehend the query posed by the user It is further divided into two categories: (a) Question Analysis and Classification and (b) Query Reformulation.The query produced by the question processing module is fed into a document retrieval engine. Document processing is based on the extraction of documents while keeping the question in mind. A set of relevant documents is chosen from which candidate answer

passages containing the relevant text are extracted. These candidate responses will be fed into the answer processing module. The answer matching and ranking module and user answer voting make up the answer processing module. Answer matching compares potential replies obtained from document processing modules to the intended answer type produced by the question processing module. The list of responses produced by the answer matching and rating component will be displayed to the users during answer voting. The best system-generated response will receive the most user votes.

Furthermore, a plethora of research has been done and many powerful deep learning models [3, 5, 11, 18, 20, 21] have also been introduced to solve this problem. For instance, in [20] they have used a "match-LSTM" to perform word-by-word matching of the passage with the question. In this [21], they have used a gated attention-based recurrent network and propose a self-matching attention mechanism to refine the representation by matching the passage against itself. This article [11] introduces pointer networks with one attention step to predict the blanking-out entities. This paper [18] proposes an iterative alternating attention mechanism to better model the links between question and passage. In this [5], they proposed iterative selecting of important parts of the passage by a multiplying gating function with the question representation. This [3] proposes a two-way attention mechanism to encode the passage and question mutually.

Moreover, since the introduction of pretrained transformer-based language model BERT, there have been many QA systems developed using it. For instance, [24] demonstrated integrating BERT with the open-source Anserini IR toolkit to create BERTserini, an end-to-end open-domain question-answering (QA) system. [22] have proposed a multi-passage BERT model to globally normalize answer scores across all passages of the same question. This use of pretrained model outperforms previous attempts at question answering.

Similarly, we are going to use this pre-trained BERT model. However, unlike the research mentioned above for which they have used datasets available online, we will introduce our own custom dataset gathered manually of news articles that have the missing information in their heading.

Our system consists of two main components. First is classification and second is QA(Question Answering) model. Firstly, we gathered the news article headings and labeled them manually as complete(0) and incomplete(1). By mentioning incomplete, that means it has the missing information that is hidden in that particular article which we are going to extract it using our second component, the QA model. As you can see the process is straightforward. First, we have a news article heading that we are going to pass through the classifier. The classifier will be going to predict whether it's complete or not. If it's complete then we will not do anything but it is incomplete, then we are going to pass that article heading to our QA model. QA model will go through an entire article and extract the answer for us.

2.3 New heading classifier model

As mention in previous section, the first step to achieve the end goal we need to build classifier for this we have created dataset whose size is 1500. This dataset was trained over differeent classifiers such as Multinomial Naïve Bayes Classifiers, Bernoulli Naive Bayes, Logistic Regression, Random Forest, support vector machine (SVM), KNeighborsClassifier and two Convolutional Neural Networks classification algorithm.

2.4 QA(Question Answering) model

After classifying the article heading comes the task of extracting the answer. For this task, we have used the BERT model. Fig 2 shows a BERT architecture for a close domain task.

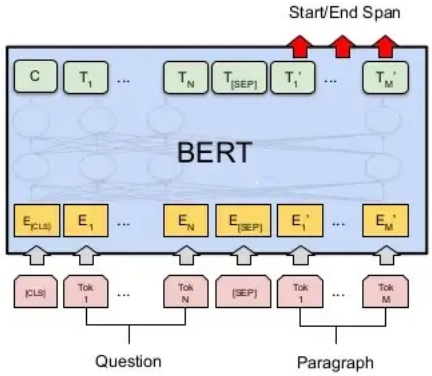

Fig. 2. BERT Architecture

3 SYSTEM

Fig 3 summarizes our proposed system.

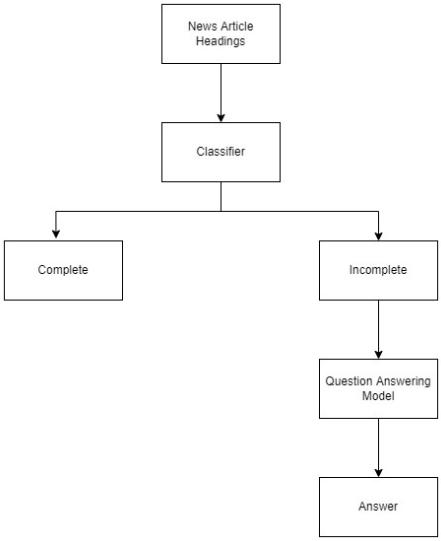

Fig. 3. Proposed System

The data is pre-processed then its trained with algorithm that are mentioned above. The steps taken to pre-processed the dataset is as follows.

1. The first step while cleaning the heading is by removing unwanted spaces, lowercasing the data, removing punctuation's, removing numbers, removing contractions.

2. Tokenizing the dataset, Tokenization is the process of dividing text into a set of meaningful pieces. These pieces are called tokens. For example, we can divide a chunk of text into words, or we can divide it into sentences. Depending on the task at hand, we can define our own conditions to divide the input text into meaningful tokens.

3. Encoding the dataset, Text encoding is a process to convert meaningful text into number / vector representation so as to preserve the context and relationship between words and sentences, such that a machine can understand the pattern associated in any text and can make out the context of sentences.

4. The dataset is splitted inot text and train dataset to have a appropiate number of dataset to train and validate the model.

For the Question Answering task, BERT takes the input question and passage as a single-packed sequence. The input embeddings are the sum of the token embeddings and the segment embeddings. The input is processed in the following way before entering the model:

1. Token embeddings: A [CLS] token is added to the input word tokens at the beginning of the question and a [SEP] token is inserted at the end of both the question and the paragraph.

2. Segment embeddings: A marker indicating Sentence A or Sentence B is added to each token. This allows the model to distinguish between sentences. In the below example, all tokens marked as A belong to the question, and those marked as B belong to the paragraph.

To fine-tune BERT for a Question-Answering system, it introduces a start vector and an end vector. The probability of each word being the start word is calculated by taking a dot product between the final embedding of the word and the start vector, followed by a softmax over all the words. The word with the highest probability value is considered.

Surprisingly, The Hugging Face Transformers library has a BertForQuestionAnswering model that is already fine-tuned on the SQuAD dataset. And we trained it on our custom dataset.

We have manually gathered the dataset for this which is completely different from the dataset mentioned in the above section. This dataset is in Squad2.0 format. So it has incomplete article headings, which in turn serves as a question. Also, it has an entire article that serves as context. We have annotated this dataset so that it has an answer field, which is a dictionary containing answer text, an answer start index, and an answer end index, and hence becomes the close domain task. We fine-tuned our model on this and test it out on a custom test dataset.

4 EVALUATION

4.1 Classification model

We trained custom dataset with respect to 8 different classification algorithms. The six classifiers obtained less than 60%, (fig 4) however the deep learning model obtained more than that. The first model obtained 66% accuracy, we used the same model for our deployment.

In Fig 5 we can see the accuracy and loss graph between training and validation dataset from first CNN classification model. We can observe the training accuracy reached 100% accuracy after 10th epoch but its not the same in validation accuracy, this shows the model was unable to make proper prediction as it was trained> the validation accuracy reached 66% accuracy.

In Fig 6 we have demonstrated the ROC curve with respect to all the models. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.

4.2 QA(Question Answering) model

The results of the QA model are surreal. We are getting an accuracy of 10 percent while testing the model on our test dataset. The Fig 7 table shows the evaluation report.

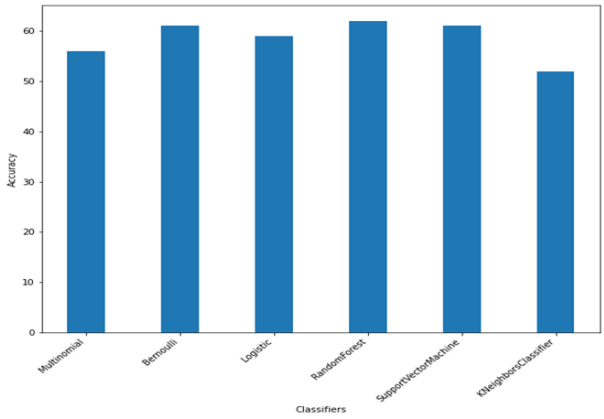

Fig. 4. Accuracy graph between six classifiers

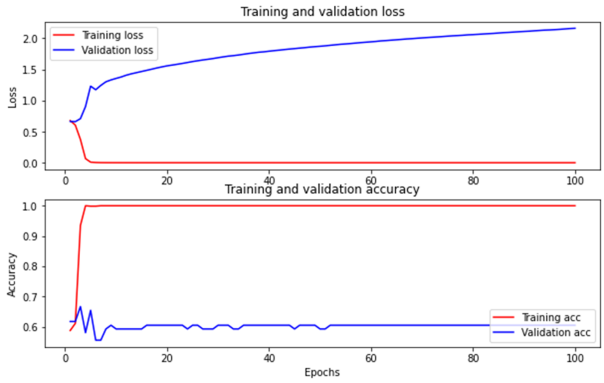

Fig. 5. Loss and accuracy graph between training and validation dataset

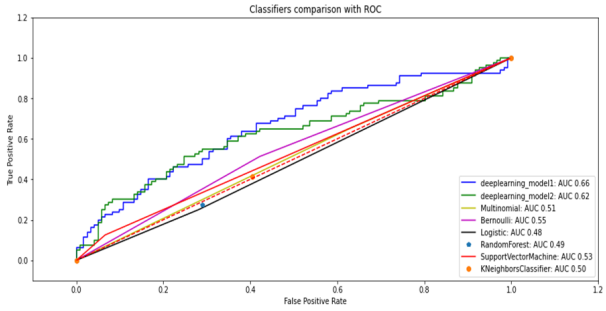

Fig. 6. Roc, accuracy with respect to all the models

As clearly seen in the table, the model is not predicting the exact start and end indexes of the most of the answer for a particular question and thus getting an accuracy of 10. it seems like for only for example 1 and 9, it's predicting the exact start index, however, for other examples, the start index prediction is not even near to the original start indexes.

		Start Index	End Index
Example 1	True	235	236
	Prediction	235	159
Example 2	True	40	45
	Prediction	289	76
Example 3	True	121	137
	Prediction	83	93
Example 4	True	275	289
	Prediction	44	46
Example 5	True	101	123
	Prediction	118	6
Example 6	True	188	191
	Prediction	147	152
Example 7	True	239	246
	Prediction	21	70
Example 8	True	87	96
	Prediction	404	131
Example 9	True	49	53
	Prediction	49	20
Example 10	True	11	18
	Prediction	27	18

Fig. 7. BERT Evaluation Reportl

Similarly, for the end index, only for example 10 it's predicting correctly, and for other examples the situation is the same as for start indexes.

By closely analyzing this table there is one peculiarity is seen for many examples, that is the start index is greater than the end index.

5 CONCLUSION

By doing this project, we introduced a unique way of avoiding many irrelevant information for news article heading which has missing information by extracting the answer from the news article, , which saves a user's time by not going through that entire article. We also created custom datasets for two of our main components, that are classification and QA(Question Answering), in an attempt to train different machine learning models on these datasets. We have also seen different results received in classification and QA. Also, we are interested in creating a user interface that is easily interacted by blind users. In addition, we want to improve the QA model and also approach it using pointer network and summarization of article, in future.

Code link for Question and Answer model

code link for Classification Algorithm

REFERENCES

[1] Faisal Ahmed, Yevgen Borodin, Andrii Soviak, Muhammad Islam, I.V. Ramakrishnan, and Terri Hedgpeth. 2012. Accessible Skimming: Faster Screen Reading of Web Pages. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 367–378. https://doi.org/10.1145/2380116.2380164

[2] Rehema Baguma and Jude T Lubega. 2008. Web design requirements for improved web accessibility for the blind. In *International Conference on Hybrid Learning and Education*. Springer, 392–403.

[3] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2016. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423* (2016).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549* (2016).

[6] Ria Hari Gusmita, Yusuf Durachman, Salman Harun, Asep Fajar Firmansyah, Husni Teja Sukmana, and Adam Suhaimi. 2014. A rule-based question answering system on relevant documents of Indonesian Quran Translation. In *2014 International Conference on Cyber and IT Service Management (CITSM)*. IEEE, 104–107.

[7] Stephanie Hackett and Bambang Parmanto. 2006. Usability of access for web site accessibility. *Journal of Visual Impairment & Blindness* 100, 3 (2006), 173–181.

[8] John Hartley. 2013. *Understanding news*. Routledge.

[9] Sonal Jain and Tripti Dodiya. 2014. Rule based architecture for medical question answering system. In *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012*. Springer, 1225–1233.

[10] Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 792–802.

[11] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547* (2016).

[12] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300* (2019).

[13] Barbara Leporini. 2011. Google news: how user-friendly is it for the blind?. In *Proceedings of the 29th ACM international conference on Design of communication*. 241–248.

[14] Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*. PMLR, 13604–13622.

[15] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2383–2392. https://doi.org/10.18653/v1/D16-1264

[16] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515* (2021).

[17] FrankLin HidekiShima MengqiuWang TerukoMitamura. [n.d.]. CMU JAVELIN System for NTCIR5 CLQA1. ([n. d.]).

[18] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830* (2016).

[19] Jiexin Wang, Adam Jatowt, Michael Färber, and Masatoshi Yoshikawa. 2021. Improving question answering for event-focused questions in temporal collections of news articles. *Information Retrieval Journal* 24, 1 (2021), 29–54.

[20] Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905* (2016).

[21] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 189–198.

[22] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167* (2019).

[23] Hui Yang, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua. 2003. VideoQA: question answering on news video. In *Proceedings of the eleventh ACM international conference on Multimedia*. 632–641.

[24] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718* (2019).