iTA: A Digital Teaching Assistant

Vishnu Dutt Duggirala
Department of Computer Science and Engineering, University of Colorado Denver
vishnudutt.duggirala@ucdenver.edu

Keywords: Machine Reading Comprehension, Question-Answering

Abstract: We have designed and implemented a question-answering chatbot, dubbed iTA (intelligent Teaching Assistant), which can answer detailed questions by effectively identifying the most relevant answers in "long" text sources (documents or textbooks). iTA answers questions by implementing a two-stage procedure. First, the topmost relevant paragraphs are identified in the selected text source using a retrieval-based approach and scores for the retrieved paragraphs are computed. Second, using a generative model we extract the relevant content from the top-ranked paragraph to generate the answer. Our results show that iTA is well suited to generate meaningful answers for questions posed by students.

1 INTRODUCTION

Online learning offers flexibility and availability, allowing students continue their studies, even during situations such as pandemics. It possesses a few challenges such as guidance counselling, taking feedbacks for course evaluation, the need of a tutor. The problems with online learning have heightened the interest in making digital tools that can help a student to face these challenges. To make it feel more like interacting with an actual human, chatbots have emerged into existence.

The chatbots can support students in several ways and they are more focused on to answer practical questions like services provided by the administrative office or financial office. They can serve as a tutor which explains a topic in a chapter and quizzes the student. Hardly any researchers concentrated on a chatbot which can give a detailed answer on a textbook question presented by a student, this encouraged us to create a tool where it takes the role of teaching assistant and helps the student to understand the academic concepts by making a machine perused the entirety of the reading material and answer the question. Machine reading comprehension or the ability to read and understand the unstructured text and then answer questions about it remains a challenging natural language processing task motivated by a wide variety of applications. Throughout this paper, the abbreviation MRC will be used to refer to Machine Reading Comprehension. For example, as shown in Figure 1, a search engine with MRC techniques can precisely return the right response to questions presented by users in natural language instead of a progression of related web pages.

To develop iTA we need to overcome two challenges: The first challenge is that most MRC solutions focus on comprehension from a passage of size no more than 500 words. In contrast, a typical textbook may contain more than 15,000 words, this is mainly due to limitation of the deployed MRC models, such as Match LSTM (Wang and Jiang, 2016), a basic comprehension neural network. To achieve a query-aware context representation without early summarization BiDAF (Seo et al., 2018) is used, and BERT (Devlin et al., 2019) that uses transformers and self-attention mechanism to reduce the computation time and extend the attention of the model for MRC tasks beyond vanilla LSTM. Still, it fails given documents longer than 512 tokens. The Second challenge is the current question answering datasets will provide extractive or specific short answers. As the questions will be complicated, short responses cannot address these questions, MS MACRO v2 (Bajaj et al., 2018) response is considered better with an average of 13.6 words. Still, the average length of the documents is 56 words, whereas TriviaQA (Joshi et al., 2017) has multi-sentence support, but their answers are shorter than most of the datasets.

We present a chatbot that can serve as a digital TA, dubbed iTA, and can answer questions based on long text resources (e.g., textbooks) selected by the user. Toward this end, we have adopted and adapted a

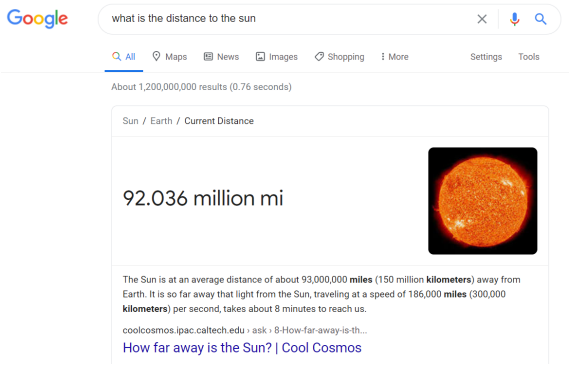

Figure 1: An example of a Google search engine with an MRC technique

model (Clark and Gardner, 2017) that is developed for multiple passages MRC tasks. It is a two-part process given a question, our implementation first narrows down to top five paragraphs using TF-IDF to compute the score of each passage, and then processes the top-ranked passage(s) using BART (Lewis et al., 2019) that trained on ELI5 (Fan et al., 2019) to generate the answer to the given question. Here is an example how iTA answers to a given question.

Question: How is range different from an array?
Answer: A range is an array of numbers in increasing or decreasing order, each separated by a regular interval. An array is a collection of numbers separated by an interval. A range can be used to represent a very large number of numbers, so it's worthwhile to use it. Ranges are useful in a surprisingly large amount of cases, so they are worthwhile to be used in a very small number of cases. The difference is in how they are used. The range is used to describe the order of the numbers in the array, while the array is used as a representation of the order in which the numbers are in the range.

2 RELATED WORK

This section shows a considerable amount of published literature of MRC methods, Conversational AI, and a similar system to iTA. Conversational AI response can be classified as Retrieval and Generation.
Retrieval-based generation is where the agent's response is based on pattern matching. The query is matched to pool of messages of message-response candidates and the response whose message has high matching score with query is selected. Generative-based response generation does not select response from any predefined candidates but instead creates a new one. Due to these reasons, generative-based has advantage of attending to unseen situations and can be creative in their responses while retrieval-based tend to be more grammatically right as they are taken from human's responses. Finally, a dialogue agent can employ both of these methods to generate response, leading to hybrid-based models.

2.1 MRC Methods

A considerable literature has grown up around the theme of MRC on single-paragraph by researchers. Existing works include developing various architectures Match-LSTM (Wang and Jiang, 2016), R-Net (Wang et al., 2017), and designing multiple attention mechanisms such as BiDAF (Seo et al., 2018), Transformers (Vaswani et al., 2017) to achieve more precise and improved answers to questions. BERT (Devlin et al., 2019) advanced state-of-the-art results on 11 NLP tasks, which includes a single-paragraph MRC task.
A few researchers have focused on multi-passages question-answering. Longformer (Beltagy et al., 2020) to address the token limitation of BERT (Devlin et al., 2019), they have used an attention mechanism which changes linearly with the sequence length, has achieved up to 8 times more.This network (Zhang et al., 2018) took the advantages of hierarchical-attention (Wang et al., 2018) to learn the paragraph level representation and implement the match-LSTM (Wang and Jiang, 2016) mechanism.

2.2 Educational Chatbots

Most of the presented educational chatbots focus on assisting students by taking a "Yes." or "No." response from the user. (Fabio Catania and Garzotto, 2020) they have developed a system which allows a 9-year-old kid to create a bitmoji by talking to a chatbot. Every time they responded to a question, they show the visual representation of the feature described directly on the avatar in the GUI. At every step, it would ask if the user is satisfied with the avatar. Suppose the answer is "No" or any negative expression, it will remove the feature and goes to the previous intent. A chatbot (Donya Rooein, 2020) which assists the student progress in the course and provides metadata and description of the video, Figure 2 shows the interface of the chatbot. iTA is a question-answering system which helps a student in an ongoing-semester by assisting in coping with the subject (Skjuve, 2020).
Tutor presented by the (Hobert, 2019) uses is a prede-

fined learning path where a student is first explained about a topic and moves on to asking a question, the student is also asked if they need any additional content for better understanding of the concept.

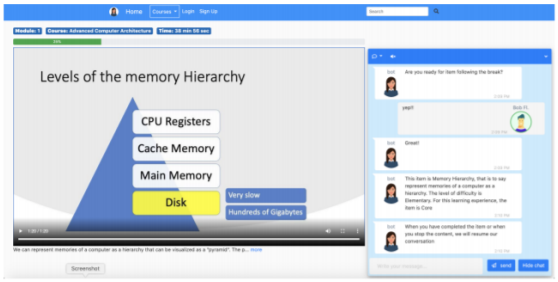

Figure 2: Adaptive Conversations for Adaptive Learning; figure courtesy of (Donya Rooein, 2020)

FIT-EBot (Hien et al., 2018) is an administrative support chatbot which will give information about the course registration, course score, prerequisite course, exam schedule many more. All the above-mentioned chatbots are not serving our purpose which is to explain the conceptual answer for any complex question.

3 METHODOLOGY

The methodological approach taken in this study is a mixed methodology based on a paragraph selection module (Clark and Gardner, 2017) to extract the highest scored passage and a sequence-to-sequence model for answer generation module that uses BART (Lewis et al., 2019) architecture—in section 3.1, we will discuss the overview of the system. In section 3.2, we will go through an in-depth review of each model in our system.

3.1 System Overview

Design Requirements to achieve iTA, a system which supports multi-paragraph, avoid noisy labels, use unstructured source for question answering, respond with long generative answers. The textbook is an unstructured data which is a '.txtfile'. Figure 3 depicts the overall architecture and data flow in iTA. iTA is a two-tier model which has a passage selection module and answer generative module. The textbook which is used has been formatted into an unstructured data, we have cleaned the data by removing the mathematical equations, tables, and syntax or python code that is present in "The fundamentals of Data Science" (). When a student asks a question, the query along with the textbook, is passed through TF-IDF in the first module to screen the top 5 featured paragraphs which possibly contains the answer, this step is employed to narrow down the given textbook which cuts the computation time with a huge margin as calculating the confidence score of each paragraph in the multi-paragraph model has been now cut to five paragraphs. Using the top two highest confidence paragraphs the second module generates a long answer.

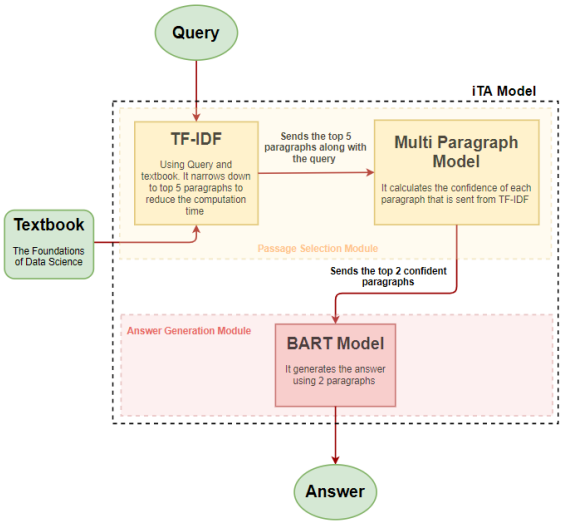

Figure 3: Overview of iTA

3.2 Model

3.2.1 Passage Selection Module

The paragraph selection module selects the paragraph that has the smallest TF-IDF cosine distance with a question. To calculate term frequencies, will use the paragraphs within the relevant documents, not the entire corpus (Clark and Gardner, 2017) this approach will give more weight to question words that are less common, extracts the relevant paragraphs, to reduce the noisy labels, the attention mechanism is used that optimizes the sum of the probabilities of all answer spans in the paragraphs and computes the confidence of each paragraph.

The weight in TF-IDF is a statistical measure used to evaluate how important a word in a collection of document or corpus. The weights are a multiplication of TF, IDF. The definition for TF-IDF in our approach (Clark and Gardner, 2017) is number of times a term occurred in a paragraph and divided by the total number of paragraphs.

GloVe(Pennington et al., 2014) to embed the ques-

Figure 4: Multi-Paragraph Reading Comprehension architecture; figure courtesy of (Clark and Gardner, 2017)

tion and context, The character-level and word-level embeddings are then concatenated and passed to the next layer, a shared bi-directional GRU(Cho et al., 2014) to map context and question. Attention and self-attention mechanisms are employed to build a representation between query-context, and passage and itself, respectively. The last layer is prediction softmax operation is applied to predict the probability of the answer.

3.2.2 Generative Module

We have used a model (Wolf et al., 2020) in which weights are pre-trained on BART (Figure 5) and trained on ELI5 (Fan et al., 2019) to generate an answer from the high confidence paragraph. BART (Lewis et al., 2019) is a sequence-to-sequence model with a bidirectional encoder (like BERT (Devlin et al., 2019)) over corrupted text and a left-to-right autoregressive decoder (like GPT (Radford, 2018)).
BART encoder is like an encoder block of BERT (Devlin et al., 2019). In each encoder, it has only multi-attention and feed-forward layers. It is more reliable when it comes to seq2seq modelling, which allows

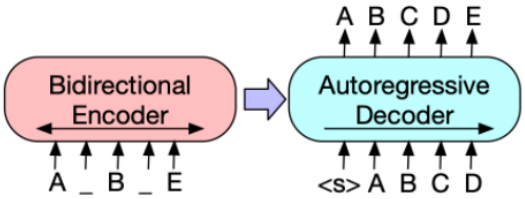

Figure 5: BART architecture; figure courtesy of (Lewis et al., 2019).

seeing the entire input sequence all at once, directly modelling these dependencies using attention. BERT decoder randomly replace the tokens with masks, and missing tokens predicted independently, so cannot be used for generation.
BART decoder uses transformer decoder block; its decoder has an extra layer which is masked self-attention which does not allow a position to peak at tokens to its right. The key difference with the transformer decoder is that it outputs one token at a time just like traditional language models (auto-regression), can be used for generation. It uses a beam search to create the response. Larger beam widths result in better performance of a model as the multiple candidate sequences increase the likelihood of better matching a target sequence. This improved performance results in a decrease in decoding speed.

4 EXPERIMENTATION

4.1 Datasets

Extractive datasets such as SQuAD (Rajpurkar et al., 2018), NewsQA (Trischler et al., 2017), SearchQA (Dunn et al., 2017) restrict the response to a word or short expression from the input, TiviaQA (Joshi et al., 2017) offers which challenges the models to perform reasoning across multiple paragraphs with an average of 2895 words in a document but the answer is still short. ELI5 (Fan et al., 2019) has overcome this long answer challenge. We observed from the Figure 6 that TriviaQA had offered multiple paragraphs, but the response is short. So, will be using TriviaQA for the multi-paragraph module and use ELI5 for generating module.

We have used TriviaQA (Joshi et al., 2017) to train the paragraph selection module with the shared-normalization confidence method (Clark and Gardner, 2017), stated that the shared-normalization confidence method fetched more transparent results than

Dataset	Average # of Words			1st Question Word Frequency (%)								# Q-A Pairs
	Question	Document(s)	Answer	Why	How	What	When	Where	Who	Which	OTHER	
ELI5	**42.2**	857.6 (212K)	**130.6**	**44.8**	**27.1**	18.3	**11.3**	2.0	1.8	0.8	6.1	**272K**
MS MARCO v2 (Nguyen et al., 2016)	6.4	56	13.8	1.7	16.8	35.0	2.7	3.5	3.3	1.8	35.3	183K
TriviaQA (Joshi et al., 2017)	14	**2895**	2.0	0.2	3.9	32.6	2.0	2.1	16.8	**41.8**	0.6	110K
NarrativeQA (Kociský et al., 2018)	9.8	656	4.7	9.8	10.7	38.0	1.7	**7.5**	**23.4**	2.2	6.8	47K
CoQA (Reddy et al., 2018)	5.5	271	2.7	2	5	27	2	5	15	1	**43**	127K
SQuAD (2.0) (Rajpurkar et al., 2018)	9.9	116.6	3.2	1.4	8.9	**45.3**	6.0	3.6	9.6	4.4	17.6	150K
HotpotQA (Yang et al., 2018)	17.8	917	2.2	0.1	2.6	37.2	2.8	2.2	13.8	28.5	12.8	113K

Figure 6: Comparison of QA datasets and how ELI5 is better; figure courtesy of (Fan et al., 2019).

the merge method. Sequence-to-sequence, answer generation module, uses a model that is trained on ELI5 (Wolf et al., 2020). To test our application, we used data science textbook as a long document, preprocessing is done on the data to remove mathematical equations.

4.2 Setup

We trained the model with Adadelta optimizer with a batch size of 60 and used TriviaQA-unfiltered data because the dataset does not specify which document contains the answer. So attempts to answer a question using a document retrieval system and top of that shared-norm approach gave improved results. Once we had this trained model, we sent our test dataset (textbook) and a question to fetch the top 5 paragraphs using the TF-IDF approach. Forwarded to the shared-norm model to get the confidence of each paragraph, more noisy labels will have less confidence value. The high confidence value paragraph will be sent to the BART generative model to get the answer.

4.3 Results

We have employed BLEU score (Papineni et al., 2002) for quantitative analysis; it is a metric for automatically evaluating machine-translated text. It measures the similarity of the machine-generated text to a set of human-created reference text. A value of 0 means that the output has no overlap with the reference text (low quality). In comparison, a value of 1 means there is perfect overlap with the reference (high quality). The score between 30 to 40 is understandable to good translations. Perplexity is one more quantitative measure which calculates how well a probability distribution predicts a sample. We used this measure to show the comprehension of each response given by the iTA.

Perplexity is one more quantitative measure which calculates how well a probability distribution predicts a sample. We used this measure to show the comprehension of each response given by the iTA. Perplexity is like a branching factor the lower the better.

We gathered the top 2 paragraphs each with 400 words from the passage selection module and fed it to the generated answer model which uses beam search, and we set beam length to 5, the minimum size of the answer to 96. Calculated the BLEU score for each response given by the machine, this is not ideal but just to show the level of understandable in the generated text.

Explanation of how our two-step process works:

If a question is asked, "What is Data Science?", it is sent to the first module in iTA where the TF-IDF uses entire corpus (textbook) along with a question to narrow down to 5 paragraphs that may contain the answer. The 5 paragraphs are sent to Multi-Paragraph model to calculate the confidence of each paragraph. Choosing the highest confident paragraph, the BART model will generate the answer as shown in Figure 7.

We have used the following textbook for this exper-

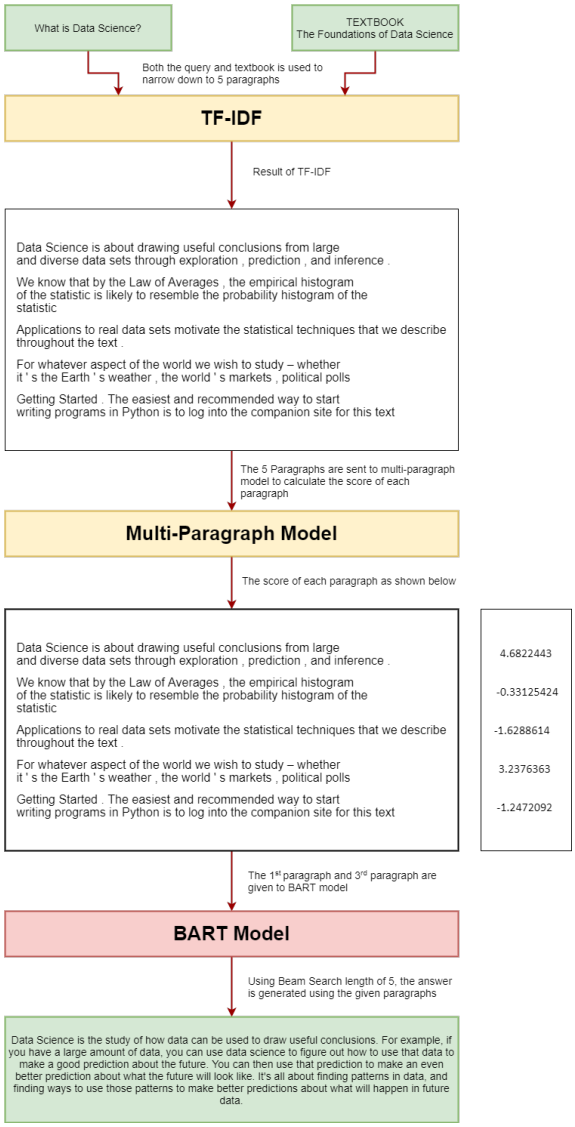

Figure 7: Data flow of iTA.

iment "The Foundations of Data Science." (Adhikari and John DeNero, 2019) and by asking the four stu-

dents with different majors to state any questions that they came across while reading the textbook. We have chosen 75 significant questions and graphs are plotted for perplexity, BLEU score, and time-taken for each response.

The average response time of iTA is 15 seconds. The

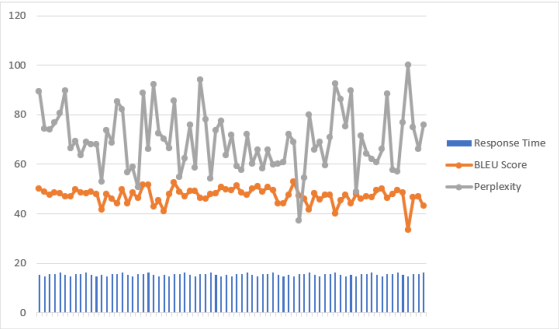

Figure 8: The Distribution of BLEU score, perplexity, and time taken to respond by the model for all 75 questions.

distribution of BLEU score ranging between 40 - 50 as shown in Figure 8 which means it has achieved a high quality translation. The values of perplexity shows that the text is understandable. Using human evaluation we conclude that out of 75 questions, iTA has failed to answer 16 questions. 38 were right and 21 responses were acceptable, in other words 80 percent of the time iTA gives the right answer.

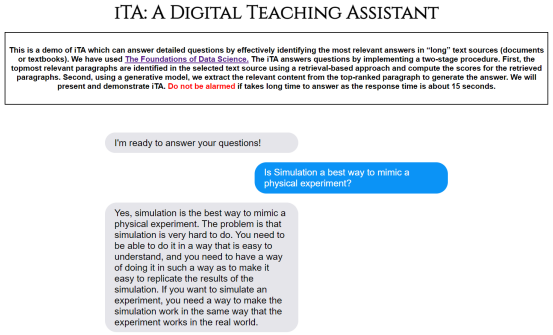

Figure 9: iTA interface

4.4 Discussion

Situations where iTA fails:

1. **Any keyword in the question is not present in the textbook or long document by default; it takes the first paragraph of the document and generates the answer.**
2. **To answer the question, the context in the document should explicitly state the keyword.**
   For example given explanation for Data type in the textbook is "*Every value has a type, and the built-in type function returns the type of the result of any expression.*" we pose the following question:
   Example Question: What are Data types?
   Generated Answer: A Data Type is a type of number. For example, an integer is a number, a float is a float, a decimal is a decimal, and a hexadecimal is a hex. Data Types can also be used to describe types of numbers. For instance, if you have a number of digits, you can use a Data Type to describe what those digits are, and how they relate to each other. So, for example, if I have a set of numbers, I can use Data Types to describe how those numbers relate to one another.
   As there is no "*Data Type*" keyword the actual explanation, The retrieval model has extracted a passage which contains more "*Data Type*" tokens.
3. **It is hard to calculate the weights for complex questions.**
   For example here "*Data Types*" and "*Python language*" are never used in a single paragraph; it took the best section possible and tried to answer the question.
   Example Question: What are Data Types in Python language?
   Generated Answer: Data types are a way of representing data in a way that the programmer can understand. For example, if you have a list of numbers, you can use a data type to represent the number in the list. For instance, if I have a set of numbers and I want to know how many of those numbers are in that set, I could use a DataType to represent that number in a list. You can also use Data Types to represent data in other ways, for example if I wanted to know the age of a person, I can use the DataType for that

4.5 Conclusion and Future Work

Reading long documents is a challenging task for machines; existing work only focus more on short passages or single-paragraph. Most available datasets concentrate on single- or double-word reply, we have implemented an application which can give a detailed answer to a question by identifying the most relevant paragraph from a long document and generates the answer. We achieved this by combining a multi-paragraph model which reduces the noisy labels while selecting relevant paragraphs and a generative model which replies a long sentence.

We will add more documents and allow users to choose which document (textbook) to ask a question. At present the response time of a chatbot is about one minute and should be working on decreasing the time.

REFERENCES

Adhikari, A. and John DeNero, B. (2019). The foundations of data science.

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., and Wang, T. (2018). Ms marco: A human generated machine reading comprehension dataset.

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

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Clark, C. and Gardner, M. (2017). Simple and effective multi-paragraph reading comprehension.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Donya Rooein, P. P. (2020). Adaptive conversations for adaptive learning: Sustainable development of educational chatbots.

Dunn, M., Sagun, L., Higgins, M., Guney, V. U., Cirik, V., and Cho, K. (2017). Searchqa: A new qa dataset augmented with context from a search engine.

Fabio Catania, Micol Spitale, G. C. and Garzotto, F. (2020). Conversational agents to promote children's verbal communication skills.

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

Hien, H. T., Cuong, P.-N., Nam, L. N. H., Nhung, H. L. T. K., and Thang, L. D. (2018). Intelligent assistants in higher-education environments: The fit-ebot, a chatbot for administrative and learning support. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, SoICT 2018, page 69–76, New York, NY, USA. Association for Computing Machinery.

Hobert, S. (2019). Say hello to 'coding tutor'! design and evaluation of a chatbot-based learning system supporting students to learn to program.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *In EMNLP*.

Radford, A. (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.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2018). Bidirectional attention flow for machine comprehension.

Skjuve, M. (2020). "from start to finish": Chatbots supporting students through their student journey.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., and Suleman, K. (2017). Newsqa: A machine comprehension dataset.

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

Wang, S. and Jiang, J. (2016). Learning natural language inference with lstm.

Wang, W., Yan, M., and Wu, C. (2018). Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714, Melbourne, Australia. Association for Computational Linguistics.

Wang, W., Yang, N., Wei, F., Chang, B., and Zhou, M. (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)*, pages 189–198, Vancouver, Canada. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Huggingface's transformers: State-of-the-art natural language processing.

Zhang, Y., Zhang, Y., Bian, K., and Li, X. (2018). Towards reading comprehension for long documents. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 4588–4594. AAAI Press.