Climate Bot: A Machine Reading Comprehension System for Climate Change Question Answering

Md Rashad Al Hasan Rony[1,2] , **Ying Zuo**[1] , **Liubov Kovriguina**[1] ,
Roman Teucher[1] , **Jens Lehmann**[1,2]
[1]NetMedia, Fraunhofer Institute for Intelligent Analysis and Information Systems, Germany
[2]Smart Data Analytics Group, University of Bonn, Germany
{rashad.rony, ying.zuo, liubov.kovriguina, roman.teucher, jens.lehmann}@iais.fraunhofer.de,
jens.lehmann@cs.uni-bonn.de

Abstract

Climate change has a severe impact on the overall ecosystem of the whole world, including humankind. This demo paper presents **Climate Bot** - a machine reading comprehension system for question answering over documents about climate change. The proposed Climate Bot provides an interface for users to ask questions in natural language and get answers from reliable data sources. The purpose of the climate bot is to spread awareness about climate change and help individuals and communities to learn about the impact and challenges of climate change. Additionally, we open-sourced an annotated climate change dataset **CCMRC** to promote further research on the topic. This paper describes the dataset collection, annotation, system design, and evaluation.

1 Introduction

The impacts of climate change are global in scope and may threaten species and people communities' survival. Among the most serious threats is the growing temperature of the Earth's atmosphere, causing sea levels to rise, ecosystems to collapse, and catastrophic weather events to become more common. Leveraging machine learning (ML) and Artificial Intelligence (AI) has already helped mitigate climate change effects. This has been done using various ML tasks involving predictive modeling, i.e., natural hazard prediction, reducing factory emissions, modeling temperature changes, and ice melting. Conversational AI applications in this domain are not yet numerous. Applying machine reading comprehension (MRC) over climate change documents can expand the benefits of question answering interfaces to this area, such as faster answer spotting in comparison to traditional search, natural human-machine interaction, and insights extraction from massive document collections. Moreover, the educational impact of such applications is hard to overestimate.

To help people know more about climate change from trusted data sources, we have created a dataset for training MRC and designed and implemented an MRC pipeline providing question answering services over climate change problems and challenges. The motivation behind the project is to speed up access to information about climate change challenges and promote awareness about it by allowing natural questions over trusted sources. We open-source the data and code used in this demonstrator [1]. A video demonstrator of the climate bot can be found on Youtube [2]. The contributions of the demo paper can be summarized as follows:

- Climate Bot, a novel MRC system for question answering over climate change documents with publicly available code.
- A climate change dataset CCMRC, as a manually annotated publicly available resource for training QA and MRC applications, having 21,081 question-answer pairs and 7,400 paragraphs, extracted from trusted data sources.

2 Climate Bot System

The primary goal of the climate bot is to perform Machine Reading Comprehension in the climate change domain. Given a user question $\mathcal{Q}$, the climate bot first fetches documents ($\mathcal{D}_n$) relevant to $\mathcal{Q}$. Then the system displays the documents and answers highlighted in them using a web interface. Here, n is the number of documents. Climate bot consists of three main components: 1) a *Retriever* 2) a *Reader*, and 3) a *User Interface (UI)*.

To set up the system, we first pre-process all the documents $\mathcal{D}$. Each document $d \in \mathcal{D}$ is passed through Sentence-BERT [Reimers and Gurevych, 2019] to get a contextualized vector representation ($\vec{d}$) of the document which is indexed into a dense space using an hierarchical indexing algorithm from Dense Passage Retriever (DPR) [Karpukhin *et al.*, 2020]. Indexing the contextualized documents into a dense space closely clusters documents with similar types and content, allowing the climate bot to find relevant documents quickly and efficiently. Only the data pre-processing steps need to be executed to extend the climate change bot with more data. The data pre-processing steps are depicted in Figure 1.

Retriever. The task of the *Retriever* component is to fetch documents relevant to the user question. In our proposed

[1] https://github.com/rashad101/Climate-Bot-IJCAI22
[2] https://youtu.be/DdRh6P4sgQw

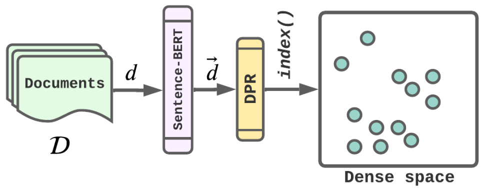

Figure 1: The data pre-processing pipeline, showing how documents are stored into a dense space.

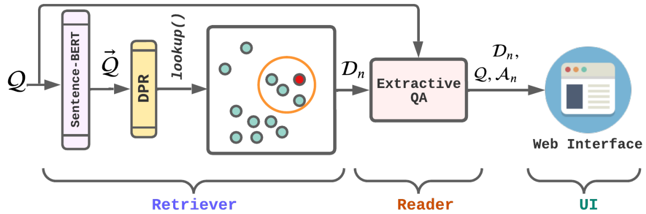

Figure 2: System architecture.

system, a Dense Passage Retriever (DPR) [Karpukhin *et al.*, 2020] is employed to retrieve user-question relevant documents from the dense space. To do so, the contextualized vector of the question is obtained from Sentence-BERT. The DPR then utilizes the contextualized representation to perform lookup following a K-nearest neighbor (KNN) approximate search algorithm and fetch n number of documents relevant to the user question. The value of n can be configured from the user interface of the system demonstrator.

Reader. The task of the *Reader* component is to extract a text span from the documents that answers the user question. We leverage the language model ALBERT [Lan *et al.*, 2020] to extract an answer span given a question and a document as input. The ALBERT model was previously pre-trained on the SQuAD [Rajpurkar *et al.*, 2018] dataset, a widely used cross-domain MRC dataset. We fine-tuned the model with the climate change data. The Reader component extracts one answer per document, which is then displayed in a user interface.

User Interface (UI). We developed a web interface that allows a user to type questions and receive the most relevant documents along with highlights of the answer to the question inside the documents (see screenshot in Figure 3).

The system architecture is illustrated in Figure 2. Furthermore, the system demonstrator we developed is shown in Figure 3. The main functionalities of the demonstrator are described below:

- Example questions, to give the user a starting point to try out our system. Example questions are clickable cards, located at the bottom part of the demonstrator. Once clicked, the question and its answer will pop up in the chat section.
- An input field, where users can type in their question and press *Enter* in their keyboard or click on the *Send* button to get the answer in the same way as with the example questions.

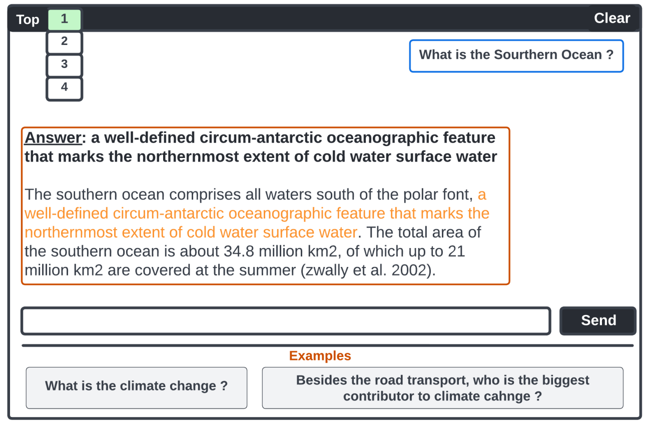

Figure 3: System demonstrator.

- The main body of the demonstrator is designated for showing the question and answer. The answer is shown in a card where the first line shows the answer in **bold**, followed by a document fetched by the *Retriever* wherein the answer is highlighted in orange.
- A drop-down button on the top-left corner to configure the value of n (Top n), indicating how many documents the *Retriever* should fetch.
- A reset button, located at the top-right corner of the demonstrator to clear excessive chat contents.

3 CCMRC: Climate Change Dataset

3.1 Data Sources and Acquisition

We collected data from various trusted data sources. The data from each source was pre-processed and split into documents/paragraphs. Given these documents, we asked the Amazon Mechanical Turk (AMT) and in-house annotators from the Smart Data Analytics group to manually write question-answer pairs. The trusted data sources used to construct the dataset are listed below.

- **Official Semantic Scholar Dump** [3], where the research articles with the words *Climate* or *Climate change* in their title, are selected.
- The official reports of the **Intergovernmental Panel on Climate Change Special Reports on Climate Change** for the years 2019-2021 [4].
- **NASA Global Climate Change** [5].
- **European Commission Climate Change Data** [6].
- Individual documents and news articles from **CNN, The Guardian, National Geographic, New York Times, World Health Organization**.

We have implemented PDF-parser and used third-party libraries to collect the research articles and reports listed above.

[3]https://api.semanticscholar.org/corpus/download/
[4]https://www.ipcc.ch/reports/
[5]https://climate.nasa.gov/ask-nasa-climate/
[6]https://ec.europa.eu/clima/climate-change_en

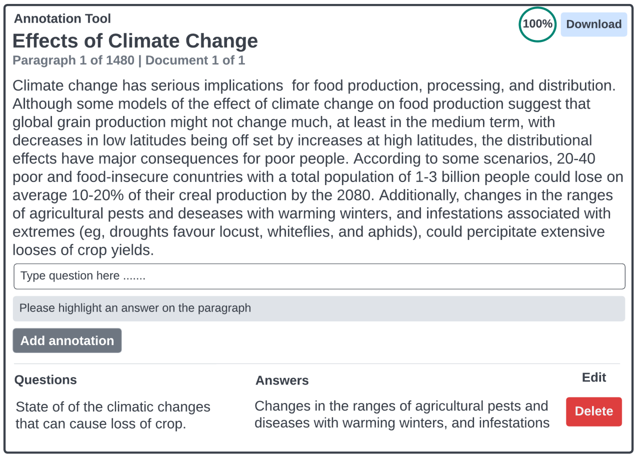

Figure 4: The in-house annotation tool used to collect question answer pairs for training the *Reader* module.

3.2 Data Annotation

Each article was split into paragraphs/documents for manual annotation. To keep the quality of annotated question-answer pairs consistent, documents with a word count of less than 150 words were excluded from manual annotation. As the train set, 7,527 paragraphs were manually annotated. We asked Amazon's Mechanical Turk (AMT) annotators to write three question-answer-pairs for each document obtained from *Semantic Scholar* articles. The annotated question-answer-pairs were then transformed to the widely used SQuAD [Rajpurkar *et al.*, 2018] dataset format for training and testing of the reader model.

To obtain a realistic estimation of the test performance of the trained question-answer (QA) system, we created an additional test set, from here on referred to as the in-house annotation test set, from the listed data sources except for *Semantic Scholar*. The in-house annotation test set contains documents from 30 articles. Figure 4 shows the annotation tool used for generating the in-house test set. The annotation tool allows the annotators to download the annotated data in SQuAD format.

Five QA-pairs from each of the 30 articles annotated by the in-house annotators were randomly sampled to assess the quality of annotated QA-pairs. Each QA-pair was evaluated by three different cross-validators who were not previously involved in the annotation process. We asked each cross-validator to rate the question and answer individually. Based on the grammar, the relevance and the contextual meaning of the QA-pair, each question and answer were scored from 1 to 4, where 1 corresponds to '*poor*' and 4 corresponds to '*excellent*'. The evaluation guidelines are provided in our Github repository [7].

3.3 Dataset Statistics

The dataset statistic is reported in Table 1. There are 7,400 paragraphs and 21,081 QA-pairs used for training and testing the reader model of the Climate bot system. The AMT annotated data were split at the paragraph level into train, validation, and test sets with the ratio of 70/20/10. We created a set of 960 QA-pairs from 495 documents as the in-house annotation test set.

	Train	Validation	Test
Number of paragraphs	5,180	1,480	740
Number of QA-pairs	14,756	4,229	2,096
Avg. word count (question)	9.59	9.49	9.57
Avg. word count (document)	212.80	210.92	208.01
Avg. word count (answer)	26.06	25.80	25.81

Table 1: Dataset statistics.

4 Evaluation

We used automatic metrics (F1 score, BLEU [Papineni *et al.*, 2002] and METEOR [Banerjee and Lavie, 2005]) to evaluate the performance of the *Reader* component. It is noteworthy that the *Retriever* component in Climate Bot works in an unsupervised manner. The evaluation result is reported in Table 2.

	F1 score	BLEU	METEOR
Validation	0.826	0.682	0.815
Test	0.816	0.678	0.808
In-house annotation test	0.661	0.416	0.694

Table 2: Performance of the *Reader* component.

The evaluation results demonstrate that the climate bot can answer climate-related user questions with high accuracy. The proposed climate bot is developed in a modular way, allowing the system to be extendable with minimal effort. We evaluated the quality of the in-house test set by calculating an average over the cross-validators ratings for the questions and answers separately. Given the scale, we received an average score of 3.69 for questions' quality and 3.79 for answers' quality.

5 Conclusion

We presented **Climate Bot**, an MRC system for question answering about climate change. We open-sourced an annotated dataset **CCMRC** and the code to encourage further research to make an impact on the climate change domain. As future work, we intend to (1) expand the dataset and improve topics distribution, (2) improve the MRC model by adding support for follow-up questions, and (3) provide supports for multilingual MRC.

Acknowledgments

We acknowledge the support of the following projects: SPEAKER (BMWi FKZ 01MK20011A), JOSEPH (Fraunhofer Zukunftsstiftung), OpenGPT-X (BMWK FKZ 68GX21007A), the excellence clusters ML2R (BmBF FKZ 01 15 18038 A/B/C), ScaDS.AI (IS18026A-F) and TAILOR (EU GA 952215).

[7] https://github.com/rashad101/Climate-Bot-IJCAI22

References

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. 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, 2005.

[Karpukhin *et al.*, 2020] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[Lan *et al.*, 2020] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[Rajpurkar *et al.*, 2018] Pranav Rajpurkar, Robin Jia, and Percy Liang. 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, July 2018. Association for Computational Linguistics.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.