Papr Readr Bot: A Conversational Agent to Read Research Papers

Michelle Xiao-Lin Foo
michellefooxiaolin@huawei.com
Device Software Lab, Munich
Research Center, Huawei
Technologies
Munich, Germany

Luca Della Libera
luca.della.libera@huawei.com
Device Software Lab, Munich
Research Center, Huawei
Technologies
Munich, Germany

Ilhan Aslan
ilhan.aslan@huawei.com
Device Software Lab, Munich
Research Center, Huawei
Technologies
Munich, Germany

ABSTRACT

Reading research papers can be a tedious and lonely task. We present Papr Readr Bot, a (chat)bot which aims to assist researchers in reading papers more effectively and with less cognitive effort by providing paper summaries, answering questions about the paper, extracting figures from the paper, taking notes, and generate citation. Papr Readr Bot demonstrates and provides hands-on experiences of various deep learning-based skills that can be integrated in useful and social conversational assistants for reading related contexts. We open-sourced Papr Readr Bot at https://github.com/michellefxl/papreadr

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; **Natural language interfaces**.

KEYWORDS

natural language understanding, summarization, question answering, keyword extraction, conversational agents

1 INTRODUCTION

Reading can be tedious. Thus, it is not surprising that fellow researchers have provided guidelines for how to read a research paper [8] and commercial tools, such as *Scholarcy*[1] also exist to help with reading efficiency. At times reading can be a lonely activity, such as when one has to go through a stack of papers on their own, but other times reading is considered as joyful and relished. Many research labs implement journal clubs and similar activities

[1] https://www.scholarcy.com

to get people to interact with each other, to share their impressions and know-how of papers they have read. Social interactions in the context of reading papers can be motivating and arguably will address potential issues related with prolonged loneliness and foster well-being.

While there are limitations, considering the types of conversations agents can currently have with users [6] with bots mainly being perceived as tools, there are also benefits of bots using social talk. In related research, for example, enjoyable bots that can tell jokes or use irony in small talk have been presented (e.g. [10, 12]). It is undeniable that bots still have many weaknesses when compared to the richness of interpersonal communication. But a bot is always there to talk with, never tired, with infinite patience, and often non-judgemental. By utilizing natural language processing bots have the potential to interact with users from all walks of life, languages, and in various contexts, such as in the office on a desktop computer or on a mobile when one is commuting to work.

Therefore, we believe that in future bots will assist in reading tasks and serve as a smart and likable companion in reading activities. However, there are many challenges since the activity of reading and the process of comprehension may happen distributedly throughout a day or days, and require a lot of flexibility from any assistive bot. Furthermore, reading is essentially a creative process and as such the attention and focus of the reader may switch between specific questions to a more open explorative conversation about one or more papers.

In the past, rules and manual dialog modelling were essential to build conversational assistants, such as mobile assistants for tourists, which aimed to help with everyday multilingual conversational activities, such as communicating with a taxi driver or ordering food in a restaurant [1]. Rule-based systems work well in cases when the conversation flow can be easily predefined. More flexibility may be achieved in the ongoing deep learning era, where conversational agents have access to increasingly powerful models. These models allow agents to perform a variety of useful tasks that can be integrated in conversations and require less explicit modelling. Paradigmatic examples are machine translation models, summarization models, and question answering models. However there are additional challenges with deep learning models where, for example, they often have to be fine-tuned to work for specific tasks.

Consequently, we need prototypes to explore conversation designs, to test combinations of different models, and to gain hands-on conversation experiences. Ultimately, our aim is to foster progress

towards bots that can comprehensively assist in reading activities and addressing all the aspects mentioned.
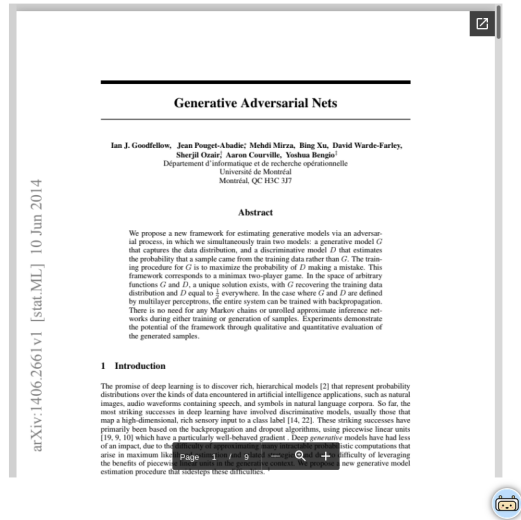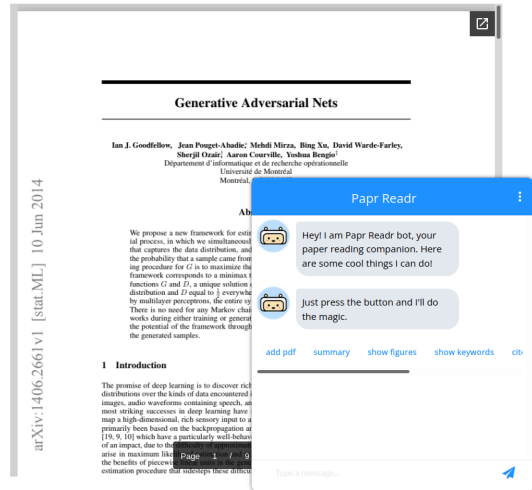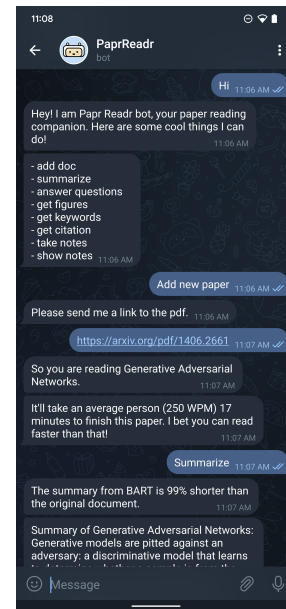

(a) Chat widget minimized.

(b) Chat widget expanded.

Figure 1: Web application interface. The window can be adjusted so that the chat is visible on the side of the PDF.

2 PAPR READR BOT

Papr Readr Bot is a bot that guides people in reading, understanding and retrieving information from one or multiple PDF documents.

Figure 2: Telegram (mobile) chatbot interface.

The chatbot can be used on different devices, mobile phones or desktop devices (e.g. laptop). For the desktop version of the demonstrator, we developed a web application which accesses the chatbot through a web-widget (see Figure 1). When the browser window is not in full screen, the chat widget can be minimized in order to have a full view of the PDF. On mobiles the chatbot can be accessed using the well-known mobile application *Telegram*[2] (see Figure 2). Users can interact with it by writing text messages and pressing the "send" button. As a first step, Papr Readr Bot asks the users to provide the link to the research paper users intend to read. The web application on the desktop includes a PDF viewer, so the PDF will be rendered on the page when the chatbot receives the link to the PDF. Users can then engage in a free-form conversation with the virtual assistant, which offers a range of functionalities to enhance the reading efficiency and experience, including text summarization, question answering, note taking, extraction of figures, keywords and references and also generate citation in *BibTeX*[3] format for the paper. In case the user is unsure about the skills of the chatbot, they can ask the chatbot for help, and the chatbot will display a list of skills that the user can choose from as seen in Figure 1. In addition to assisting the reader in extracting information from papers, Papr Readr Bot also tells the reader about some interesting facts, such as the time needed to read the current paper, calculated based on the average reading speed of an adult which is 250 words per minute and the percentage of content reduced through summarization. Besides that, Papr Readr Bot has a cheerful personality, manifested through a friendly and informal conversational style, which makes reading papers a less dull experience.

---

[2]https://telegram.org
[3]http://www.bibtex.org/Format/

2.1 System Architecture and Implementation

The front-end web application and chat widget is written in HTML and JavaScript. Papr Readr Bot is an intent-based chatbot deployed on a local server. When the end user interacts with the chatbot through the chat widget, the user's input is sent to Rasa [2]'s back-end. Rasa is an open-source Python-based framework to develop custom AI-powered chatbots. It includes two main components, namely Rasa NLU and Rasa Core. Rasa NLU acts as an interpreter that parses user's inputs, recognizes intents, extract entities and structured information. Rasa Core is responsible for state tracking, dialog management, and response generation. In particular, based on Rasa NLU's preprocessed input, it predicts the action to perform from a predefined set of options. An action can be as simple as sending a standard message to the user (e.g. "How can I help you?"), or as complex as scheduling an activity in a smart home scenario (e.g. "Turn on the lights in five minutes"). Custom Rasa actions are written for the various skills that Papr Readr Bot can carry out. Following is an overview of the custom actions and details of their implementations:

- Extract summary: Summarization is done using the *BART large CNN*[4] model from *Meta AI*[5], which is available on the popular *Hugging Face*[6] machine learning platform. It is pre-trained on English language and fine-tuned on *CNN Daily Mail*[7]. The summary of the paper is saved and if the user later asks for the summary of the same paper, Papr Readr Bot will read it from memory
- Question answering: The *Tiny ROBERTA*[8] model from Deepset pre-trained on *SQUAD 2.0*[9] dataset which is also available on Hugging Face is used. The chatbot will retrieve the top 3 answers from the question answering model and present them to the user
- Extract keywords: Using the Python package *YAKE*[10] which is an implementation by Campos et al. [3–5]. The top 10 keywords are retrieved from the paper
- Extract figures: All the figures are extracted from the PDF and saved. Papr Readr Bot will display the images in the chat widget and the user can click on the images to open them in separate browser tabs
- Generate citation: *arxivcheck*[11], which is a module of *bibcure*[12] is used to extract the BibTeX citation of the current paper given its arXiv link. Alternatively the citation can be retrieved using the paper's title through *title2bib*[13], another module of bibcure
- Add PDF: Given a PDF URL, a function is written to extract the details of the PDF from its available metadata. But the PDF's metadata is often incomplete, and so the citation method is used as an alternative way to retrieve the paper's details. The action also checks if the paper already exists in

the history of read papers and estimates the reading time for the paper. Papr Readr Bot will tell you when it recognizes a paper that has already been read
- Make notes: The user can add notes to the paper through this function. Each paper will have its own JSON file that collects all the notes
- Show notes: The chatbot will retrieve all the saved notes of the current paper and show them to the user
- Help: On the web application chat widget, Papr Readr Bot will display a list of selectable skills to the user, whereas in Telegram, only a list is shown

In order to extract the text from the PDF, the Python package *PyMuPDF*[14] is used. PyMuPDF has various functions that enable us to interact with the PDF on a deeper level, including extracting figures and meta information from the paper. Before running actions such as summarization, question answering, etc., the text is first preprocessed to remove unnecessary information such as the acknowledgements and references sections. The user can still query for those sections separately. This is done in order to reduce noise that would be introduced if those sections are included.

2.2 Demo at the Conference

Our demonstration aims to showcase Papr Readr Bot to the CUI community. After the initial setup and a brief introduction on the potential benefits of using a bot as an assistant for reading papers, we will encourage participants to engage in a free-form conversation with our chatbot to autonomously discover the functionalities that it offers. In-person attendees will be able to interact with both the mobile and the desktop version of the demo, hosted on a local server. Virtual attendees will be able to use the mobile version. To this end they will need to install Telegram on their smartphones, the chatbot's username on Telegram will be made available to them. In case of technical problems (e.g. a query is not understood or the response is unsatisfactory), we plan to collect their feedback (with their permission and consent) and discuss opportunities for improvement.

3 CONCLUSIONS AND FUTURE WORK

In this work we introduced Papr Readr Bot, a novel tool that combines state-of-the-art deep learning-based natural language processing techniques to facilitate researchers in reading papers more effectively and with less cognitive effort. Although only a first prototype, it already shows great potential, and thanks to the rapid advancements in conversational AI technology we are witnessing nowadays, in near future it could be developed into a fully-fledged virtual mentor or companion, capable of guiding and interacting flexibly with users, accommodating their needs.

As immediate next steps towards this goal, we aim to extend Papr Readr Bot in a few ways. Firstly, the intent recognition engine should be made more robust against noisy or unexpected input messages. Indeed, dialog breakdowns negatively affect usability as users tend to lose their trust in the system. This can be accomplished statically, by retraining the underlying model on a greater amount of manually-labeled data, or dynamically, by employing active learning techniques [9] to periodically recalibrate the intent

---

[4] https://huggingface.co/facebook/bart-large-cnn
[5] https://ai.facebook.com/
[6] https://huggingface.co
[7] https://huggingface.co/datasets/cnn_dailymail
[8] https://huggingface.co/deepset/tinyroberta-squad2
[9] https://huggingface.co/datasets/squad_v2
[10] https://pypi.org/project/yake/
[11] https://github.com/bibcure/arxivcheck
[12] https://github.com/bibcure/bibcure
[13] https://github.com/bibcure/title2bib

[14] https://pypi.org/project/PyMuPDF/

classifier based on interaction logs and user feedback. Secondly, additional features such as information retrieval from web pages to answer queries that require external knowledge, the ability to provide related work reading recommendations, speech input for better accessibility, and integration with other platforms and services (e.g. *Facebook Messenger*[15]) should be implemented. Lastly, Papr Readr Bot can be used as a playground to explore with users less "functional" but more "stylistic" aspects of human-computer interaction, including user experience based on different style transfers [7] and linguistic alignments [11], in order for the virtual assistant to adapt to the user's conversational style and sound more natural, engaging and trustworthy.

4 ACCESSIBILITY AND INCLUSIVITY STATEMENT

As described in Section 2.2, we will offer participants who are interested in testing our demo two user interfaces: a desktop version and a mobile version (both iOS and Android smartphones are supported). Virtual attendees will be able to experiment with the mobile interface, in-person attendees with both.

ACKNOWLEDGMENTS

REFERENCES

[1] Ilhan Aslan, Feiyu Xu, Hans Uszkoreit, Antonio Krüger, and Jörg Steffen. 2005. COMPASS2008: Multimodal, Multilingual and Crosslingual Interaction for Mobile Tourist Guide Applications. In *Intelligent Technologies for Interactive Entertainment*, Mark Maybury, Oliviero Stock, and Wolfgang Wahlster (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 3–12.

[2] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181* (2017).

[3] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences* 509 (2020), 257–289. https://doi.org/10.1016/j.ins.2019.09.013

[4] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. A Text Feature Based Automatic Keyword Extraction Method for Single Documents. In *Advances in Information Retrieval*, Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury (Eds.). Springer International Publishing, Cham, 684–691.

[5] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. YAKE! Collection-Independent Automatic Keyword Extractor. In *Advances in Information Retrieval*, Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury (Eds.). Springer International Publishing, Cham, 806–810.

[6] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, et al. 2019. What makes a good conversation? Challenges in designing truly conversational agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[7] Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics* 48, 1 (04 2022), 155–205. https://doi.org/10.1162/coli_a_00426 arXiv:https://direct.mit.edu/coli/article-pdf/48/1/155/2006608/coli_a_00426.pdf

[8] Srinivasan Keshav. 2007. How to read a paper. *ACM SIGCOMM Computer Communication Review* 37, 3 (2007), 83–84.

[9] Bing Liu and Sahisnu Mazumder. 2021. Lifelong and Continual Learning Dialogue Systems: Learning during Conversation. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 17 (May 2021), 15058–15063. https://ojs.aaai.org/index.php/AAAI/article/view/17768

[10] Hannes Ritschel, Ilhan Aslan, David Sedlbauer, and Elisabeth André. 2019. Irony Man: Augmenting a Social Robot with the Ability to Use Irony in Multimodal Communication with Humans. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 86–94.

[11] Laura Spillner and Nina Wenig. 2021. *Talk to Me on My Level – Linguistic Alignment for Chatbots*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3447526.3472050

[12] Klaus Weber, Hannes Ritschel, Ilhan Aslan, Florian Lingenfelser, and Elisabeth André. 2018. How to shape the humor of a robot-social behavior adaptation based on reinforcement learning. In *Proceedings of the 20th ACM international conference on multimodal interaction*. 154–162.

---

[15]https://www.messenger.com