Proceedings of the 7th International Conference on Inventive Computation Technologies (ICICT 2024)
Xplore Part Number : CFP24F70-ART ; ISBN : 979-8-3503-5929-9

Constructing the ChatGPT for PDF Files with Langchain – AI

Dr.T. Prem Jacob[1]
Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology
Chennai, India
premjac@yahoo.com

Dr. Beatriz Lucia Salvador Bizotto[2]
Department of Social and Applied Sciences
University Center Unifacvest
Lages, Brazil, South America
prof.beatriz.bizotto@unifacvest.edu.br

Dr Mithileysh Sathiyanarayanan[3]
Research & Innovation
MIT Square, London
mithileysh@mitsquare.com

2024 International Conference on Inventive Computation Technologies (ICICT) | 979-8-3503-5929-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICICT60155.2024.10544643

***Abstract*— Queries in PDFs can be time-consuming and labor-intensive because of the unstructured nature of the PDF document type and the need for accurate and relevant search results. By applying cutting-edge algorithms for natural language processing to examine PDF documents and extract relevant data, LangChain solves these difficulties. It makes use of an easy search interface, adjustable filters, and efficient indexing and retrieval mechanisms to enhance the search experience. To efficiently retrieve relevant information from PDF documents, users can annotate critical portions, store queries, and create bookmarks with LangChain. The characteristics of LangChain improve overall productivity and greatly simplify PDF querying. Semantic search, driven by the latest Transformer language models, represents a significant evolution in information retrieval systems. This research work explores the capabilities of semantic search to efficiently retrieve documents from large collections in response to natural language queries. Unlike traditional keyword-based approaches, semantic search connects the power of Transformer models to discern meaning, providing users with more contextually relevant and accurate results within seconds. This technology not only enhances the user experience by delivering superior matches from document collections but also lays the foundation for tackling more intricate tasks, like text summarization and question-answering. The research investigates the impact of semantic search on information retrieval efficiency and accuracy, comparing its performance with conventional methods. The findings presented herein not only showcase the immediate benefits of semantic search but also open paths for future research and development in natural language processing and its applications.**

Keywords—LangChain, ChatGPT, OpenAI, Deep Learning, Chroma DB, Vector Embeddings, Redis, Pinecone.

I. INTRODUCTION

There was a significant increase in the use of generative AI algorithms in 2022. The infamous chatbot, ChatGPT—an abbreviation for Chat Generative Pre-Trained Transformers—was published by OpenAI. The world has been surprised by this advanced AI system, which is driven by deep learning technology, because of its remarkable capacity to produce writing that appears human and hold conversations with people. Has anyone ever considered posing queries to computer document files as an alternative to doing manual searches for information inside them? This is made feasible by the development of conversational and generative AI technologies. ChatGPT, the notorious chatbot, was published by OpenAI[22][23]. Chat Generative Pre-Trained Transformers is what it stands for. The world has been shocked by this advanced AI system, which is driven by deep learning technology, because of its remarkable capacity to produce writing that appears human and hold conversations with people[24]. Information retrieval and search from PDF documents have become more difficult due to the growing use and prominence of digital products[29][30]. But LangChain, a ground-breaking tool based on NLP (Natural Language Processing), and LLM (Large Language Models) overcomes these difficulties. With the use of sophisticated NLP algorithms, LangChain streamlines the search process and information extraction from PDFs[31]. LangChain uses Streamlit, a web application framework that does not require knowledge of other web development frameworks such as CSS and HTML, to produce an interface that is easy for users to navigate. Models may be seamlessly deployed with Streamlit and need very little code work. Users may easily interact with PDFs with LangChain and Streamlit, which greatly improves the convenience of document search and retrieval.

II. RELATED WORKS

The goal of the literature review for the "LangChain PDF Query" research work is to examine pertinent studies and technological developments in the domains of linguistic models, AI (Artificial Intelligence), NLP, and query systems. Convolutional Sequence to Sequence Learning is a revolutionary method of modeling sequences with convolutional neural networks that shed light on how language patterns might be more accurately comprehended and represented[1]. This study fell under the category of sequence-to-sequence learning. As shown in attention mechanisms have revolutionized language models[2]. Advanced query systems were made possible by its transformative "transformer" architecture, which made attention-based models for a variety of NLP tasks extremely effective and fast. Investigating AI's potential uses in the legal field[3]. An open-source Python program called LangChain assists in establishing connections between external data sources and large language models. It increases the agentic and data-awareness of conversation models such as GPT-4 and GPT -3.5 [4-7]. Thus, LLMs may be fed fresh data that hasn't been trained on using LangChain. Numerous chains offered by LangChain abstract away the difficulties involved in communicating with language models. Creating vector embeddings and store vectors, requires many other tools, such as vector databases and models. Gracenote is an artificial intelligence system made to abide by rules. This study demonstrates how generative AI can be used to solve

challenging legal problems[25]. The fact that "Chat GPT-4" performed better as compared to GPT-3.5 in certain use scenarios, such as drug-related information searches, shows how language models for domain-dependent information retrieval are always improving. For data security and privacy, effective keyword search across encrypted cloud information is essential. a method for doing so that offers insights on safe information retrieving in cloud settings[8][9].

A combination and indexing technique that utilizes feature patterns and semantic evaluation to address the difficulties associated with huge data. This study provides useful methods for effectively organizing and querying big datasets[10]. A technology in the field of scientific literature that allows users to read scientific text and decipher information from written literature in PDF format. This tool shows how data extraction from scientific publications has advanced. It is crucial to consider the tools and platforms that are accessible to carry out the research work. While Python LangChain gives comprehensive instructions for embedding language models into programs, Streamlit offers an intuitive interface for data visualization and interactivity. GPT-3.5 and other OpenAI models form a vital basis for language-based AI applications[11]. An efficient query method utilizing LangChain and language models was described, which can be useful for creating the LangChain PDF Query. To sum up, this summary of literature has given readers a thorough grasp of the studies and technological advancements pertinent to the "LangChain PDF Query" research work. It addressed massive data processing, safe cloud data search, domain-specific information retrieval, attention-based mathematical models, AI in legal compliance, sequence-to-sequence learning, and critical implementation tools. These insightful observations will direct the creation of a successful query system that leverages LangChain and language models, advancing the AI-driven processing of languages[12].

Large Language Operations revolve around text embeddings. Although storing and retrieving genuine language is extremely wasteful, theoretically works with language models that contain natural language. For instance, quick searches can be done across substantial data sets in the research work[13][26]. Such processes cannot be carried out on natural language data. Convert text data into vector forms to increase its efficiency. To create embeddings from texts, specific machine learning models exist[14]. The texts are transformed into vectors with several dimensions. Once incorporated, these data may be sorted, searched, and grouped among other things. To determine how closely two phrases are connected, compute their distance from one another. The finest aspect is that, unlike standard database searches, these procedures capture the semantic proximity of two phrases rather than being restricted to keywords[15–17]. ML has greatly increased its power. For all of the popular vector databases, including Alpine dB, Pinecone, Redis, and Chroma, LangChain offers wrappers. In addition to supporting OpenAI models, Cohere's models—GPT4 ALL, an open-source substitute for GPT models—are also supported by LLMs. Wrappers for HuggingFace, Cohere, and OpenAI embeddings are provided[18][19]. These embedding models are also applicable. Thus, to put it briefly, LangChain is a meta-tool that simplifies the many complexities involved in connecting with underlying technologies, thereby facilitating the rapid development of AI applications by anybody[20][21].

III. PROPOSED WORK

This research influences the OpenAI embeddings model for the creation of embeddings and the deployment of AI application aimed at end-users, popular open-source models like Hugging Face models and Google's Universal Sentence Encoder are adopted. The storage of vectors is facilitated by Chroma DB, an open-source vector store database. In the development of a LangChain, the Conversational Retrieval Chain is employed, specifically designed to enhance conversational interactions with chat models that incorporate a historical context. This research work details the seamless integration of these components, showcasing their collective efficiency in constructing a robust and context-aware conversational AI application for real-world use as in Fig.1.

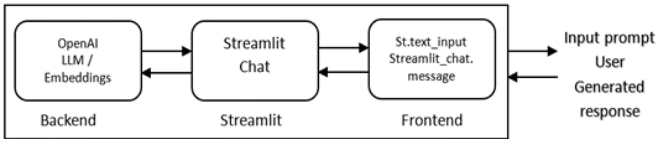

Fig. 1. PDF Chat Model

The graphical user interface used is Streamlit, to process the backend data, create embeddings, and respond from the large language model that is coming from the OpenAI API. However, it can easily be replaced with open-source models. The user will upload a PDF file and then it will accept prompts from the user as input and generate responses based on the LLM and embeddings that were used.

Having a bunch of PDF files and want a conversation with them is just like having a conversation with ChatGPT. The framework LangChain, which powers language model applications, allows to link with the language model API to external data sources and subsequently enables the language model to communicate with its surroundings, as seen in Fig.2. So, in this case, the data sources can be used with any model from OpenAI along with all the recently open-sourced models such as Lamma alpaca or GPT[26]. It simply reads that PDF file and extracts data from it. If there are 100 or 200 pages in the document cannot fit into the large language model because it will hit the token name. So, divide it into smaller chunks such that the length of the chunk is smaller than the token size.

The document is divided into 10 different chunks and each 10 chunks is mapped into their corresponding embeddings. An embedding is a vector a list of floating-point numbers and essentially it works as a compression algorithm[27]. So, let us say each text Chunk has a thousand characters, but using embeddings it can use it to reduce it to a much smaller size let us say the embedding size is only three[28]. So that will be the compression that is performed here.

OpenAI's text embeddings are used here. With the help of embeddings rather than comparing text directly the embeddings can be compared simply and see which two different texts are closer to each other. Based on these embeddings the knowledge base is created. So based on the embeddings of the documents that have been stored, will get the results and it will be ranked based on the closeness or relatedness to the query. So, will get the results here, and then it will use a generative large language model to generate a response agenda back to the user. The document indexing system, document embedding techniques, semantic search techniques are implemented for improving the search experience in chat GPT for PDf files with LangChain.

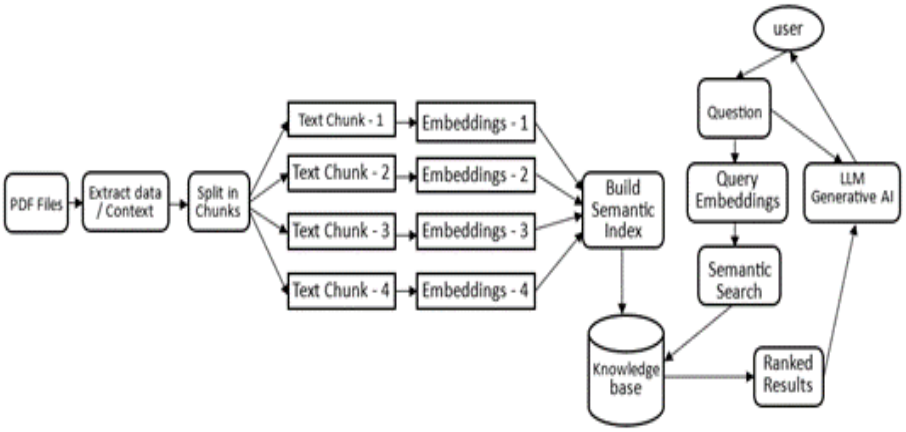

Fig. 2. PDF Chat Architecture

IV. RESULTS AND DISCUSSION

In the chat PDF UI tool designed, simply drag, and drop the PDF file, for example, to upload the constitution of India PDF file drag and drop it and simply interact with it, like querying how many states and union territories are there in India, based on the data, that are available in the PDF file it comes up with an answer. It provides information on the current number of states and union territories in India like, there are 28 states and 8 union territories in India.

A. Word Embedding

In the context of word embeddings, dimensionality reduction refers to methods for reducing the number of vector-based features while keeping the essential semantic information that the embeddings capture. Reducing computing complexity, justifying the effects of dimensionality, and enhancing the usability of the embeddings are only a few advantages of this method.

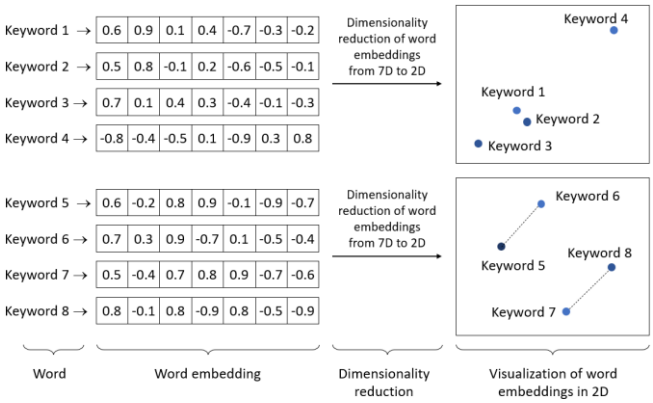

Fig. 3. Word Extraction and mapping of the features

The need to convert text information into vector forms to increase its efficiency. To create embeddings from texts, certain machine learning models exist. The texts are transformed into vectors with several dimensions. Once incorporated, these data can be sorted, searched, and grouped among other things as shown in Fig.3. To determine how closely two sentences are related, it can compute their distance from one another. The finest feature is that, unlike standard database searches, these operations capture the semantic proximity of two sentences rather than being restricted to keywords. Its power has increased significantly because of machine learning.

B. Develop a Chat Interface

The application's UI will have two main features: a chat window and a text box to receive end users' OpenAI API keys is another addition. The web application's Row and Column interfaces enable the alignment of several components, which will help to personalize the online interface.

C. Streamlit

With the help of the open-source library Streamlit, it can quickly and effectively create unique web apps for ML and Data Science in the research work. This framework makes it simple to create interactive visualization models, dashboards, and plots without requiring to worry about the backend deployment architecture or underlying web framework.

Additionally, it allows users to add widgets, which facilitates user interaction with the web application and the models utilized. Popular Python and ML libraries like TensorFlow, Scikit-learn, Seaborn, Matplotlib, Pandas, and NumPy are all integrated into this framework, allowing us to swiftly create and deploy the interface. Building dynamic data apps with Streamlit is simple and requires little coding because of its user-friendly interface.

Streamlit is designed for quick prototyping, enabling data scientists and developers to try out different ideas and produce fully working products. Computational workflows are facilitated and expedited by the data cache. Streamlit enables multiple people to collaborate in real-time on the same research work at the same time. Among the many interaction elements that Streamlit provides are sliders, dropdown menus, and checkboxes, which allow for real-time data manipulation and exploration.

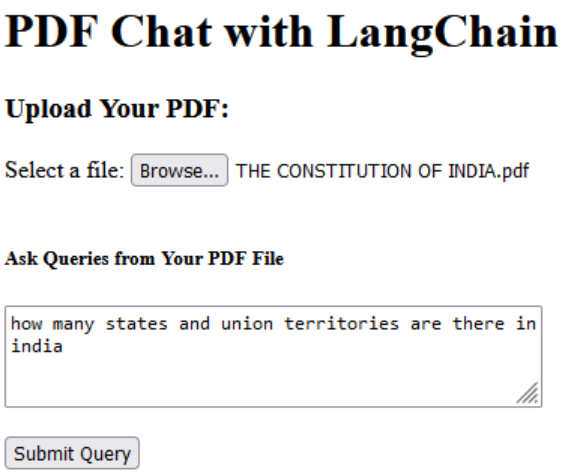

Fig. 4. PDF Chat UI Tool

This is exactly what the web tool's UI would seem. The individual can upload a file on their device that is smaller than 200 megabytes by clicking on Browse Files. After processing for a few minutes, it will receive an additional input box where it can submit the query as shown in Fig.4.

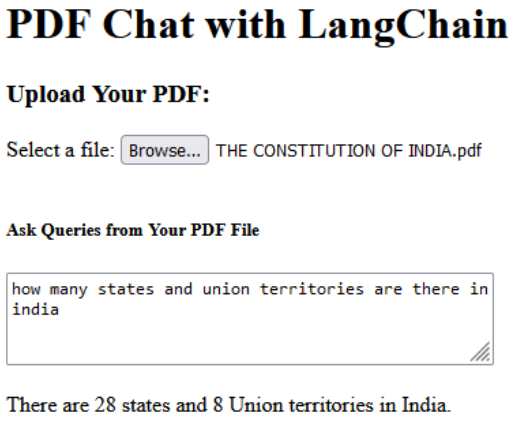

Fig. 5. Processed Query with PDF Chat with LangChain

With the input inquiry box now in place, queries about the PDF are submitted as Fig.5. and Fig.6. The PDF used here discusses the constitution of India. Now distinguish between queries and ask various inquiries, such as "How many states and union territories are there in India?". After reviewing the file, the Large Language Model provides an accurate response to the specified query.

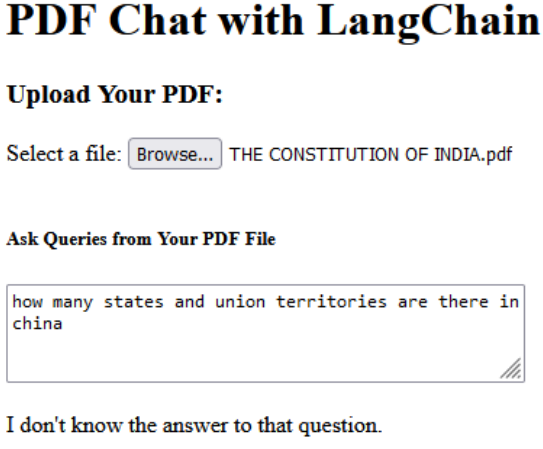

Fig. 6. Processed Query not related to the PDF file

The result "I don't know the answer to that question" signifies that the PDF produced by the Large Language Model was unable to offer an accurate answer to the question as shown in Fig.6. Even after examining the file's contents, the model was unable to provide a sufficient response to the inquiry. This tool uses Streamlit, Large Language Models, and LangChain to make it easier to extract important information from PDFs. This creative approach saves time and effort by greatly streamlining and improving the procedure, enabling users to retrieve any necessary information from PDF documents. The application enhances the efficiency and precision of the querying process by incorporating LangChain technology, rendering it a priceless resource for anyone interacting with PDFs. Users may quickly extract pertinent data, increasing productivity and lowering the amount of labor-intensive manual work often needed to analyze PDF documents. The total user experience is further improved by Streamlit user-friendly UI and intuitive features. This online application changes the way that PDF querying is done by enabling users to navigate and extract information from PDFs with efficiency.

V. CONCLUSION

Through this research work, experienced LLMs, their drawbacks, and how the LangChain framework addresses some of these issues by creating a custom LangChain PDF chatbot in this research work. There is no indication that the current wave of Generative AI Technologies will fade anytime soon. The flexible design of LangChain PDF chatbots makes it possible to overcome some of the drawbacks of conventional LLMs while creating custom ones. By implementing the chat functionality to query a PDF document using LangChain and the OpenAI API, by leveraging text splitting, embeddings, and question-answering capabilities, users can be provided with an interactive chat interface to extract information from PDFs. This approach can be extended and customized based on specific requirements and can be a valuable tool for information retrieval and knowledge extraction from PDF documents.

REFERENCES

[1] M. Liebrenz, R. Schleifer, A. Buadze, D. Bhugra, A. Smith, "Generating scholarly content with ChatGPT: ethical challenges for medical publishing", Lancet Dig. Health, 5 (3) (2023), pp. e105-e106.

[2] Pradeep, K., & Jacob, T. P. (2016, December). Comparative analysis of scheduling and load balancing algorithms in cloud environment. In 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) (pp. 526-531). IEEE.

[3] Taecharungroj, "What can ChatGPT do?" Analyzing early reactions to the innovative AI chatbot on Twitter", Big Data Cogn. Comput., 7 (1) (2023), p. 35.

[4] R. Kinoshita, S. Shiramatsu," Agent for recommending information relevant to the web-based discussion by generating query terms using GPT-3", 2022 IEEE International Conference on Agents (ICA), IEEE (2022, November), pp. 24-29.

[5] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, T.Y. Liu, "BioGPT: generative pre-trained transformer for biomedical text generation and mining", Briefings Bioinf., 23 (6) (2022)

[6] M. Abdullah, A. Madain, Y. Jararweh," ChatGPT: fundamentals, applications, and social impacts", 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS), IEEE (2022, November), pp. 1-8.

[7] Jacob, T. P., Pravin, A., & Asha, P. (2018). Arduino object follower with augmented reality. Int. J. Eng. Technol, 7(3.27), 108-110.

[8] Chan, "GPT-3 and InstructGPT: Technological Dystopianism, Utopianism, and "Contextual" Perspectives in AI Ethics and Industry", AI and Ethics (2022), pp. 1-12.

[9] Jacob, T. P., & Ravi, T. (2013). An optimal technique for reducing the effort of regression test. Indian Journal of Science and Technology, 5065-5069.

[10] H. Zhu, P. Tiwari, A. Ghoneim, M.S. Hossain, "A collaborative ai-enabled pre-trained language model for aiot domain question answering", IEEE Trans. Ind. Inf., 18 (5) (2021), pp. 3387-3396.

[11] Varun, K. S., Puneeth, I., & Jacob, T. P. (2019, April). Virtual mouse implementation using open CV. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 435-438). IEEE.

[12] R. Yan, X. Jiang, D. Dang, "Named entity recognition by using XLNet-BiLSTM-CRF", Neural Process. Lett., 53 (5) (2021), pp. 3339-3356.

[13] Jacob, T. P., & Ravi, T. A. (2014). A novel approach for test suite prioritization. Journal of Computer Science, 10(1), 138.

[14] A.M. Sowjanya, "Self-supervised model for speech tasks with hugging face transformers", Turkish Online J.Qualit. Inq., 12 (10) (2021).

[15] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

[16] Jacob, T. P., & Ravi, T. (2013). Optimization of test cases by prioritization. Journal of computer science, 9(8), 972.

[17] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are few-shot learners. OpenAI. Retrieved from https://openai.com/research/musenet

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[19] Sophia, J. J., & Jacob, T. P. (2021, August). Edubot-a chatbot for education in covid-19 pandemic and vqabot comparison. In 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 1707-1714). IEEE.

[20] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized BERT approach. arXiv preprint arXiv:1907.11692.

[21] Krishnadoss, P., & Jacob, P. (2019). OLOA: Based Task Scheduling in Heterogeneous Clouds. International Journal of Intelligent Engineering & Systems, 12(1).

[22] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Brew, J. (2019). Hugging Face's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771.

[23] Jacob, T. P., Pravin, A., & Kumar, R. R. (2022). A secure IoT based healthcare framework using modified RSA algorithm using an artificial hummingbird based CNN. Transactions on Emerging Telecommunications Technologies, 33(12), e4622.

[24] Khandelwal, U., He, H., Qi, P., & Jurafsky, D. (2018). Sharp nearby, fuzzy far away: How neural language models use context. arXiv preprint arXiv:1805.04623.

[25] Hinduja, D., Dheebhika, R., & Jacob, T. P. (2019, April). Enhanced Character Recognition using Deep Neural Network-A Survey. In 2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0438-0440). IEEE.

[26] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Polosukhin, I. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

[27] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language understanding. arXiv preprint arXiv:1910.13461.

[28] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[29] Phang, J. R., Févry, T., & Bowman, S. R. (2018). Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. arXiv preprint arXiv:1811.01088.

[30] Choi, E., & Choi, J. (2020). Does BERT understand negation? Probing the linguistic abilities of language models. arXiv preprint arXiv:2002.03501.

[31] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237.