2024 Brazilian Symposium on Robotics (SBR), and 2024 Workshop on Robotics in Education (WRE) | 979-8-3503-9108-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/SBR/WRE63066.2024.10837977

LLM Text Generation for Service Robot Context

Lázaro Q. Silva[1] Ana Patricia F. M. Mascarenhas[1] Marco A. C. Simões [1]; Ivanoé J. Rodowanski[1]
Jorge Alberto P. de Campos[1] Josemar Rodrigues de Souza[1] and Jose Grimaldo da Silva Filho[1]

***Abstract*— Service robots are designed to perform useful activities for humans or other machines, such as cleaning a house, providing guidance, or cooking. To perform these activities, robots must be increasingly autonomous and capable of executing physical and cognitive tasks. Such skills involve human-robot interaction, where humans give a specific command, and the robot performs the task. In this scenario, voice recognition and detection tools can ensure communication through natural language. However, obtaining comprehensive and contextualized answers to specific questions or topics becomes challenging for the robot, which relies on static and pre-existing information to generate rigid and repetitive responses. This limitation hinders the creation of a more natural and intuitive dialogue between humans and machines. This paper proposes the LLP4ServiceRobot solution, which uses fine-tuned large language models to generate more dynamic texts and responses. The development methodology involves selecting large language models, fine-tuning them, and training them for text generation and question-answering tasks. We validated our model using two performance metrics: the F1-Score, for the question-answering model; and ROUGE, for the text generation model. Results showed that the question-answering model achieved an F1-Score of 52%, while the text generation model achieved a ROUGE-1 score of 67%. Additionally, an experiment was conducted to evaluate the Question Answering (QA) model. Results showed that the model could correctly answer 60% of the questions in a test dataset. The proposed model will be used in the service robot BILL, part of the RoboCup@Home league research at ACSO - the Center for Research in Computer Architecture, Intelligent Systems, and Robotics at the State University of Bahia - UNEB.**

I. INTRODUCTION

Robots can be seen as programmable devices with a certain degree of autonomy that operate in the real world to perform tasks [1]. The development of robots aims to facilitate and transfer repetitive or inhospitable tasks to machines. There are different categories of robots, such as industrial, medical, and assistive. Among them, service robots stand out for assisting humans or other machines in dangerous, dirty, or unpleasant tasks, such as firefighting, victim rescue, and repetitive domestic chores [2].

Service robots require a high degree of interaction with humans and often use speech in communication. Speech has been explored in several commercial applications (e.g. virtual assistants), but much research is still necessary to improve it. Robots, for example, generally limit their communication capabilities to understand specific commands and respond to questions within a pre-trained context.

The process of a robot understanding and responding to voice commands involves capturing the sound, converting it into a digital signal, voice recognition, natural language processing (NLP) [3], mapping the intent to a specific action, and providing feedback to the user. To improve this communication, techniques enabling a more complete and natural conversation between robots and humans are being researched, among which the use of Large Language Models (LLMs) [4] has shown promise.

LLMs are machine learning algorithms [5] designed to process and understand human language, capable of generating coherent text, answering questions, and performing automatic translations, among other NLP-related tasks [6]. Trained with large amounts of data, these models learn patterns, relationships, and meanings of words and phrases, enabling increasingly natural and efficient interactions, as observed in applications like ChatGPT [7].

In robotics, the use of Large Language Models (LLMs) can significantly enhance robots' ability to understand and respond to complex commands. These models, trained on vast amounts of textual data, allow robots to process natural language more accurately and improve human-robot interaction [8]. To contribute to this scenario, this paper presents the *LLM4ServiceRobot* solution, a proposal for optimizing LLMs for text generation and question answering in the context of service robotics. The proposal involves selecting the model and subsequently fine-tuning and training it with a focus on text generation and question answering.

To validate the *LLM4ServiceRobot* solution, we used two performance metrics: F1-Score for the question and answer model, and ROUGE for the text generation model. Additionally, an experiment was conducted using the *Likert* scale as a real-world evaluation of the *QA (Question Answering)* model's usage.

The remainder of the text is organized as follows: Sections II and III respectively provide the context and state of the art related to using LLMs in NLP applications. Section IV presents the proposed solution, and Section V details the validation of the solution. Finally, Section VI presents the conclusion and future work.

II. BACKGROUND

Enhancing human-robot interaction in service robotics faces significant challenges, particularly when using Large

* ACSO is supported by UNEB, FAPESB, CAPES, and CNPq.

[1]Researchers are from the Center for Research on Computer Architecture, Intelligent Systems and Robotics(ACSO) of State of Bahia University (UNEB), Silveira Martins st., 2555, Cabula, Salvador, Bahia, Brazil. teambahiart@gmail.com ; https://www.acso.uneb.br/bahiart/

979-8-3503-9108-4/24/$31.00©2024 IEEE

Language Models (LLMs), due to the need for substantial processing power and the construction of representative databases. Understanding the models and techniques employed is crucial for adapting and fine-tuning these models to specific needs. This section introduces LLMs and explores how they influence tasks related to text generation and question answering.

A. Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of Artificial Intelligence (AI) focused on the interaction between humans and computers through natural language. It enables computers to understand, interpret, generate text, and communicate like humans [3]. Large Language Models (LLMs) represent a subfield of NLP that employs deep learning techniques to generate and understand language in an advanced manner [6]. These models have broad applicability, including text generation, machine translation, automatic summarization, sentiment analysis, and dialogue generation [9]–[11].

Training LLMs involve two stages: pre-training, where the model learns relationships and word probabilities from large amounts of unlabeled text, and fine-tuning, where it is refined for specific tasks with labeled data [12].

During inference, when the model is used to generate text or make predictions, it receives an initial context, also known as a prefix. It generates subsequent words based on the probabilities learned during training. The model considers the previous context, assigns weights to possible words, and selects the most likely word based on these weights. This process is repeated iteratively until the sequence is completed.

Transformer neural networks are an architecture focused on NLP, particularly in developing LLMs. The *Transformer* removes the recurrent layers in traditional neural networks like RNNs and relies heavily on the attention mechanism, allowing the model to capture relationships between words [13] efficiently.

The *Transformer* architecture consists of two main components: the encoder and the decoder. The encoder takes a sequence of words as input, represented as word vectors (*word embeddings*), and processes this information through several layers. Each layer performs two main steps: *multi-head attention*, which enables the model to focus on different parts of the sequence, and *feed-forward*, which applies a non-linear transformation to the word vectors. The decoder generates the output sequence using the context encoded by the encoder and a partial output sequence.

The *Transformer* architecture is parallelizable, suitable for training on Graphics Processing Units (GPUs), and efficient in processing long sequences, as well as capturing long-distance dependencies more effectively than Recurrent Neural Networks [14]. Combining Convolutional Neural Networks (CNNs) [15] with *Transformers* can be advantageous, leveraging CNNs' ability to extract local features and *Transformers*' ability to model global dependencies, which is useful in tasks such as time series forecasting [16].

B. Fine Tuning

Fine-tuning is a technique that modifies a pre-trained model to perform specific tasks, transferring the vast linguistic and contextual knowledge of these models to particular applications. This technique reduces the need to train models from scratch, saves computational resources, and significantly improves performance [17].

The *fine-tuning* process involves several steps. First, data preparation is required, creating a specific and labeled training dataset for the desired task. It is essential to ensure that the data is clean and properly formatted. Next, weight initialization uses the weights from the pre-trained model as a starting point, allowing the retention of the knowledge acquired during pre-training.

Training adjusts the weights and parameters of the model based on the labeled data, using optimization algorithms such as gradient descent. The learning rate and other hyperparameters need to be tuned to optimize performance. During training, the model is evaluated with previews of the generated responses, using metrics such as loss rate and accuracy to make necessary adjustments and avoid *overfitting*.

After training, the model is evaluated using a test dataset and scoring metrics, such as *F1-Score* and *ROUGE*, to ensure the effectiveness of the *fine-tuning* [18].

III. Related Works

This section presents works that use LLMs in the context of natural language processing applications. The observed works include the types and techniques associated with model creation, model applicability, and the dataset and parameters used for training.

A widely used model in current literature is *BERT* from *Google* [19]. This model is trained using a vast corpus of text that does not need to be labeled. Additionally, the training occurs both in supervised and unsupervised forms. The work by [20], for example, creates a new model based on *BERT* for providing legal information about COVID-19.

Another adopted model is GPT [21] from *OpenAI*, which has demonstrated remarkable accuracy, and its generative architecture enables continuous learning of representations and patterns. However, while *GPT-2* is available for free, *GPT-3* and beyond are paid. Moreover, fine-tuning an architecture like *GPT-3* requires significant hardware capability. As an alternative, the *GPT-2* model has proven effective, as it allows the model to learn the structure of language broadly and generally before being fine-tuned on a labeled dataset for specific tasks. Considering that labeled data may be scarce, this approach provides significant contextualization.

T5, or *Text-To-Text Transfer Transformer* [22], is a language model developed by *Google Research* that stands out for its "text-to-text" approach. Instead of treating different natural language processing (NLP) tasks as distinct problems, T5 adopts a unified approach where all tasks are formulated as text transformation problems.

The *BERT*, *GPT*, and *T5* models are the most frequently found in the literature. Although, to date, we have not found

these models associated with service robotics, their use in other areas highlights their potential contribution to robotics as well.

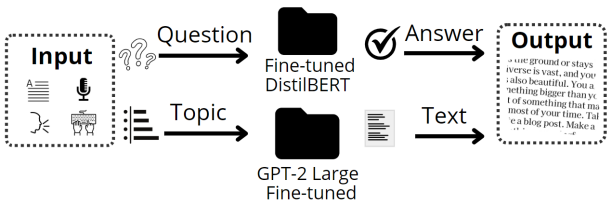
IV. SOLUTION *LLM4ServiceRobot*

This section presents the *LLM4ServiceRobot* solution, a proposal for using large language models to enhance human-robot interaction in the context of service robotics.

The *LLM4ServiceRobot* utilizes the *BERT* and *GPT-2* models, fine-tuned with specialized techniques. These models are pre-trained with parameters and contextual word representations tailored for service robots. The complete solution also involves integrating these models with *BILL* service Robot from the Center for Computer Architecture, Intelligent Systems, and Robotics (ACSO) at the Universidade do Estado da Bahia (UNEB).

Figure 1 illustrates an overview of the *LLM4ServiceRobot* solution: the interaction process begins with a human providing a command or question to the robot using voice input. This input is captured by a microphone and converted into text for processing. For testing purposes, it is also possible to directly input a command or question in textual format through a graphical interface. The input text is then processed by the *DistilBERT* and *GPT-2* models, which have been fine-tuned for the service robotics context. The *DistilBERT* model handles short questions and answers, such as *Who discovered Brazil?*, or basic robot commands, like *Get a glass of water*. The *GPT-2* model handles more detailed responses, represented by longer texts, such as *Talk about the Cold War*. Thus, at the end of the process, the robot produces the corresponding output, meaning a text response is generated. This text is synthesized by the robot and played through a speaker.

Fig. 1. Overview *LLM4ServiceRobot*



The core of the *LLM4ServiceRobot* solution consists of the fine-tuned *DistilBERT* and *GPT-2* models. The model fine-tuning process involved data acquisition, model configuration, and performance monitoring. The Google Colab development environment was used, which provides the necessary computational resources, as shown in Table I. The models were trained using the *Hugging Face Transformers* library and *PyTorch*, with hyperparameters adjusted to optimize accuracy and precision.

Tabela I
COLAB DEVELOPMENT ENVIRONMENT

Google Colab Environment Configurations	
GPU	T4 GPU
GPU RAM	15GB
System RAM	12.7GB
Disk	78.2GB

A. Fine Tuning GPT-2

The fine-tuning of *GPT-2* was performed using the pre-trained *gpt2-large* model and a specific dataset selected to evaluate the model's ability to generate coherent and relevant text based on pre-trained knowledge. During training, the weights of the self-attention layers and the feed-forward layers were adjusted to enhance text generation related to the provided topics. For example, the attention vector weights were modified to place greater emphasis on key topic-specific words, ensuring that the model maintained a more precise focus during text generation.

The *Hugging Face Transformers* library was used for training, which provides tools for efficient loading, training, and fine-tuning pre-trained models. This library enabled the access and configuration of the *GPT-2 large* model, and the *PyTorch* framework served as the primary environment for model training and hyperparameter tuning. Hyperparameters were defined based on empirical testing to optimize the model's accuracy and precision.

For the learning rate, several values were tested to find the most stable and efficient rate. The optimal value of 2e-4 was chosen because it maintained training stability. Lower values slowed down the learning process, while higher ones caused abrupt fluctuations and instability in model optimization, leading to divergence in output quality.

The AdamW optimizer was used to adjust the model weights during training, benefiting from weight decay to prevent overfitting. A weight decay rate of 0.05 was selected after experiments, balancing weight regularization with learning capacity. This choice was especially important since the large size of the *GPT-2-large* model, combined with the limited dataset, required careful regularization to avoid overfitting to the training data.

Dropout was applied as an additional regularization technique, where a fraction of neurons in a layer was randomly deactivated during each epoch. Empirical testing determined the most suitable dropout rate, ensuring that the model did not rely too heavily on specific features. This helped the model generalize well to new data.

To further optimize training, a linear learning rate scheduler with warm-up was used. The learning rate started at a very low value and gradually increased during the initial epochs until it reached the defined rate of 2e-4. This strategy ensured smooth optimization, preventing instability at the beginning of training while facilitating stable convergence over time.

B. Fine Tuning BERT

For *BERT*, we used data from the publicly available *SQUAD* dataset provided by Stanford University, which

contains pairs of contexts and questions. This dataset is widely recognized for evaluating reading comprehension and question-answering capabilities, making it ideal for training models to understand and respond to context-based queries effectively.

Data preparation included augmentation techniques, such as creating variations in questions and contexts. The data was then divided into three sets: training, validation, and testing, to evaluate the model appropriately.

The training was performed with *DistilBERT*, a smaller version of *BERT*. We used additional layers and applied regularization techniques, such as dropout, to prevent overfitting. Fine-tuning involved setting hyperparameters such as the learning rate and dropout probability to improve model performance.

V. LLM4ServiceRobot Assessment

This section presents the validation and results of the *GPT-2* and *BERT* models. The *GPT-2* model is employed for generating texts based on topics, while the *BERT* model is used for answering questions in the context of Question Answering (QA). Each model requires a validation approach that considers its functionalities and objectives.

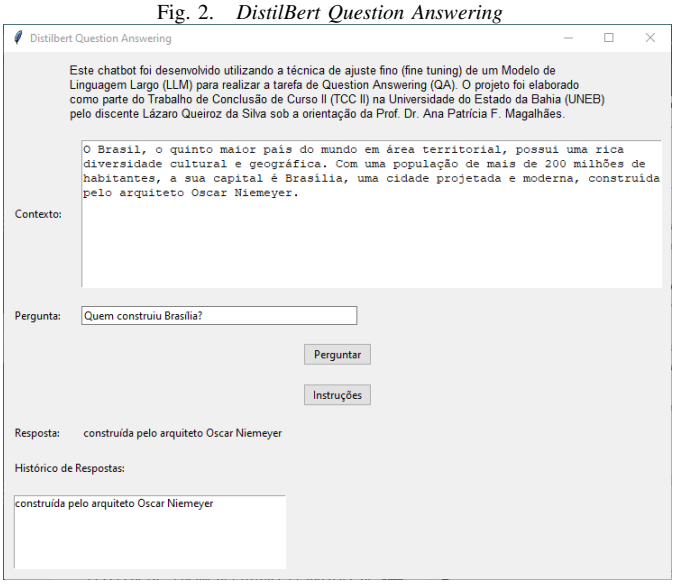
A. BERT Model Validation

This validation aimed to assess the quality of the answers generated in a real-world scenario and to determine the model's effectiveness in understanding and responding to specific questions.

The validation of the fine-tuned *BERT* model involved the following steps:

- Selection of the Test Dataset: A representative test dataset was initially selected, including questions and contexts relevant to applying the fine-tuned *BERT* model. A total of 10 (ten) questions were chosen, related to a specific context.
- Definition of the Likert Scale: This step of the model evaluation involved defining the Likert scale as an opinion-based validation regarding the quality of the generated answers. Ten people from the Information Systems course of the 2018.2 semester at UNEB University were selected. The evaluation was conducted through a form where each person defined a context to be provided to the model, and five questions related to the context were asked. Based on the generated responses, the participants rated them as correct, partially correct, incorrect, partially incorrect, or not answered. This evaluation provided insight into the model's performance in a real-world scenario. For this validation, an interface 2 was developed, similar to a chat, where participants entered the context and questions and received the answers as output.

Fig. 2. *DistilBert Question Answering*



Fonte: Author

- Definition of Metrics and Reference Answers: Reference answers were established for each question in the test dataset. Based on these reference answers, the *F1-Score* metric was calculated to assess the similarity between the model-generated answers and the reference answers. This enabled a comprehensive evaluation of the model's accuracy, integrating the subjective evaluation obtained through the *Likert* scale and the objective metrics provided by the *F1-Score*.

The validation demonstrated that the fine-tuned *BERT* model was able to generate relevant answers but also revealed areas for improvement. The *Likert* scale provided a subjective assessment of the responses, reflecting users' perceptions of the quality of the generated answers. On the other hand, the *F1-Score* metric offered an objective evaluation of the precision and coverage of the responses. For example, an identified improvement area was the need to enhance the accuracy of answers to more complex questions, which can be addressed by adjusting the model's hyperparameters and expanding the training dataset.

B. Validation of the GPT-2 Model

This validation aimed to assess the ability of the fine-tuned *GPT-2* model to generate relevant and coherent texts based on defined topics. The validation of the fine-tuned *GPT-2* model included the following steps:

- Selection of the Test Dataset: Initially, a representative test dataset was selected for the application of the fine-tuned *GPT-2* model. Ten topics were defined to be provided to the model, such as Service Robotics.
- Definition of Metrics and Reference Responses: Reference texts for the topics in the test dataset were defined. Based on these references, the *ROUGE* metric was calculated to evaluate the similarity between the texts generated by the model and the reference texts.

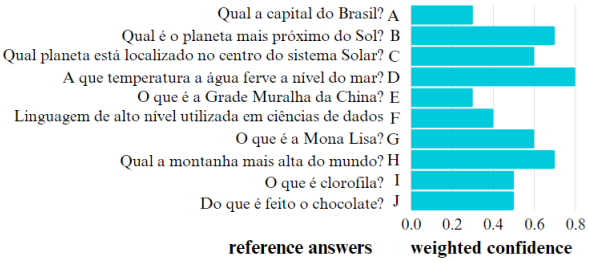
This allowed for a more comprehensive assessment of precision, taking into account objective metrics.

The validation concluded that the fine-tuned *GPT-2* model demonstrated good capability in generating relevant and coherent texts based on the defined topics. The *ROUGE* metric quantitatively evaluated the similarity between the generated texts and the reference texts, enabling a precise analysis of the model's response quality. The evaluation indicated that the model is well-tuned for topic-based text generation. However, areas for improvement were also identified, particularly regarding enhancing the dataset and experimentation with new parameters to optimize the model's performance further.

C. Results of the DistilBERT Evaluation using the F1-Score Metric

The evaluation of the *DistilBERT* model using the *F1-Score* metric initially showed an accuracy of 16.7% with a precision rate of 0.6. After fine-tuning the hyperparameters and utilizing the *SQuAD 2.0* dataset, accuracy improved to 52%. Fig. 3 illustrates the confidence levels of the model's responses compared to the reference answers. The solution answered most of the questions correctly. However, the graph in Fig. 3 shows the exact comparison of the solution's answers with the reference answers, which led to distortions in the confidence level. For example, in the first question "Who discovered Brazil", the reference answer was "Who discovered Brazil was Pedro Álvares Cabral" and the solution answered only "Pedro Álvares Cabral". This answer was considered different from the reference answer.

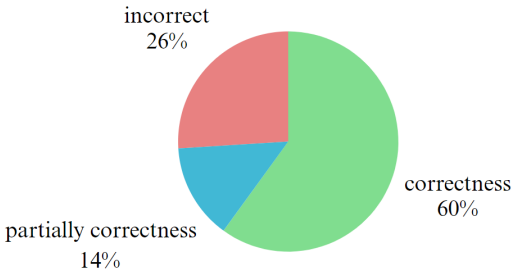
Fig. 3. Histogram of *BERT* Using the F1-Score Metric



D. Results of the DistilBERT Evaluation using the Likert Scale

Using the *Likert* scale with 10 students, the model achieved an accuracy rate of 60%, with 26% errors and 14% partial correctness. Fig. 4 shows the percentages of correct responses evaluated, reflecting an overall positive performance. The model did not produce partially incorrect responses or leave questions unanswered, indicating a satisfactory consistency in generating results. Responses partially incorrect are those that did not match the reference answer exactly (as explained in Section C).

Fig. 4. Results of the Experiment with the *Likert* Scale

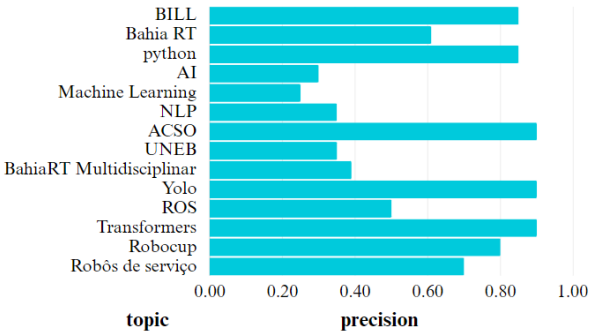


E. Evalution of GPT-2 using ROUGE metric

The *GPT-2* model was evaluated using the ROUGE-1 metric, with training conducted over 15 epochs and 14 reference topics. The average accuracy achieved was approximately 67% across all tested topics.

A histogram of ROUGE-1 accuracy for each topic showed that *ACSO*, *Yolo*, and *Transformers* had the highest precision (0.9), while *RoboCup* had a precision of 0.8. The lowest precisions were observed for *AI* and *Machine Learning* with 0.3 and 0.2, respectively. These results underscore the importance of using diverse and relevant datasets to improve the effectiveness of the *GPT-2* model.

Fig. 5. Histogram of *GPT-2* using the ROUGE-1 metric for the 14 test topics



VI. Conclusion and Future Works

This project proposed the use of large language models to enhance human-robot interaction in the context of service robotics. The *BERT* and *GPT-2* models were trained and fine-tuned.

For the validation of the *BERT* model, the results obtained were an accuracy of 52% based on the *F1-Score* metric and 60% in a qualitative experiment using the *Likert* scale. These results confirm the effectiveness of the model's fine-tuning for the *Question Answering* task.

For the *GPT-2* model, an average accuracy of 67% was achieved using the *ROUGE-1* metric for 14 topics, with some individual topics reaching an accuracy of up to 90% considering the reference contexts. This result highlights the potential effectiveness of fine-tuning techniques in improving the model's accuracy for text generation.

The results obtained demonstrated that the new interaction solution for the BILL robot allows for more comprehensive responses.

It is important to note that throughout the training and hyperparameter tuning process, significant challenges were encountered, with computational resource limitations being the primary difficulty. Modern language models, such as *BERT* and *GPT-2*, are complex and require substantial resources, including processing power, memory, and storage capacity. These resource constraints directly impacted the performance and scalability of the adjustments made. To overcome these limitations, optimizations were necessary during the training and fine-tuning of the models. This involved careful selection of appropriate batch sizes and choosing pre-trained models that best fit the available hardware constraints. Additionally, this project was developed on the *Google Colab* research and development platform, a cloud-based solution provided by *Google*. This strategic choice allowed access to scalable computational resources, including GPUs, at no cost, mitigating the hardware challenges encountered in the local environment.

Currently, we are testing new values for hyperparameters and using larger and more comprehensive datasets to understand their impact on model generation. Additionally, having an environment that provides more processing power is necessary for greater ease in performing these adjustments and generating the models.

REFERÊNCIAS

[1] M. Jörling, R. Böhm, and S. Paluch, "Service robots: Drivers of perceived responsibility for service outcomes," *Journal of Service Research*, vol. 22, no. 4, pp. 404–420, 2019.

[2] M. Jörling, R. Böhm, and S. Paluch, "Service robots: Drivers of perceived responsibility for service outcomes," *Journal of Service Research*, vol. 22, no. 4, pp. 404–420, 2019, disponível em: https://doi.org/10.1177/1094670519842334.

[3] K. Chowdhary and K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

[4] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[5] D. Glukhov, I. Shumailov, Y. Gal, N. Papernot, and V. Papyan, "Llm censorship: A machine learning challenge or a computer security problem?" *arXiv preprint arXiv:2307.10719*, 2023.

[6] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[7] T. Eloundou, S. Manning, P. Mishkin, and D. Rock, "Gpts are gpts: An early look at the labor market impact potential of large language models," 2023. [Online]. Available: https://arxiv.org/abs/2303.10130

[8] J. A. Gonzalez-Aguirre, R. Osorio-Oliveros, K. L. Rodríguez-Hernández, J. Lizárraga-Iturralde, R. Morales Menendez, R. A. Ramírez-Mendoza, M. A. Ramírez-Moreno, and J. d. J. Lozoya-Santos, "Service robots: Trends and technology," *Applied Sciences*, vol. 11, no. 22, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/22/10702

[9] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu *et al.*, "Summary of chatgpt/gpt-4 research and perspective towards the future of large language models," *arXiv preprint arXiv:2304.01852*, 2023.

[10] W. Jiao, J.-t. Huang, W. Wang, X. Wang, S. Shi, and Z. Tu, "Parrot: Translating during chat using large language models," *arXiv preprint arXiv:2304.02426*, 2023.

[11] N. Azzouza, K. Akli-Astouati, and R. Ibrahim, "Twitterbert: Framework for twitter sentiment analysis based on pre-trained language model representations," in *Emerging Trends in Intelligent Computing and Informatics: Data Science, Intelligent Information Systems and Smart Computing 4*. Springer, 2020, pp. 428–437.

[12] H. Schwenk and J.-L. Gauvain, "Training neural network language models on very large corpora," in *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 2005, pp. 201–208.

[13] D. Rothman and A. Gulli, *Transformers for Natural Language Processing: Build, train, and fine-tune deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, and GPT-3*. Packt Publishing Ltd, 2022.

[14] S. A. Marhon, C. J. F. Cameron, and S. C. Kremer, *Recurrent Neural Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 29–65. [Online]. Available: https://doi.org/10.1007/978-3-642-36657-4_2

[15] S. Indolia, A. K. Goswami, S. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network- a deep learning approach," *Procedia Computer Science*, vol. 132, pp. 679–688, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918308019

[16] J. T. Darmawan, J.-S. Leu, C. Avian, and N. R. P. Ratnasari, "Mitnet: a fusion transformer and convolutional neural network architecture approach for t-cell epitope prediction," *Briefings in Bioinformatics*, p. bbad202, 2023.

[17] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, "Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping," *arXiv preprint arXiv:2002.06305*, 2020.

[18] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[20] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping," *IEEE Robotics and Automation Letters*, 2023.

[21] T. Gong, C. Lyu, S. Zhang, Y. Wang, M. Zheng, Q. Zhao, K. Liu, W. Zhang, P. Luo, and K. Chen, "Multimodal-gpt: A vision and language model for dialogue with humans," 2023. [Online]. Available: https://arxiv.org/abs/2305.04790

[22] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683