IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 July 2024, accepted 28 July 2024, date of publication 1 August 2024, date of current version 13 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3436902

RESEARCH ARTICLE

Large Language Models and Rule-Based Approaches in Domain-Specific Communication

DOMINIK HALVONÍK[1] **AND JOZEF KAPUSTA**[1,2]
[1]Faculty of Natural Sciences and Informatics, Constantine the Philosopher University in Nitra, 949 01 Nitra, Slovakia
[2]Institute of Security and Computer Science, University of the National Education Commission, Krakow, 30-084 Kraków, Poland

Corresponding author: Jozef Kapusta (jkapusta@ukf.sk)

This work was supported in part by the Scientific Grant Agency of the Ministry of Education of the Slovak Republic and Slovak Academy of Sciences under Contract VEGA-1/0490/22; and in part by European Commission ERASMUS+ Program 2021, under Grant 2021-1-SK01-KA220-HED-000032095.

ABSTRACT Currently, we are once again experiencing a frenzy related to artificial intelligence. Generative Pre-trained Transformers (GPT) models are highly effective at various natural language processing tasks. Different varieties of GPT models are widely used these days to improve productivity. Graphic departments generate art designs, developers engineer intricate software solutions, leveraging services predicated on the GPT framework, and many other industries are also following the lead and implementing these new sets of tools in their workflow. However, there are areas in natural language processing where a simple solution is often more suitable and effective than current Large Language Models. In this article, we decided to analyze and compare the practical use of one of the more popular GPT solutions, J-Large, and the simple rule-based model we implemented. We integrated these two models into the internal information system of a private company focused on communication with customers in the gaming industry. Both models were trained on the same dataset provided as a log of conversational interactions for the last two years in the given system. We observed that GPT models exhibited superior performance in terms of comprehensibility and adequacy. The rule-based models showed noticeable proficiency in handling domain-specific tasks, mainly when fed with datasets extracted from the historical communication between users and a specialized domain system, such as a customer care department. As a result, with a sufficiently tailored and specific dataset at their disposal, rule-based models can effectively outpace GPT models in performing domain-specific tasks.

INDEX TERMS Chatbot, generative pre-trained transformers, large language models, transformer model, rule-based model.

I. INTRODUCTION

Recent advancements in Natural Language Processing (NLP) have yielded a suite of Generative Pre-trained Transformer (GPT) models capable of high-quality natural language generation (NLG). These models use a transformer architecture and extensive pre-training on vast text corpora to discern and learn language's inherent patterns and structures. Progressing from GPT-1 through GPT-4, we observe considerable enhancements in the capabilities and quality of NLG. Despite these strides, GPT models have shortcomings, necessitating comparative studies delineating their merits and demerits. Although researchers have conducted numerous studies comparing the performance of various GPT models [1], these often target specific applications or domains and generally require operation via public cloud service due to their high demand.

Moreover, the extensive data requisite for training these models requires extensive data sets. Also, the operational cost of these software applications is not sustainable from a longer-term perspective for medium and small-size institutions. Motivated by these factors, we elected to contrast

The associate editor coordinating the review of this manuscript and approving it for publication was Xiong Luo.

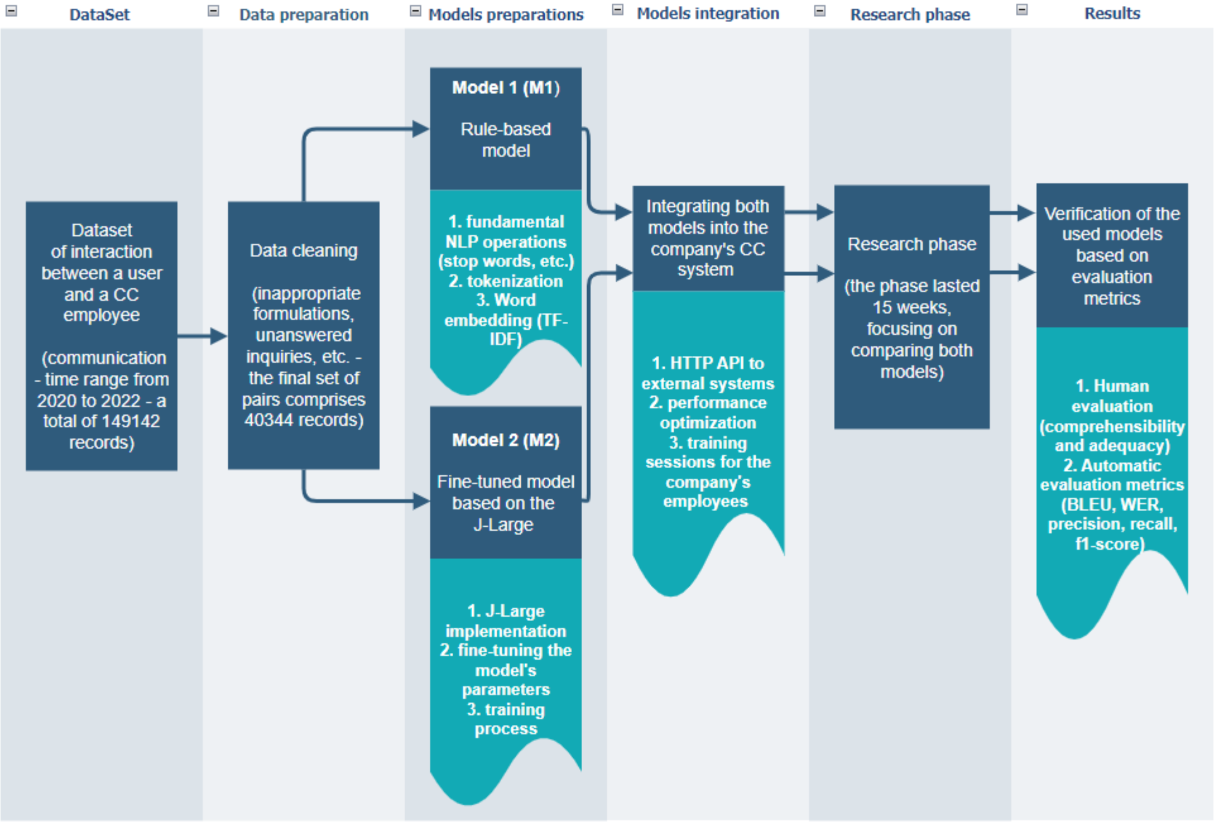

DataSet
Data preparation
Models preparations
Models integration
Research phase
Results
Dataset of interaction between a user and a CC employee
(communication - time range from 2020 to 2022 - a total of 149142 records)
Data cleaning
(inappropriate formulations, unanswered inquiries, etc. - the final set of pairs comprises 40344 records)
Model 1 (M1)
Rule-based model
1. fundamental NLP operations (stop words, etc.)
2. tokenization
3. Word embedding (TF-IDF)
Model 2 (M2)
Fine-tuned model based on the J-Large
1. J-Large implementation
2. fine-tuning the model's parameters
3. training process
Integrating both models into the company's CC system
1. HTTP API to external systems
2. performance optimization
3. training sessions for the company's employees
Research phase
(the phase lasted 15 weeks, focusing on comparing both models)
Verification of the used models based on evaluation metrics
1. Human evaluation (comprehensibility and adequacy)
2. Automatic evaluation metrics (BLEU, WER, precision, recall, f1-score)

FIGURE 1. The individual steps of the experiment.

the performance of a GPT model with a rule-based model under the realistic conditions of a corporate Customer Care (CC) department's information system. This empirical comparison allows us to examine if GPT models are necessary for every chatbot or if chatbots might handle specific tasks more effectively based on a different algorithmic principle with less data-intensive training requirements.

We conducted an experiment that validated and compared a GPT model with a rule-based model under real-world conditions of CC. Our experiment extended over six months and encompassed various tasks, including data collection from the preceding two years within the company hosting the experiment. This process generated a corpus of 40,344 question-answer pairs to train a customized fine-tuned model on the J-Large platform and our proprietary rules-based model. Throughout the experiment, CC department employees received responses generated by both models and chose an appropriate answer or rejected both, offering their response instead. We subsequently analyzed the collected data to determine the effectiveness of both models in generating suitable customer responses.

The aim of our experiment was to determine the suitability of both approaches for practical application in CC. We wanted to understand the advantages and disadvantages of both proposed solutions. Given the challenges in implementing a robust large language model (LLM) such as GPT, as well as the necessary financial costs, we sought to determine to what extent such a system can be fully replaced by a simpler solution – such as a rules-based model.

The individual steps of the experiment, including the steps for training, fine-tuning, and model implementation, are visualized in figure (fig. 1) and described in detail in Section III: Materials and Methods. We based our work on the previous communication between CC staff and customers over the past two years. This formed the foundation for our two created models, which we subsequently validated under real-world conditions of a commercial company.

Our article offers a perspective on language models in the context of practical applications in the domain of customer communication within the gaming industry. Currently, there are several articles comparing LLMs with each other. In our article, we aim to answer the question of whether LLMs can be replaced by a simpler model in practical applications. We compare LLMs with a simpler approach, the rule-based model. This comparison is based on the results from the real-world usage of both environments in commercial practice.

The structure of the paper is as follows. The current state of research in the area of the influence of window size and dimension size parameters is summarized in the second part. Spam datasets used in the research, as well as related text pre-processing techniques and text vectorization models used, are described in the third section. The most important results are summarized in the fourth section. Discussion and conclusions form the content of the last part of the paper.

II. RELATED WORK

Nowadays, it is quite normal for large corporations to strive for automating seemingly simple customer contact operations. Several analyses have confirmed that customers calling customer lines regularly try to get the same information, and only a tiny percentage of those need a ''custom'' approach to address their requirements [2] successfully. The same formula can be applied to a wide range of communication channels between a person and any institution, whether it is a student looking for an answer to when they can send their application or a postal service customer who does not know which service is best for sending their package. It is well-known that it is more cost-effective to create a technical tool to address these demands [3] than to employ hundreds of customer support agents who will communicate with people looking for specific information through chat or other means.

Predictions from 2022 even said that specific segments of the economy, such as banking, would use chatbots to interact with customers in 2022 for up to 90% of total customer-bank communication [4]. All this is possible through using neural networks and their training for NLG purposes.

A. GENERAL OPERATION OF TRANSFORMER MODELS

Several market leaders currently use trained neural networks to create individual chat instances for different purposes. Whether it is GPT-4, Jurassic-1 or Wu Dao 2.0, we are still talking about algorithmically related solutions. The differences between them are primarily in the trained set and additional operations done before the input is sent to the neural network and output is presented to the user.

These neural networks are defined as ''pre-trained'' neural networks, meaning they were not created to be used for a specific scenario. However, the mentioned models are from the Transformer models category. It is a neural network architecture developed by Google Brain in 2017 [5]. Models that fall into the Transformers category use a self-attention mechanism appropriate for understanding natural language. It should be mentioned that the introduction of the attention mechanism in 2015 caused a considerable breakthrough and enabled the creation of the first models of this type, such as GPT-1 or BERT, from Google in the following years. Attention is a function that calculates the probability of the occurrence of another word surrounded by others.

B. OPENAI MODELS

The newest GPT-4 transformer model, like its numerical predecessors, was implemented by the research organization OpenAI and is considered the gold standard in the industry. The organization was founded in 2015 and is considered a direct competitor of DeepMind [6]. Microsoft declared that it has agreed to exclusive GPT-X licensing [7].

TABLE 1. The ratio of datasets on the GPT-3 training set.

Dataset	Number of tokens	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60.00%	0.44
WebTex2	19 bilion	22.00%	2.90
Books1	12 billion	8.00%	1.90
Books2	55 billion	8.00%	0.43
Wikipedia	3 bilion	3.00%	3.40

GPT-3 is a third-generation, autoregressive language model that uses deep learning to produce human-like text. To put it more simply, it is a computational system designed to generate sequences of words, code or other data, starting from a source input called the prompt. It is used, for example, in machine translation to statistically predict word sequences. The language model is trained on an unlabeled dataset comprising texts, such as Wikipedia and many other sites, primarily in English and a few other languages. These statistical models must be trained with large amounts of data to produce relevant results. The first iteration of GPT in 2018 used 110 million learning parameters (i.e., the values a neural network tries to optimize during training). A year later, GPT-2 used 1.5 billion of them. GPT-3 used 175 billion parameters. Nowadays, GPT-4 uses 170 trillion parameters, which is a significant increase compared to GPT-3.5. This is expected to significantly improve the model's ability to generate coherent and contextually appropriate responses to text prompts and its overall language understanding and NLP capabilities [8].

The more parameters a model has, the more data is needed to train the model. According to the creators, the OpenAI GPT-3 model was trained on 45 TB of text data from several sources. Several data sets which are used to train the model are listed in table (Tab 1).

It is trained on Microsoft's Azure's AI supercomputer [9]. It is a costly training, estimated to have cost $ 12 million [10]. The selected approach is suitable for multiple use cases, not only chatbots but also summarization, grammar correction, email composition, translation, question answering and many more.

In 2020, the British journal The Guardian published an article written by GPT-3 based on the provided requirements [11]. The text was edited, and the article was sensationalist. However, it must be said that after the article's publication, a wave of criticism arose about the presentation of the text. Many leading AI figures have criticized The Guardian for misleading the general public. As examples, they referred to concepts such as ''good" and ''evil" in the article, which are, of course, concepts which GPT-3 is unable to grasp [12].

Based on the GPT-3 documentation, there are four main standard models (Tab. 2) currently publicly available [13].

TABLE 2. The ratio of datasets on the GPT-3 training set.

Engine	Description	Max Request	Number of parameters	Training Data
text-davinci-002	Most capable GPT-3 model. Can do any task the other models can do, often with less context. In addition to responding to prompts, it also supports inserting completions within text.	4000 tokens	175 billion	Up to Jun 2021
text-curie-001	Very capable, but faster and lower cost than Davinci.	2048 tokens	13 billion	Up to Oct 2019
text-babbage-001	Capable of straightforward tasks, high speed, and lower cost.	2048 tokens	6.7 billion	Up to Oct 2019
text-ada-001	Capable of very simple tasks, usually the fastest model in the GPT-3 series, and lowest cost.	2048 tokens	2.7 billion	Up to Oct 2019

TABLE 3. Basic overview of Jurassic-1 parameters in each engine.

Engine	Description	Number of parameters
J1-Jumbo	With 178B parameters, it is the largest and most sophisticated language model ever released for general use by developers. Jumbo is the most capable model in the J1 family but is also the slowest and most expensive to run.	178 billion
J1-Large	With 7.5B parameters, it is saller, faster and more affordable but overall less capable than Jumbo, though still very effective for many use cases.	7.5 billion

C. AI21 STUDIO MODELS

Jurassic-1 was implemented by a company called AI21 Labs. AI21 Labs is an Israeli company that was founded in 2017, including Prof. Yoav Shoham (Professor Emeritus at Stanford), Ori Goshen (Founder of CrowdX), and Prof. Amnon Shashua (Founder, Mobileye). In August 2021, the company announced it had trained and released two large NLP models, Jurassic-1 Large and Jurrasic-1 Jumbo, via an interactive web UI called AI21 Studio [14]. Like GPT-3, Jurassic-1 consists of auto-regressive models trained on a mix of English corpora that scales up to 178 billion parameters. It diverges, however, from GPT-3 in several important respects, such as the size of vocabulary and the depth/width ratio of the neural net. Jurassic-1 is based on Transformer architecture with the modifications proposed by Radford et al. [14]. Input tokens are converted to vector representation with a *nvocab-by-dmodel* embedding matrix and fed into the Transformer network. The architecture comprises *nlayers* Transformer layers using a hidden dimension *dmodel*, each equipped with a self-attention module with *nheads* attention heads of size *dhead* and a feed-forward module [15].

Based on the information from the AI21 Studio website [16], there are two main standard models publicly available, which are listed in table (Tab 3).

Jurassic-1 is offered via AI21 Studio, the company's new NLP-as-a-Service developer platform, a website and API where developers can build text-based applications like virtual assistants, chatbots, text simplification, content moderation, creative writing, and many new products and services [17].

D. BEIJING ACADEMY OF ARTIFICIAL INTELLIGENCE MODEL

In June 2021, researchers at the Beijing Academy of Artificial Intelligence (BAAI) announced the release of Wu Dao 2.0, a multimodal AI model capable of generating text indiscernible from human-crafted prose. Containing 1.75 trillion parameters, the parts of the machine learning model learned from historical training data, Wu Dao 2.0 is ten times larger than OpenAI's 175-billion-parameter GPT- 3, which was the newest model from OpenAI at that time [18].

Wu Dao 2.0 is the latest example of what former OpenAI policy director Jack Clark calls model diffusion, or multiple state and private actors developing GPT-3-style AI models. For example, Russia and France train smaller-scale systems via Sberbank and LightOn's PAGnol, while Korea's Naver Labs invests in the recently created HyperCLOVA. Clark notes that because these models reflect and magnify the data they are trained on, various countries care about how their cultures are represented in the models [19].

Wu Dao 2.0, which arrived three months after version 1.0's March debut, is built on an open-source system akin to Google's Mixture of Experts, dubbed FastMoE. Mixture of Experts, a paradigm first proposed in the '90s, keeps models specialized in different tasks within a more extensive model using a "gating network." BAAI says Wu Dao 2.0 was trained with 4.9 terabytes of Chinese and English images and text on supercomputers and conventional GPU clusters, giving it more flexibility than Google's system because FastMoE does not require proprietary hardware [20].

Wu Dao 2.0's multimodal design affords it various skills, including performing NLP text generation, image recognition, and image generation tasks. Given natural language descriptions, it can write essays, poems, and couplets in traditional Chinese, caption images, and create nearly photorealistic artwork. According to Engadget, Wu Dao 2.0 can also power "virtual idols" and predict the 3D structures of proteins, like DeepMind's AlphaFold [21].

E. GOOGLE MODEL

Google's GPT model BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement) [22].

One of the unique features of BERT is its ability to use multi-task learning to perform multiple NLP tasks simultaneously. The model can perform both generation and prediction tasks, making it a versatile tool for a wide range of NLP applications.

Google AI has utilized BERT in its research, including developing the Google Research Football Environment and creating an AI agent to play soccer using the model [23].

BERT's transformer-based architecture is built on the self-attention mechanism, allowing the model to weigh the importance of different input tokens and generate output tokens based on their relative importance [5]. The self-attention mechanism is effective in various NLP tasks, including machine translation and language modelling. The implementation into a practical application was carried out by creating a chatbot named BARD. BARD is now known as Gemini [24].

F. RULE-BASED MODELS

Rule-based models are straightforward when compared to learning-based models. There is a specific set of rules. If the user query matches any rule, the answer to the query is selected. Otherwise, the user is notified that the answer to the user query does not exist. One of the advantages of rule-based models is that they always give accurate results. However, on the downside, they do not scale well. To add more responses, new rules must be defined. Examples of rule-based model implementations are IBM Watson or Google Dialogflow [25].

The rule-based model is based on specific rules to answer the text given by humans. The Rule-based model may be based on a rule given by humans, but it does not mean that we would not use any dataset to create one [26].

Using a list of sentences measures the similarity between the query text we put into our chatbot and every single text in the list of sentences. Whichever result produces the closest similarity (for example, the highest cosine similarity) would become the chatbot answer.

Using cosine similarity, we would create a chatbot that answers the queries by measuring the similarities between the query and the corpus we developed [27]. This methodology is also referred to as the 'retrieval-based chatbots' approach. These systems evolve the capability to discriminate and select a fitting response to user queries.

A rule-based approach is currently being employed in several applications within the domain. For instance, it has been utilized in the development of therapeutic chatbots [28], [29] aimed at aiding customer support agents [30]. Additionally, educational institutions have adopted this approach to address diverse inquiries about academic matters, encompassing facilities, procedures, and policies [31].

Chen et al. [32] emphasize the frequent application of this concept in scenarios centered around information retrieval, where users seek information based on predefined constraints. An illustrative instance of a frame-based framework is when users provide specific details, such as their departure and arrival cities, when searching for a route. Nonetheless, such a framework may encounter challenges adapting to unstructured conversations [33] that deviate from predetermined plans. Moreover, the system's usability becomes increasingly challenging to enhance, particularly when user inputs fail to match any information within the database [34].

G. HYBRID MODELS

Given that both aforementioned approaches, rule-based models and large language models (LLMs), possess significant advantages as well as fundamental drawbacks in different areas, the need has arisen to combine these two approaches in order to leverage the strengths of both while minimizing their weaknesses to the greatest extent possible. Common features of such combinations involve using rule-based/intent-based model components to identify context or perform specific actions, and then formatting the output using LLMs to generate syntactically acceptable responses that simulate human.

This method is used in various fields, for example, in the medical field with Med|Primary AI assistant [35]. These tools in specified area are primarily utilized to enhance efficiency in patient diagnosis, where, as mentioned earlier, the selection of the diagnosis is handled by a rule-augmented AI-empowered system that incorporates a rule-based decision system, and the subsequent presentation of results to the patient is delivered by an LLM [36]. Despite the existence of numerous tools for creating chatbots [37], one of the most popular tools for hybrid models is RASA. Originally, it was a platform for creating only rule-based chatbots, but with the advent of more sophisticated LLMs capable of simulating human responses much more convincingly, the platform was enhanced to allow for the integration of LLMs via API at the user's discretion [38]. This approach is currently considered the most advanced in terms of response quality and accuracy, though it remains relatively expensive due to the necessity of having a custom LLM.

H. MODELS COMPARISON METHODOLOGIES

Numerous studies and articles compare NLG models in academic and professional literature. It is common for these comparisons to include results and characteristics from various platforms that are currently on the market. Watson from IBM, Dialogflow, LUIS, and RASA are the most well-known platforms. These platforms have proven effective in general question-answering services, as highlighted by the study conducted by Setyawan [39].

In addition, other research endeavours have focused on comparing services in the realm of Natural Language Understanding (NLU) that are publicly available now [40]. These studies share several common elements, particularly when it comes to the design and execution of specific experiments. Most of these studies start by defining a domain-specific corpus and subsequently create two or more chatbots powered by each tested service. After developing these chatbots, different prompts are executed to generate the desired outputs. The final evaluation of these outputs is then conducted using selected deterministic or stochastic methods, depending on the preferences of the research team [41].

The main point of divergence in these research works is the choice of a specific domain and the dataset size provided, as mentioned in the study by Liu in 2019 [42].

Due to these and other similarities, we decided not to develop our comparison framework. This decision was primarily motivated by the publication from Maroengsit, released in 2019 [43].

In a study by Pandey and Sharma [44], the researchers designed and implemented twelve chatbot variants comprising both rule-based and generative-based models. Their findings revealed that generative-based chatbots, leveraging the encoder-decoder model, exhibited superior performance compared to their rule-based counterparts. Generative-based chatbots offer more versatility by integrating additional layers, such as transformer structures and attention layers, while rule-based chatbots necessitate annotated input from a medical expert to function optimally.

III. MATERIALS AND METHODS

A. OVERVIEW

The focus of this paper is to experimentally compare the outputs of two language models that are based on different algorithmic foundations. To ensure a fair comparison, we aim to create similar conditions for both models, which will help us avoid bias when evaluating their performance. Through this comparison, we hope to gain a more comprehensive understanding of the strengths and weaknesses of each model and provide insights for future improvements.

B. DATA PREPARATION

In the beginning, it was necessary to extract a sufficiently high-quality dataset. Based on consultations with employees of the CC department, we have decided that even though the system had data starting from 2013, a more relevant sample would only comprise of the time range from 2020 to 2022.

We were able to extract a total of 149142 records using this approach. Each record represents one interaction between a user and a CC employee of the respective company. However, due to many records containing vulgar language or inappropriate formulations, we removed all records containing words from the provided list from the dataset through simple pattern matching. Additionally, we excluded all unanswered inquiries. By doing so, we arrived at a final set of pairs comprising 40344 records. The overall number of languages present in the dataset was 16.

After assembling the final dataset, we modified it so that both the user input and the employee response were in English. We took this step primarily due to the dataset's low representation of specific languages. For example, Finnish had only seven records. We opted for this adjustment to create a more robust dataset with regard to the context. The translation was performed using the Google Translate cloud service. We developed a simple PHP program using the viniciusgava/google-translate-php-client library. The dataset was used as a training set for both compared models. In this way, 80688 posts were translated into the target language, in our case, English.

C. MODEL PREPARATION

It is evident that perhaps the most critical phase for obtaining relevant results is the model training phase. If we were to differentiate the mentioned process in any way between the two models, the results of our research would become biased. Therefore, we have decided to conduct the training similarly for both models, even though using a unified approach may not be optimal for performance.

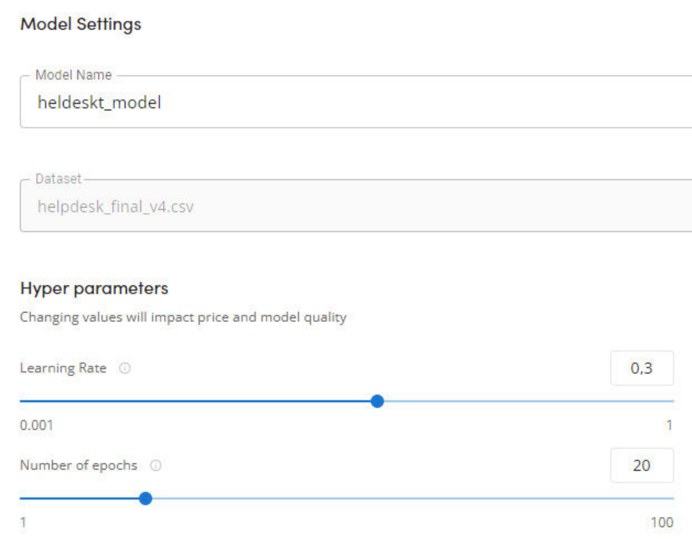
Maintaining consistency in the training process can ensure fairness and comparability between the two models. While it may come at the cost of performance optimization, it provides a solid foundation for conducting unbiased research and obtaining reliable results.

Adhering to the instructions provided in the documentation ensures that our dataset is appropriately prepared and compatible with the fine-tuned model based on the J-Large model. By following these guidelines, we help ensure the correct functioning and training of the model based on our specific requirements. Interestingly, the system divided the dataset into 500 test sets by default. We have done the same in our rule-based model based on this default behaviour.

Afterwards, we selected the basic parameters for training the fine-tuned model. Since no guidelines define the optimal settings for such a model in the chosen domain, we set the number of epochs to 20 and the learning rate to 0.3.

While there are no definitive rules for determining the ideal configuration, selecting initial parameter values is common when training a fine-tuned model. The chosen values of 20 epochs and a learning rate of 0.3 provided a starting point for training the model and were adjusted based on the performance and convergence of the training process (fig. 2).

It is important to note that fine-tuning the model's parameters may require experimentation and iterative refinement. Monitoring the training progress, evaluating model performance, and adjusting as necessary will



Model Settings

Model Name: heldeskt_model

Dataset: helpdesk_final_v4.csv

j1-large

Hyper parameters

Changing values will impact price and model quality

Learning Rate: 0,3

Number of epochs: 20

FIGURE 2. **Fine-tuned model configuration in web UI.**

help achieve the best results for specific domains and datasets.

The training process for our fine-tuned model took several hours. After completing the training phase, the prepared model was available as an API. Having the trained model accessible through an API allows us to utilize its capabilities and integrate it into the corporate information systems of the CC department We can send requests to API, provide input data, and receive model-generated responses based on the learned patterns and knowledge from the training process.

For the Rule-based model, the execution of fundamental NLP operations was necessary. Previously, helpdesk employees' responses to user inquiries were available. Basic NLP methods were applied to all these queries.

We cleaned the data and dropped null values in the pre-processing part of our work. We have removed the stop words. Stop words are a set of commonly used words in any language. In NLP, stop words eliminate unimportant words, allowing methods to focus on critical ones instead. The next pre-processing operation was tokenization. This step is fundamental in traditional NLP methods. Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords (n-gram characters).

The selection of the appropriate word embedding method is the most critical component. Word embedding or word vector is an approach to represent documents and words. Vectors created from documents are input vectors for creating, training, and testing classifiers in NLP classification tasks. In our case, word vectors are utilized to calculate the similarity between a user's question and previous questions in the dataset. The Term Frequency–Inverse Document Frequency (TfIdf) method was used for word embedding. It is a technique to quantify a word in documents. We generally compute a value for each word that signifies its importance in the document_and_corpus.

TfIdf is a traditional technique leveraged to assess the importance of tokens to one of the documents in a corpus [45]. The TfIdf approach creates a bias in those frequent terms highly related to a specific domain, typically identified as noise, thus leading to lower term weights because the traditional TfIdf method is not explicitly designed to address large news corpus. Typically, the TfIdf weight comprises two terms: the first computes the normalized Term Frequency (Tf), and the second is the Inverse Document Frequency (Idf).

Let t be a term/word, d be a document, w be any term in a document, then the frequency of the term t is calculated as:

$$tf\ (t, d) = \frac{f(t, d)}{f(w, d)}, \tag{1}$$

where $tf\ (t, d)$ is the number of terms in the document d, and $f(t, d)$ is the number of all terms in the document. When calculating TfIdf, the number of all documents in which the term occurs is also considered. We denote this number as $idf\ (t, D)$– inverse document frequency, and we can express it as:

$$df\ (t, D) = \ln \frac{N}{\sum (d \in D : t \in d) + 1}, \tag{2}$$

where D is the corpus of all used documents, and N is the number of documents in the corpus. The formula of TfIdf can be written as

$$tfidf\ (t, d, D) = tf\ (t, d) \times idf\,(t, D), \tag{3}$$

Formula tf has various variants such as $log(tf\ (t, d))$, $log(tf\ (t, d) + 1)$. Similarly, the idf has several variants of the calculation [46]. In our experiments, we performed the TfIdf calculation using the scikit-learn library (https://scikit-learn.org) and this method was used as the base method for comparison with the newly proposed methods.

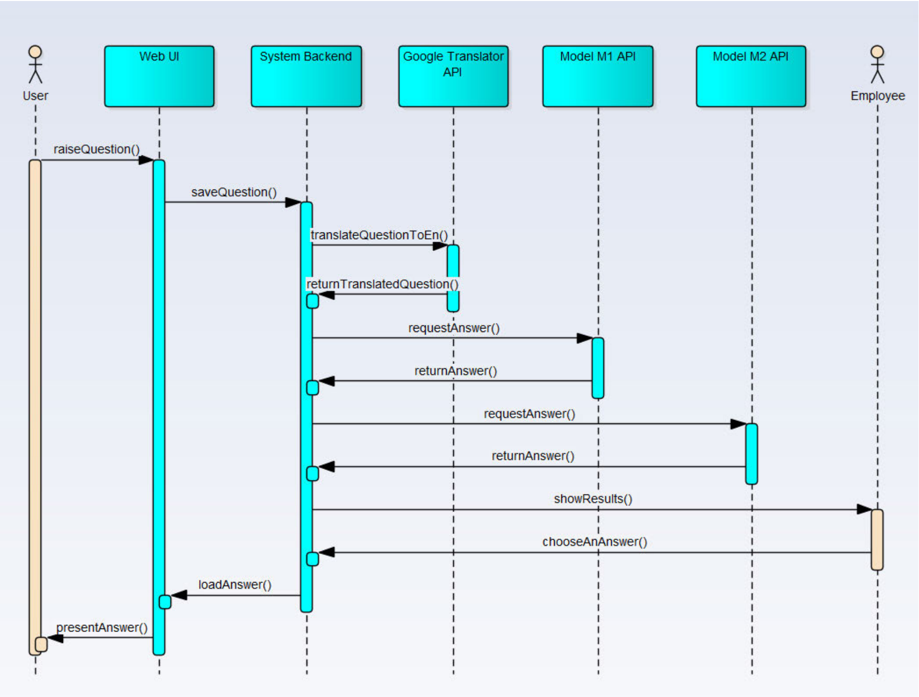


FIGURE 3. System communication representation.

After the preparation of the word embedding, the retrieval-based model was implemented. Using cosine similarity, we created a model that answers the queries by measuring the similarities between the query and the corpus we developed. Whichever result produces the closest similarity (for example, the highest cosine similarity) would become the model's answer.

Cosine similarity is a metric helpful in determining how similar the data objects are, irrespective of their size. In cosine similarity, data objects in a dataset are treated as vectors. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

The formula to find the cosine similarity between two vectors $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$ is:

$$cos\theta = \frac{\vec{\boldsymbol{a}}\vec{\boldsymbol{b}}}{\|\vec{\boldsymbol{a}}\| \left\|\vec{\boldsymbol{b}}\right\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}, \tag{4}$$

where: $\vec{\boldsymbol{a}}\vec{\boldsymbol{b}} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$ is the product (dot) of the vectors $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$.

D. MODEL INTEGRATION

Integrating both models into the company's CC system for communication with VIP customers was done through HTTP API calls to external systems representing the respective models. Both in the formulation of the HTTP request and in the HTTP response, we were limited by the capabilities provided by the respective system for the model M2, which was the application instance of the fine-tuned GPT model. The M2 model's API response was extensive and unnecessarily complex for our needs [47]. Therefore, for model M1, which was the application instance of the rule-based model, we decided to simplify the HTTP response to a minimalistic JSON format with only one attribute, "answer," which contained the preferred response to the submitted query (fig. 3).

Given that the communication between the company's system and the user did not occur in real-time, we did not have to consider performance optimization regarding response time for the individual models. Therefore, we created an API for the M1 model using the Python programming language and the Flask framework. The main reason was that the model was in the.pickle format, which we also compiled in the training phase via Python libraries.

E. DATA EXTRACTION

After integrating the mentioned application interfaces into the system, training sessions were conducted to instruct the company's employees on using and interacting with the system. The employees had several options for utilizing the new functionality:

1. Choose the full answer provided by the M1 model
2. Choose the full answer provided by the M2 model
3. Choose the answer provided by the M1 model and modify it if necessary
4. Choose the answer provided by the M2 model and modify it if necessary
5. Not choose any of the provided answers and write a custom response

These options allowed employees to tailor the responses according to their judgment and provide the most suitable assistance to customers. The system recorded and labelled each of the mentioned actions. This research phase lasted from 2022-09-20 to 2023-01-02. The timeframe can be precisely identified based on the first and last recorded entries.

This dataset captured the relevant information for analysis and evaluation, a comprehensive understanding of the system's performance during the specified research phase. Despite our efforts to provide consistent instructions to each employee to ensure a unified approach in determining the correctness or incorrectness of the answers, it is vital to acknowledge the role of human factors in this phase. For example, during the data evaluation after one month, we observed that some employees had stricter criteria for determining the answers' correctness than others.

While human judgment and individual variations may have influenced the evaluation process to some extent, the overall findings and outcomes of the research were not significantly affected. The primary focus was on the comparison and performance analysis of the M1 and M2 models, considering their respective answers and the final accuracy of the responses provided to users. It is also worth mentioning that selected employees do not know which answer is from which model. The system also randomly switched the order of the provided answers from the models. So, from the employee's perspective (Web GUI), sometimes the answer from M1 was presented first, and other times it was presented as the second option in the Web GUI.

IV. RESULTS

We verified the suitability of the used models with the help of automatic evaluation metrics and by ascertaining the opinions of CC employees who worked with both models. Both application instances were deployed for 15 weeks at the CC department. Human evaluation was created by senior CC employees who evaluated:

- comprehensibility
- adequacy.

During the monitored period, 6,550 responses were sent by employees. A simple graph (fig. 4) shows the number of answer choices recommended by models M1 and M2.

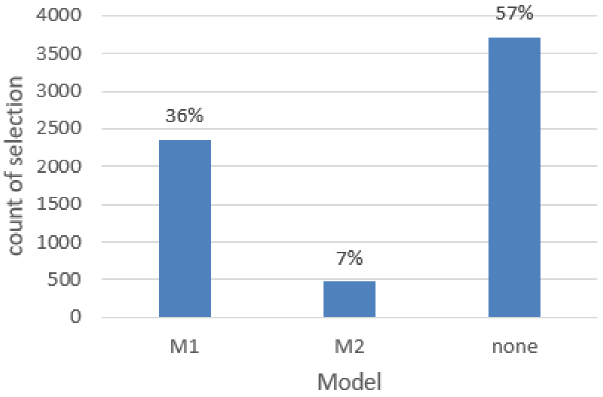


FIGURE 4. Number of selections of the recommended answer from individual models by employees.

It is clear from the results that in up to 57% of the answers, the employees did not choose any of the answers recommended by the models. This "none" option was selected every time employees did not choose the same answer provided by any model, but they modified the response even when the changes were minimal.

Employees were also asked to rate the comprehensibility and adequacy of each recommendation. In the case of comprehension, the evaluator assesses how understandable the given text is and to what extent it uses appropriate words. Several factors affect comprehension, such as grammatical errors and missing words. This evaluation uses a scale where high marks mean good comprehension (in our case, a value of 5) and low marks mean little to no comprehension (in our case, a value of 0). The evaluator assigns a high grade if the answer is entirely understandable. It is grammatically correct and readable. On the other hand, the evaluator assigns a low mark if the answer is incomprehensible, it is difficult to determine the sentence's meaning, and it contains many grammatical errors.

In the case of adequacy, they assessed how relevant the information contained in the answer was, taking into consideration the context of the question and whether the answer was helpful in solving the problem. Unfortunately, due to time constraints, the employees only evaluated the comprehensibility and adequacy of the selected answer. For example, if they chose the answer offered by the M1 model, they evaluated the comprehensibility and adequacy of this answer, while the answer generated by the M2 model was not evaluated. In our graph (fig. 5), we present a box plot evaluating the comprehensibility and adequacy of the selected answers.

It can be seen from the graph that the vast majority of the answers recommended by the M1 model had a comprehension value of 5 (median, upper, and lower quartile are equal to 5). This is obvious because the M1 model did not generate an answer. It just used one of the previous answers created by a human. The results for the M2 model, which also had a high adequacy value (median, upper quartile equal to 5), were surprising. The M2 model has already generated the answers, which means it was able to create comprehensible answers.

In the case of adequacy, even better results were observed for the M2 model. It should also be noted that after choosing the answer recommended by either of the M1 or M2 models, the employees still had the option to modify this answer. For this reason, an answer with adequacy = 1 could be selected in the case of adequacy = 3 (for example, the lower quartile is equal to 3).

When evaluating the graph (fig. 5), it is also necessary to consider the results of the overall answer selection of one of the models. Model M2 was selected only in 7% of all cases. However, when its answers were indeed selected, thesy had high comprehensibility and were even more adequate than in the case of model M1. On the other hand, the M1 model was selected in up to 36% of responses.

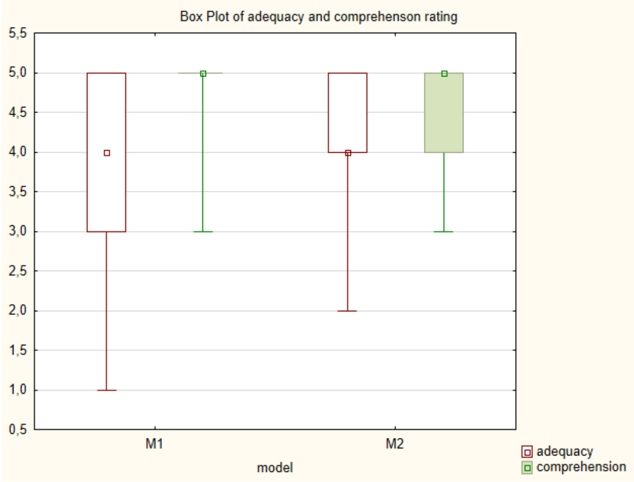


FIGURE 5. Boxplot for adequacy and comprehensibility of selected answers of individual models.

However, we have to combine the results obtained with the help of human evaluators with other approaches. This is mainly because they rated only selected answers.

In addition to employee evaluation, we verified the results using automatic machine translation evaluation metrics. These metrics talk about the differences between the translation generated by the machine (machine translation) and the human translation. From our point of view, we also focus on machine-generated responses and human responses.

Therefore, we decided to use basic machine translation metrics to evaluate our models.

A. BLEU METRIC

The BiLingual Evaluation Understudy (BLEU) metric compares the n-grams of a candidate and one or more reference translations in the evaluation. BLEU, therefore, places great emphasis on the similarity of n-grams between the candidate translation and the reference translations.

It introduces the concept of n-gram accuracy. For example, in the case of a unigram, it is the share of common words in the candidate and reference translation and the total number of words in the candidate translation. BLEU introduces the term modified n-gram accuracy. It is calculated by counting the highest number of occurrences of specific n-grams in the reference translation.

To calculate the n-gram accuracy for bigrams, trigrams, or n-grams, we compare a pair, triple, or tuple of words instead of one word.

BLEU also introduces a penalty for brevity (BP - brevity penalty). This penalty ensures the candidate translation is as long as the reference translation.

Figure (fig. 6) visualizes the results measured using the BLEU metric. We can use this metric to determine the number of words we want the model to match concurrently. For instance, we can opt for words to be matched individually (1-gram), in pairs (2-gram), or triplets (3-grams). The graph displays the metrics for 1 to 4-grams (BLEU-1 – BLEU4).

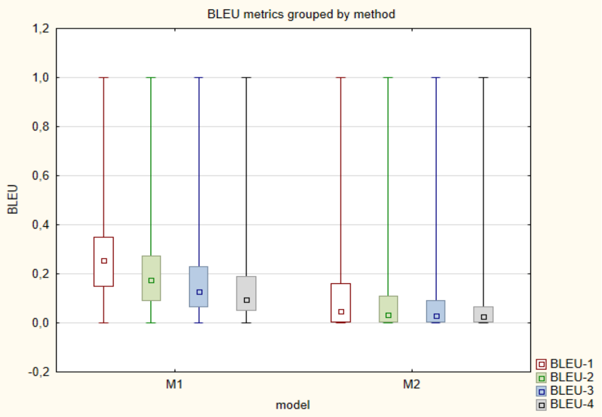


FIGURE 6. Boxplot for BLEU metrics.

The graph provides clear evidence that all the monitored metrics scored 1. This score signifies a perfect alignment between the model's suggested response and the actual response or a complete disparity in the suggested response compared to the real one. The optimal BLEU score is 1, representing a complete match between the proposed and actual responses. It is also observable that superior scores will be noted for 1-gram and 2-gram as opposed to 3-gram or 4-gram. From the visualization of the results, it is clear that better response suggestions were noted for the M1 model.

B. WER METRIC

Word Error Rate (WER) is based on the edit distance (edit operations) and does not allow the reordering of words. The WER is derived from the Levenshtein distance, working at the word level instead of the phoneme level. The WER is a valuable tool for comparing different systems and evaluating improvements within one system.

WER is a significant and widely used metric, not only employed in the evaluation of machine translation but also in gauging the performance of speech recognition APIs, which are instrumental in driving interactive voice-based technology, such as Siri or Amazon Echo.

The results of the WER are visualized in the graph (fig. 7). The worst possible WER score is 1, representing the maximum number of alterations, while the best score is 0, indicating no revisions. The results suggest that a smaller ratio of edits was required for texts recommended by model M1. However, it is evident that given the number of recommendations examined and the length of the texts, the outcomes are not satisfactory for either model.

C. PRECISION, RECALL AND F1-SCORE METRICS

The precision, recall and f1-score metrics are based on the proximity of the hypothesis with the reference, similar to bag-of-words, regardless of the word's position in the sentence. In our case, we consider the recommendation of one of the models (M1, M2) as a hypothesis, and the link is the text that a human sent. In the case of the evaluation of our models, we compared all 6550 responses sent. For each answer,

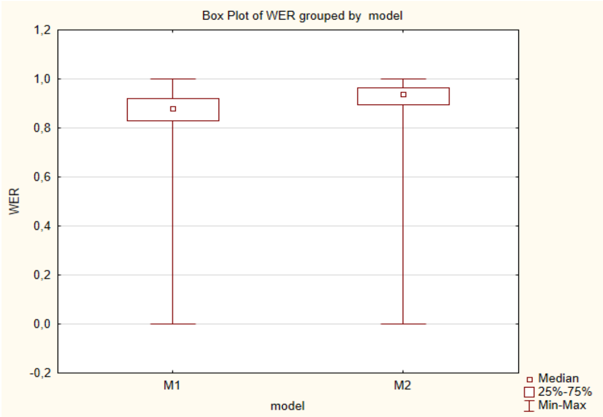


FIGURE 7. Boxplot for WER metrics.

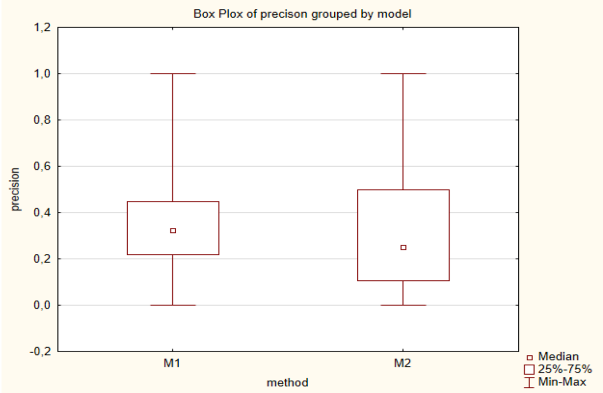


FIGURE 8. Boxplot for the precision metric.

we expressed its precision, recall and f1-score compared to both models.

Precision is a measure of how many correct words are present in hypothesis h given the reference r, i.e. the proportion of words in hypothesis h (recommendation) that are present in link r (real response sent):

$$Precision\ (h|r) = \frac{|h \cap r|}{h}, \tag{5}$$

The results of the precision metric show (fig. 8) that the M1 model (median, upper quartile) achieved better results. A smaller quartile range was also observed in the case of the M1 model. Only the upper quartile values were observed higher in the M2 model. At the same time, the results point to the fact that, in many cases, it was necessary to modify the recommendation. It should be noted that the results include all recommendations of individual models, even those that were not selected.

Recall metrics are calculated as the number of correct words in hypothesis h divided by the number of reference words r, i.e. the proportion of words in link r (real response sent) that are present in hypothesis h (recommendation):

$$recall\ (h|r) = \frac{|h \cap r|}{r}. \tag{6}$$

The results of the recall metric (fig. 9) are very similar to those of precision. All monitored quantities (upper quartile, median, lower quartile) were ranked higher in favour of the M1 model. The quartile range was smaller in the M2 model.

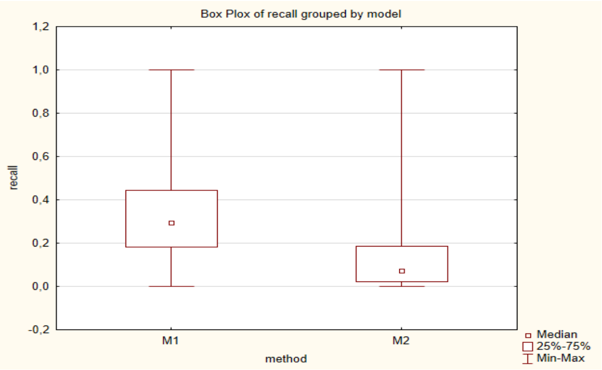


FIGURE 9. Boxplot for the recall metric.

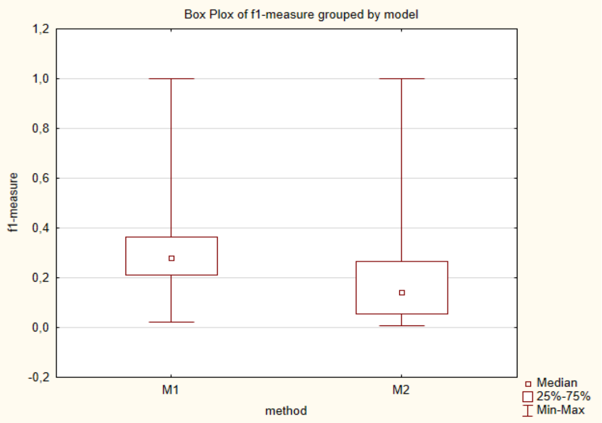


FIGURE 10. Boxplot for the f1-measure metric.

The last metric, in our case, is the f1-measure. The metric represents the harmonic mean of precision and recall:

$$f_1 = \frac{2 * precision * recall}{precision * recall}, \tag{7}$$

The visualized results of the f1-measure metric (fig. 10) confirm the results of the previous metrics, which were higher in favour of the M1 model.

V. CONCLUSION

In our research, we focused on comparing two models used in NLP tasks. This comparison was carried out from the perspective of human evaluations of the models as well as from the perspective of automatic metrics. We consider human evaluation to be a priority because, in our opinion, it most accurately assesses the practical use of the evaluated models. This type of evaluation is a typical qualitative assessment that can capture not only accuracy but also comprehensibility, contextual consideration, and naturalness of the responses. Human evaluation can consider subjective aspects of responses, such as style, tone, and appropriateness for the target audience. On the other hand, issues with this evaluation include being time-intensive and having limited scalability. Humans can evaluate responses within a broader context, capturing subtle nuances that models might find difficult to recognize. For these reasons, in our article, we evaluated not only the results from the perspective of human evaluation but also used automatic evaluation metrics.

Our research set out to dissect the performance and usability of two standard models used in NLP tasks, specifically Generative Pre-trained Transformer (GPT) models and rule-based models. Our interest lies in exploring their effectiveness when deployed for domain-specific tasks within a customer care setting. The findings from our extensive empirical research are intriguing and instructive, paving the way for a deeper understanding of these models and their application.

The rule-based models showcased a noticeable proficiency in handling domain-specific tasks, mainly when they were fed with datasets extracted from the historical communication between users and a specialized domain system, such as a customer care department. This conclusion is well aligned with the inherent structure of rule-based models, designed to efficiently process tasks governed by a clear and unchanging set of rules. Unlike the GPT models, which leverage a broad scope of training data to enhance their overall competency, the rule-based models' performance was primarily dictated by the relevance and specificity of the dataset relative to the task. As a result, with a sufficiently tailored and specific dataset at their disposal, rule-based models can effectively outpace GPT models in performing domain-specific tasks. This result is the most important finding of our article. The rules-based model (M1) achieved better results than the GPT model in terms of the examined metrics. The frequency of response selection (by employees of the CC department) is higher compared to the GPT model. Given the relatively low costs and lesser 'robustness' of the system, this represents a significant benefit for any organization considering the deployment of a dialogue system.

On the other hand, we observed that GPT models exhibited superior performance in comprehensibility and adequacy when they successfully generated the correct answer. The GPT models, trained on enormous amounts of text data, can generate text that closely mirrors natural human language. This is a clear advantage in generating syntactically accurate responses and contextually coherent and semantically meaningful text. This aspect of the GPT models was particularly relevant and advantageous in a customer care setting, where interactions that are meaningful and engaging can greatly enhance user experience.

A rule-based chatbot is trained to generate the most appropriate response from a predefined set of responses. This methodology is particularly well-suited for scenarios characterized by a narrow and clearly defined spectrum of potential user inputs, enabling the chatbot to deliver accurate responses promptly, even without an extensive corpus of training data. Nevertheless, it may encounter difficulties when confronted with more intricate or open-ended interactions in which user input is less foreseeable.

However, the training of GPT models is not without its challenges. These models require substantial datasets and computational resources, which might not always be readily available. It also brings the issue of practicality in deploying these models. In many scenarios, balancing the undeniable benefits of GPT models, such as superior comprehensibility and adequacy, against the efficient, domain-specific performance and less resource-demanding nature of rule-based models could be a more pragmatic approach.

To encapsulate, our research has highlighted the multifaceted nature of chatbot design and development. The efficacy of a chatbot, whether it uses a GPT or rule-based model, cannot be attributed to one single factor. Instead, it is a complex interplay of various aspects, such as the task's nature, the model's characteristics, the richness and relevance of the dataset, and the available resources. Our study underscores the necessity for a nuanced understanding of the strengths and weaknesses of different models. This knowledge can inform a strategic alignment of the model selection with the specific task requirements and resources, ensuring an effective and efficient outcome.

Considering our findings, we believe there is an immense scope for future research. Future studies could explore innovative ways to blend the strengths of GPT and rule-based models, fostering a hybrid model that offers an optimal balance of performance, comprehensibility, adequacy, and resource efficiency. Such an endeavor could potentially lead to a revolutionary approach in chatbot design, harnessing the best of both worlds to serve the ever-evolving demands of users.

ACKNOWLEDGMENT

The authors would like to extend their sincere gratitude to POWERPLAY MANAGER, s.r.o., for graciously allowing them to conduct their research experiment within their organization. Their invaluable cooperation has been instrumental in the successful completion of this study. They would also like to thank all team members involved for their support and facilitation.

REFERENCES

[1] B. Peng, C. Li, P. He, M. Galley, and J. Gao. *Instruction Tuning With GPT-4*. Accessed: Mar. 7, 2024. [Online]. Available: https://instruction-tuning-with-gpt-4.github.io/

[2] T. Hu, A. Xu, Z. Liu, Q. You, Y. Guo, V. Sinha, J. Luo, and R. Akkiraju, "Touch your heart: A tone-aware chatbot for customer care on social media," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2018, pp. 1–12.

[3] A. A. Georgescu, "Chatbots for education—Trends, benefits and challenges," in *Proc. 14th Int. Conf. eLearning Softw. Educ.*, Apr. 2018, vol. 14, no. 2, pp. 195–200.

[4] J. Chu, "Recipe bot: The application of conversational AI in home cooking assistant," in *Proc. 2nd Int. Conf. Big Data Artif. Intell. Softw. Eng. (ICBASE)*, Sep. 2021, pp. 696–700.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 5999–6009.

[6] *Microsoft Invests in and Partners With OpenAI To Support Us Building Beneficial AGI*. Accessed: Mar. 7, 2024. [Online]. Available: https://openai.com/blog/microsoft-invests-in-and-partners-with-openai

[7] *Microsoft Teams Up With OpenAI To Exclusively License GPT-3 Language Model—The Official Microsoft Blog*. Accessed: Mar. 7, 2024. [Online]. Available: https://blogs.microsoft.com/blog/2020/09/22/microsoft-teams-up-with-openai-to-exclusively-license-gpt-3-language-model/

[8] OpenAI. *GPT-4 Technical Report*. Accessed: Sep. 15, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774v3#

[9] *OpenAI's Massive GPT-3 Model is Impressive, but Size Isn't Everything | VentureBeat*. Accessed: Mar. 7, 2024. [Online]. Available: https://venturebeat.com/ai/ai-machine-learning-openai-gpt-3-size-isnt-everything/

[10] *A Robot Wrote This Entire Article. Are You Scared Yet, Human? | GPT-3 | The Guardian*. Accessed: Mar. 7, 2024. [Online]. Available: https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3

[11] *The Guardian's GPT-3-Written Article Misleads Readers About AI. Here's Why—TechTalks*. Accessed: Mar. 7, 2024. [Online]. Available: https://bdtechtalks.com/2020/09/14/guardian-gpt-3-article-ai-fake-news/

[12] *Models—OpenAI API*. Accessed: Mar. 7, 2024. [Online]. Available: https://platform.openai.com/docs/models/overview

[13] *AI21 Labs Makes Language AI Applications Accessible To Broader Audience | Bus. Wire*. Accessed: Mar. 7, 2024. [Online]. Available: https://www.businesswire.com/news/home/20210811005033/en/AI21-Labs-Makes-Language-AI-Applications-Accessible-to-Broader-Audience

[14] *Language Models Are Unsupervised Multitask Learners*. Accessed: Sep. 15, 2023. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[15] *Jurassic-1: Technical Details and Evaluation*. Accessed: Sep. 15, 2023. [Online]. Available: https://cdn.prod.website-files.com/60fd4503684b466578c0d307/61138924626a6981ee09caf6_jurassic_tech_paper.pdf

[16] *Introducing J1-Grande!* Accessed: Mar. 7, 2024. [Online]. Available: https://www.ai21.com/blog/introducing-j1-grande

[17] L. Reed, C. Li, A. Ramirez, L. Wu, and M. Walker, "Jurassic is (almost) all you need: Few-shot meaning-to-text generation for open-domain dialogue," in *Proc. Int. Conf. Conversational AI Natural Hum.-Centric Interact.*, in Lecture Notes in Electrical Engineering, vol. 943, 2021, pp. 99–119.

[18] A. Przegalinska and D. Jemielniak, *Strategizing AI in Business and Education*. Cambridge, U.K.: Cambridge Univ. Press, Apr. 2023.

[19] *The Bizarre and Terrifying Case of the 'Deepfake' Video that Helped Bring an African Nation to the Brink*. Accessed: Sep. 15, 2023. [Online]. Available: https://www.motherjones.com/politics/2019/03/deepfake-gabon-ali-bongo/

[20] J. He, J. Qiu, A. Zeng, Z. Yang, J. Zhai, and J. Tang, "FastMoE: A fast mixture-of-expert training system," 2021, *arXiv:2103.13262*.

[21] *AI Weekly: China's Massive Multimodal Model Highlights AI Research Gap | VentureBeat*. Accessed: Mar. 7, 2024. [Online]. Available: https://venturebeat.com/business/ai-weekly-chinas-massive-multimodal-model-highlights-ai-research-gap/

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Oct. 2018, pp. 4171–4186.

[23] *Introducing Google Research Football: A Novel Reinforcement Learning Environment*. Accessed: Sep. 15, 2023. [Online]. Available: https://research.google/blog/introducing-google-research-football-a-novel-reinforcement-learning-environment/

[24] *Google Bard is Now Gemini: How to Try Ultra 1.0 and New Mobile App*. Accessed: Jul. 9, 2024. [Online]. Available: https://blog.google/products/gemini/bard-gemini-advanced-app/

[25] S. A. Thorat and V. Jadhav, "A review on implementation issues of rule-based chatbot systems," in *Proc. Int. Conf. Innov. Comput. Commun. (ICICC)*, 2020, doi: 10.2139/ssrn.3567047.

[26] R. Agarwal and M. Wadhwa, "Review of state-of-the-art design techniques for chatbots," *Social Netw. Comput. Sci.*, vol. 1, no. 5, pp. 1–12, Sep. 2020.

[27] N. V. Shinde, A. Akhade, P. Bagad, H. Bhavsar, S. K. Wagh, and A. Kamble, "Healthcare chatbot system using artificial intelligence," in *Proc. 5th Int. Conf. Trends Electron. Informat. (ICOEI)*, Jun. 2021, pp. 1–8.

[28] A. A. Abd-alrazaq, M. Alajlani, A. A. Alalwan, B. M. Bewick, P. Gardner, and M. Househ, "An overview of the features of chatbots in mental health: A scoping review," *Int. J. Med. Informat.*, vol. 132, Dec. 2019, Art. no. 103978.

[29] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. S. Lau, and E. Coiera, "Conversational agents in healthcare: A systematic review," *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 9, pp. 1248–1258, Sep. 2018.

[30] K. Moore, S. Zhong, Z. He, T. Rudolf, N. Fisher, B. Victor, and N. Jindal, "A comprehensive solution to retrieval-based chatbot construction," *Comput. Speech Lang.*, vol. 83, Jan. 2024, Art. no. 101522.

[31] H. Akkineni, P. V. S. Lakshmi, and L. Sarada, "Design and development of retrieval-based chatbot using sentence similarity," in *Proc. Int. Conf. IoT Anal. Sensor Networks*, in Lecture Notes in Networks and Systems, vol. 244, 2022, pp. 477–487.

[32] Z. Chen, Y. Lu, M. P. Nieminen, and A. Lucero, "Creating a chatbot for and with migrants: Chatbot personality drives co-design activities," in *Proc. ACM Designing Interact. Syst. Conf.*, Jul. 2020, pp. 219–230.

[33] B. Thomson, *Statistical Methods for Spoken Dialogue Management*. London, U.K.: Springer, 2013, doi: 10.1007/978-1-4471-4923-1.

[34] R. Dsouza, S. Sahu, R. Patil, and D. R. Kalbande, "Chat with bots intelligently: A critical review & analysis," in *Proc. Int. Conf. Adv. Comput., Commun. Control (ICAC3)*, Dec. 2019, pp. 1–6.

[35] D. P. Panagoulias, M. Virvou, and G. A. Tsihrintzis, "Augmenting large language models with rules for enhanced domain-specific interactions: The case of medical diagnosis," *Electronics*, vol. 13, no. 2, p. 320, Jan. 2024.

[36] D. P. Panagoulias, F. A. Palamidas, M. Virvou, and G. A. Tsihrintzis, "Rule-augmented artificial intelligence-empowered systems for medical diagnosis using large language models," in *Proc. IEEE 35th Int. Conf. Tools with Artif. Intell. (ICTAI)*, Nov. 2023, pp. 70–77.

[37] J. S. Cuadrado, S. Pérez-Soler, E. Guerra, and J. de Lara, "Automating the development of task-oriented LLM-based chatbots," in *Proc. 6th ACM Conf. Conversational User Interfaces (CUI)*, 2024, pp. 1–10.

[38] *Using LLMs With Rasa*. Accessed: Jul. 9, 2024. [Online]. Available: https://rasa.com/docs/rasa/next/llms/large-language-models/

[39] M. Y. Helmi Setyawan, R. M. Awangga, and S. R. Efendi, "Comparison of multinomial naive Bayes algorithm and logistic regression for intent classification in chatbot," in *Proc. Int. Conf. Appl. Eng. (ICAE)*, Oct. 2018, pp. 1–5.

[40] X. Liu, A. Eshghi, P. Swietojanski, and V. Rieser, "Benchmarking natural language understanding services for building conversational agents," in *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction* (Lecture Notes in Electrical Engineering), vol. 714. Singapore: Springer, 2019, pp. 165–183.

[41] D. Braun, A. Hernandez-Mendez, F. Matthes, and M. Langen, "Evaluating natural language understanding services for conversational question answering systems," in *Proc. 18th Annu. SIGdial Meeting Discourse Dialogue*, 2017, pp. 174–185.

[42] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?" in *Proc. Deep Learn. Inside Out (DeeLIO), 3rd Workshop Knowl. Extraction Integr. Deep Learn. Architectures*, 2022, pp. 100–114.

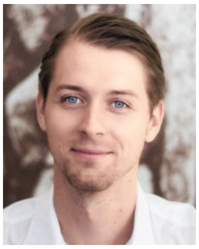
[43] W. Maroengsit, T. Piyakulpinyo, K. Phonyiam, S. Pongnumkul, P. Chaovalit, and T. Theeramunkong, "A survey on evaluation methods for chatbots," in *Proc. 7th Int. Conf. Inf. Educ. Technol.*, Mar. 2019, pp. 111–119.

[44] S. Pandey and S. Sharma, "A comparative study of retrieval-based and generative-based chatbots using deep learning and machine learning," *Healthcare Anal.*, vol. 3, Nov. 2023, Art. no. 100198.

[45] P. Qin, W. Xu, and J. Guo, "A novel negative sampling based on TFIDF for learning word representation," *Neurocomputing*, vol. 177, pp. 257–265, Feb. 2016.

[46] C.-H. Chen, "Improved TFIDF in big news retrieval: An empirical study," *Pattern Recognit. Lett.*, vol. 93, pp. 113–122, Jul. 2017.

[47] *J2 Complete API*. Accessed: Mar. 7, 2024. [Online]. Available: https://docs.ai21.com/reference/j2-complete-api-ref



DOMINIK HALVONÍK is currently an Assistant Professor with the Department of Informatics, Faculty of Natural Sciences, Constantine the Philosopher University in Nitra. His research interests include the investigation of user behavior in games, building chatbots, and natural language processing.

JOZEF KAPUSTA is currently an Associate Professor with the Department of Informatics, Faculty of Natural Sciences, Constantine the Philosopher University in Nitra. His research interests include natural language processing, machine learning, and artificial intelligence.