Text Message Routing System for Chat-Based Applications

Breno U. de Angelo[1,2], Guilherme G. Zanetti[1,2], Alberto F. De Souza[1,2], Claudine Badue[1], Abner G. Jacobsen[2], Thiago Oliveira-Santos[1]

[1]Department of Informatics – Universidade Federal do Espírito Santo (UFES)
Vitória – Espírito Santo – Brazil

[2]Research Department – Aumo S.A.
Vitória – Espírito Santo – Brazil

{breno.angelo,guilherme.zanetti,alberto,claudine}@lcad.inf.ufes.br

abner@aumo.ai, todsantos@inf.ufes.br

Abstract. Research Context*: The integration of Large Language Models (LLMs) into banking customer service promises enhanced engagement but faces a trilemma of prohibitive operational costs, stochastic inaccuracy (hallucinations), and strict safety requirements. In the Brazilian financial sector, where inaccuracies can lead to direct financial harm, relying solely on monolithic LLMs is economically unsustainable and operationally risky.* ***Practical Problem****: Financial institutions struggle with the "inference cost trap," where token-based pricing scales linearly with usage, and the risk of hallucinations—where models generate plausible but incorrect responses for unfamiliar queries.* ***Proposed Solution****: This study validates a semantic routing architecture that functions as an intelligent decision layer, classifying user intent into four categories (Relevant, Unrelated, Chitchat, Spam) to route queries to cost-effective handlers. Five classification paradigms were benchmarked: Zero-shot GPT, Few-shot GPT, QLoRA fine-tuned GPT, Embedding Similarity Search, and BERT-based neural models.* ***Related IS Theory****: The research builds upon Task-Technology Fit (TTF) theory and contributes to the Green IS agenda by validating energy-efficient architectures.* ***Research Method****: A balanced dataset of 6,760 messages, curated from SMS spam data, SQuAD questions, private chats, and banking FAQs, was used to evaluate accuracy, resource usage, and cost.* ***Summary of Results****: Embedding Similarity (98.52%) and BERT-based models (98.96%) achieved superior accuracy compared to Zero-shot GPT (53.78%) and matched the performance of Fine-tuned GPT (97.93%). Crucially, the embedding approach reduced operational costs by 96% (from US$18.10 to US$0.32 per million requests) and slashed memory consumption from 6.52 GB to 1.04 GB.* ***Contributions and Impact to IS Area****: The study contributes a validated pattern for Frugal AI in Portuguese, demonstrating that open-source embedding models can effectively govern LLMs. This approach mitigates hallucination risks, reduces dependency on foreign APIs, and aligns with sustainable computing principles.*

1. Research Context

Chat-based applications have increasingly evolved into indispensable tools for customer service, virtual assistance, and automated support systems. Their growing importance is directly tied to recent advancements in Natural Language Processing (NLP), particularly

with the emergence and rapid progress of LLMs. These models have enabled conversational agents to provide natural, context-aware, and coherent interactions with users, offering a level of engagement that was previously unattainable with rule-based or shallow machine learning approaches.

The adoption of chat-based systems spans multiple domains, including healthcare, retail, finance, and education, where they serve as the first line of interaction between organizations and their clients. In these contexts, the ability of conversational systems to deliver quick and precise answers contributes to increased customer satisfaction, reduction of response times, and significant operational savings. Within the financial sector, for example, chat-based applications can assist customers with everyday queries such as balance verification, transaction details, and product information, reducing the need for direct human intervention and enabling banks to scale their services more efficiently.

At the same time, the rise of generative AI has reshaped expectations about the capabilities of conversational systems. Modern users anticipate that virtual assistants will not only provide predefined answers but also handle diverse requests in a flexible and adaptive manner. This expectation makes the role of LLM-powered assistants central in customer-facing applications. Consequently, research interest in enhancing the robustness, scalability, and efficiency of such systems has intensified, with particular emphasis on ensuring reliable classification of user queries and context-sensitive responses.

In this study, we focus specifically on the banking domain, where interactions often involve sensitive information and require careful handling. The context under analysis is a chat interface in which a customer interacts with an LLM-powered assistant. The system must be capable of recognizing and appropriately responding to relevant questions based on available documentation, while also being able to gracefully manage queries that fall outside its scope. Additionally, even when users provide off-topic input, the assistant is expected to maintain conversational flow, preserving engagement and user trust. This dual requirement—providing accurate answers when possible and maintaining natural dialogue otherwise—highlights the necessity of structured and well-designed approaches to semantic routing in chat-based applications.

In the Brazilian context, the rapid digitalization of banking—accelerated by systems like Pix—has led to an explosion in customer service interactions. However, the reliance on massive foreign LLMs for every interaction creates a dependency on fluctuating exchange rates and foreign APIs, challenging the digital sovereignty of local institutions. Furthermore, the Information Systems community (SBSI) has increasingly called for "Green Information Systems" that prioritize energy efficiency. The high energy consumption associated with unnecessary LLM inference contradicts these goals, necessitating architectural patterns that prioritize "Frugal AI".

From a theoretical perspective, using a massive reasoning engine for simple classification tasks represents a misalignment of Task-Technology Fit. This work proposes resolving this by interposing a lightweight, embedding-based decision layer. This approach resolves the tension between advanced reasoning and the constraints of cost, latency, and safety.

2. Practical Problem

The use of LLMs for customer support in chat applications still faces two major challenges: the general problem of hallucinations inherent to LLMs and the high operational cost.

Hallucinations in generative language models are still an open problem [Ji et al. 2023], and occurs when models generate responses that seem to be accurate to a non-expert but contain erroneous or misleading information. This issue is particularly prevalent when the model encounters unfamiliar demands. Recent studies have presented possible explanations for this problem [Lindsey et al. 2025] and [Kalai et al. 2025], but there are still no definite solutions.

LLMs inferences are expensive, requiring significant computational resources. Progress has been made in this field both to increase throughput [He and Zhai 2024, Agrawal et al. 2024, Kwon et al. 2023] and reduce memory usage via quantization techniques for example [Lin et al. 2023, Frantar et al. 2023], but the use of powerful LLMs still incurs in high costs, both when employing self-hosting or when using APIs for LLM inference service providers. When self-hosting, the server capacity must scale with the model's usage and, in the case of employing a LLM inference service provider, most will charge based on the number of input and output tokens, making frequent API calls costly.

In the case at stake — a banking customer support application — hallucinations may result in wrong directions for the customer causing financial loss. Furthermore, the high scale of such applications lead to high operational cost as described.

3. Proposed Solution

To mitigate these issues, we propose the implementation of a semantic routing system. Instead of directly forwarding user's text messages to a powerful LLM-based chatbot system, the text is first classified into specific categories. Only messages closely aligned with the application's commercial intent are sent to the chatbot. For other messages, our semantic routing system allows for more cost-effective solutions to respond to users.

The semantic routing approach for chat-based systems has been presented in recent literature [Manias et al. 2024a] and is being explored by the community [Horsey 2024, Azharudeen 2024] for uses similar to the one presented in this paper, with highlight to the semantic-router[1], an open-source Python library developed to facilitate the implementation of this technique. We examined it in our study in the classifier based on vector similarity search.

In this work, we compare five different text classification methods in a real-world application of semantic routing for an end-user chat system. We implement and test the performance of zero-shot and few-shot prompting with Generative Pre-trained Transformers (GPTs), QLoRA GPT fine-tuning [Dettmers et al. 2023], text embedding vector similarity search, and training a classification head on top of a Bidirectional Encoder Representations from Transformers (BERT) based encoder model [Devlin et al. 2019] in this specific application.

[1] https://github.com/aurelio-labs/semantic-router/

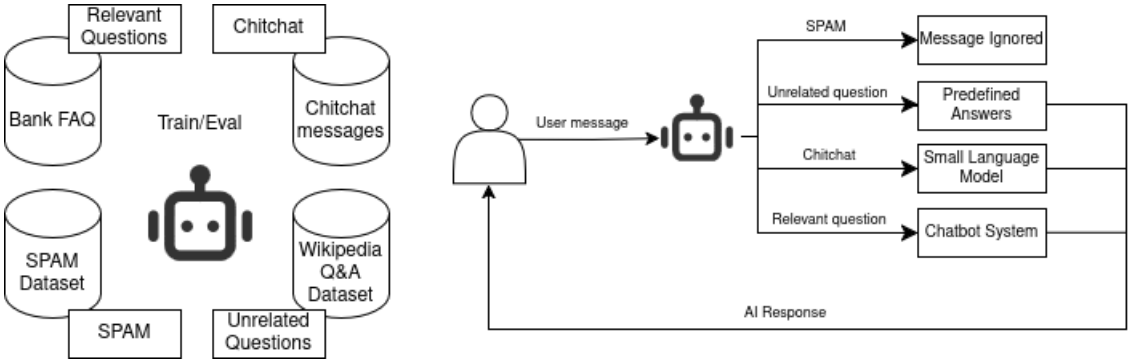

Figure 1. Flow chart of chat-app with message routing system.

3.1. Routing

In the proposed routing system the user input can be considered spam, unrelated question, chitchat or relevant question, as shown in Fig. 1. For each case, the message is forwarded to be answered with the appropriate tools.

Relevant questions are the core interactions that the chatbot is designed to handle. These inquiries require precise responses and are routed to the main chatbot system. While advanced retrieval and prompting techniques can enhance the chatbot's accuracy, this aspect is beyond the scope of our study. Our primary focus is ensuring that only pertinent queries are directed to the chatbot, thereby maintaining system efficiency and reliability.

Unrelated questions are those that do not pertain to the chatbot's primary domain. These must be managed to avoid misallocation of computational resources. We propose using a pool of generic, instructive responses that guide users towards appropriate usage of the chatbot, thus preserving engagement without straying into untested areas that could compromise system robustness.

Chitchat encompasses casual, conversational messages. Although these need to be addressed to maintain user engagement, they do not require detailed responses. Smaller generative models like Mistral-7b or Llama3-7b can be utilized to generate varied and engaging replies, tailored to maintain the natural flow of conversation.

Spam messages are defined as unsolicited messages intended for advertising or fraudulent activities. Such messages must be ignored to prevent unnecessary computational costs and potential security risks.

The classification problem at study is approached with different techniques and is deeply experimented and analyzed in the following subsections.

3.1.1. Generative Pre-trained Transformers (GPTs)

Generative language models can be repurposed as classifiers by prompting them to provide short, categorical responses. In our case, the model is questioned to classify the text given by outputting one of the following terms: `RELEVANT_QUESTION`, `UNRELATED_QUESTION`, `CHITCHAT` or `SPAM`.

The answer given by the model is stripped to remove line-feeds and blank spaces at the beginning and at the end of the response. If the resulting answer is one of the specified terms, the message is routed to the appropriate tool. Otherwise, the message is considered an invalid response and the classification has failed. In this case, a default route must be picked according to the application for handling the exception.

This study quantitatively analyzed how few-shot and fine-tuning can improve the result achieved by zero-shot classification.

3.1.2. Zero-Shot

The model, prompted in Brazilian Portuguese, classifies messages without prior examples. The translated prompt directs the model to categorize the text into one of the predefined classes.

Since the relevant questions were obtained from a bank's website, the model is told to be an assistant of the bank. In this way, it is able to rationally determine whether a question is relevant for the use case. The system prompt used is shown bellow.

```
You are a bank's Virtual Assistant.

You will be given a text, and your goal is to classify
it as one of the following classes:

SPAM: Messages offering services or fraudulent
messages with the purpose of tricking the readers;

UNRELATED_QUESTION: Questions not related to banking
activities;

RELEVANT_QUESTION: Questions related to banking
activities;

CHITCHAT: Casual chitchat conversation.

Answer only by informing the class, with no extra
word.
```

3.1.3. Few-Shot

In few-shot classification, examples are added to the chat-history to increase accuracy.

In our tests, one example of each class was picked from the training dataset and included as translated below:

```
SYSTEM PROMPT: Zero-shot system prompt

USER: URGENT! You've won a FREE 1-week membership
to our £100,000 Jackpot Prize!  Send word:  CLAIM to No:
[...]

ASSISTANT: SPAM

USER: What replaced lower-skilled workers in the
```

```
United States?
     ASSISTANT: UNRELATED_QUESTION
     USER: How do I unblock the card due to incorrect
password entry?
     ASSISTANT: RELEVANT_QUESTION
     USER: Hello Hello
     ASSISTANT: CHITCHAT
```

3.1.4. Fine-Tuned

A Language Model can be fine-tuned to acquire certain behaviors and increase accuracy in specific tasks. Therefore, a Language Model is fine-tuned with QLoRA, using the Zero-shot system prompt both during training and evaluation.

3.1.5. Embedding Similarity Search

This approach is setup by using an embedding model to generate vector representations of the messages from the training dataset and storing them in a vector database. During inference, the embedding model is used to generate a vector representation of the input. The label output is the label of the nearest neighbor of the input.

The choice of vector database can vary depending on the needs. For a testing environment with small number of samples, a local, in RAM, database can be enough. They load all the vectors in RAM and compute the cosine similarity [Mussmann and Ermon 2016] between the query string and all the stored embedding vectors during search, returning the top_k closest ones, with top_k usually being a small integer. Another more sophisticated approach is to use a dedicated vector database, this allows exhaustive search for kNN [Cunningham and Delany 2021] exact solutions and HNSW (Hierarchical Navigable Small World Graph) [Malkov and Yashunin 2018] for approximate solutions that can be computed fastly and, therefore, used in production. Dedicated vector database can also usually work with permanent storage for large number of vectors.

3.1.6. Bidirectional Encoder Representations from Transformers based neural models

Similar to the method above, the BERT based neural models method uses an embedding model to generate the vector representation of the input. A classification head, composed of a single linear layer is appended to the model and the model is trained using the crafted dataset to output the logits of each class.

4. Related Works

This section reviews the evolution of intent detection, the emergence of semantic routing as a cost-optimization strategy, and the theoretical implications for safety and sustainability in Information Systems.

4.1. Evolution of Intent Detection: From Keywords to Embeddings

Historically, banking chatbots relied on pattern matching, which was computationally inexpensive but brittle. The advent of transformer-based architectures revolutionized this landscape. [Casanueva et al. 2020] established benchmarks for fine-grained intent classification in finance, showing that specialized NLU models could achieve high accuracy. In the Portuguese context, [Souza et al. 2020] introduced BERTimbau, demonstrating that encoder-only models fine-tuned for Portuguese significantly outperform multilingual baselines. However, the emergence of Generative AI led to attempts at using LLMs for zero-shot classification. While versatile, recent studies indicate that LLMs often lag behind specialized BERT-based models in specific classification tasks when latency and cost are factored in.

4.2. Semantic Routing and Frugal AI Architectures

To address the high cost of LLM inference, the concept of Frugal AI has gained prominence. [Chen et al. 2023] proposed the ”LLM Cascade” pattern, where queries are processed by sequentially larger models, reducing costs by up to 98%. Similarly, [Ong et al. 2024] introduced preference-based routing to dynamically select between ”strong” and ”weak” models.

Our work aligns with the semantic routing approach, which utilizes vector similarity search to make routing decisions in under 100 ms. Unlike probabilistic LLM routers, embedding-based routing offers deterministic behavior, which is critical for compliance in banking environments.

4.3. Hallucination Mitigation via Structural Guardrails

Hallucinations in LLMs are often described as structural artifacts of probabilistic next-token prediction. In customer service, distinguishing between factual errors and faithfulness errors is critical. Semantic routing functions as a Guardrail. By strictly classifying inputs as Unrelated or Spam before they reach the generative component, the system enforces a deterministic refusal, effectively removing the opportunity for the LLM to hallucinate an answer for out-of-scope queries.

4.4. The Portuguese NLP Landscape

Most semantic routing benchmarks remain English-centric. [Pires et al. 2023] highlighted that while frontier models (e.g., GPT-4) perform well in Portuguese, smaller open-source models often suffer performance degradation due to the ”curse of multilinguality.” [Gonçalves et al. 2025] highlighted that banking chatbots often fail to understand Portuguese as a second language for deaf users. This reinforces the need for specialized embedding models (like the multilingual-e5 used in this study) rather than relying on the generalized capabilities of a multilingual LLM.

4.5. Green IS and Sustainability

Finally, this work contributes to the Green IS discourse. [Verdecchia et al. 2023] emphasize that inference efficiency is often more environmentally significant than training cost for widely deployed models. By demonstrating that embedding-based routing consumes significantly less VRAM (1GB) compared to generative routing (7GB), this architecture validates that Frugal AI can effectively reduce the carbon footprint of enterprise systems.

5. Related IS Theory

The solution proposed relies on identifying the user intent for correct semantic routing. The use of intent classifiers for chatbots has been explored. [Manik et al. 2021] implements one to identify whether or not the user query is in the assistant's scope.

In more recent approaches, [Manias et al. 2024b] uses GPT-3.5 with few-shot prompting for classifying user intent. Based on the intent, the prompt of an LLM assistant is adapted to enhance the response to the user. In further research, [Manias et al. 2024a] uses the semantic-router implementation of *Aurelio AI* with OpenAI's encoder for the intent classification and compares it against the results in their previous research. The semantic-router achieved higher performance than the few-shot classification even when all invalid answers from the GPT were not considered by the accuracy metric.

Similarly, this work aims to detect the user intent as one of four categories. To do so, we explore different text-classification techniques. In this field, various papers have presented important comparisons.

[Gasparetto et al. 2022] compares the reported performance of shallow and deep learning models for text classification across relevant datasets. The paper shows results using Naive Bayes, Linear SVM, LSTM, CNN, BERT and GPT.

With the advances in GPTs, text-classification via prompt became possible. [Wang et al. 2023] compares the performance of RNN, LSTM, GRU, BERT and zero-shot classification with Llama2, GPT-3.5 and GPT-4. The results are presented for sentiment analysis in Covid-19 tweets and economics texts, classification of e-commerce texts, and spam detection on an SMS dataset. In the economics texts, GPT-4 outperformed all models, but in the others, RNN and GRU presented the best results.

In both researches, BERT has presented consistent results across different datasets. Our research compares it against newer classification methods using generative models and the semantic-router framework powered by open-source models. To determine the viability of using generative models for text-classification we use zero-shot, few-shot prompting and fine-tune a model for the specific task and dataset. The results are then compared considering token usage, memory usage, scalability and development time.

Our work also presents an architecture for avoiding unnecessary load in the LLM that powers a chatbot, which can be applied for other scenarios.

6. Research Methods

6.1. Dataset

The dataset for training and evaluation consists of messages curated to represent each classification category. Below, we describe each message category, the source of our dataset samples for the category, and provide one example message from each category translated from Portuguese to English (since the target chat-based human assistance application is intended for Portuguese-speaking users):

- **Spam messages:** Sourced from an open-source SMS spam dataset[2] and translated into Brazilian Portuguese using GPT-4o, ensuring linguistic relevance. Two extra

[2] https://www.kaggle.com/datasets/vishakhdapat/sms-spam-detection-dataset/data

messages were artificially created from each original one, by prompting GPT-4o to rewrite the original passage with a different subject while maintaining the same structure.
    - **Example:** `Please CALL 08712404000 immediately as there is an urgent message waiting for you.`
- **Unrelated questions:** Extracted from the translated version of the SQuAD dataset[3], covering a broad spectrum of general knowledge queries.
    - **Example:** `What plateau is the left part of Warsaw on?`
- **Chitchat messages:** Gathered from a private chat application, providing realistic conversational data.
    - **Example:** `Good morning.`
- **Relevant questions:** Collected from a company's official FAQ website[4], reflecting real-world user inquiries. To simulate actual usage scenarios, half of the relevant questions were concatenated with chitchat messages.
    - **Example:** `How does` *`my bank`* `credit card bill installments work?`

The dataset is balanced with 1690 messages per category and is split into training (4259 messages), validation (1825 messages), and testing (676 messages) subsets, ensuring an equal representation of each label across these splits. This balanced dataset forms the foundation for the subsequent classification and routing tasks.

By leveraging this comprehensive and balanced dataset, our proposed system can be rigorously evaluated, ensuring its applicability and reliability in real-world chat-based applications.

Because the dataset contains proprietary customer-support interactions and is subject to confidentiality agreements, it cannot be publicly released. The implementation code is also not publicly shareable due to company policy. However, all experimental settings, training parameters, and evaluation methods are documented in detail to enable independent replication on similar datasets.

6.2. Experiments

The methods that required training were trained using the train split of the dataset and all evaluations were made with the test split. The performance of the methods were compared using the accuracy metric, defined as the ratio of the number of correct predictions to the total number of predictions.

For greater insights, confusion matrices are presented for each method.

6.3. Zero-Shot and Few-Shot

For zero-shot and few-shot classification, it was used Mistral-7B-Instruct-v0.3 as classifier with temperature set to 0.0. Mistral-7B was the model of choice due to its reduced size, a financially important aspect, and due to its ability at handling Brazilian Portuguese text compared to models like Llama3-7B and phi-3-small.

[3] `https://github.com/nunorc/squad-v1.1-pt`

[4] `https://banco.bradesco/html/pessoajuridica/atendimento/fale-conosco/perguntas-frequentes.shtm`

The experiment was run on a system equipped with an NVIDIA A100 GPU, utilizing vLLM [Kwon et al. 2023], for batch processing and enhanced throughput.

6.4. Fine-Tuned

Mistral-7B-Instruct-v0.3 was chosen to be fine-tuned as classifier for the four labels, for the same reasons explained before. The training used the same system prompt as zero-shot and was trained with input being the text from the train dataset and the expected output being the correct label.

Query, Key, Value, and Output projection matrices of the self-attention layers were augmented with LoRA adapters, which were trained using QLoRA for one epoch with batch size of 4. The LoRA matrices rank chosen was $r = 128$, with scaling factor $\alpha = 128$, which adds the fine-tune weights to the matrices without scaling. The dropout used was 0.1, and the learning rate was $1e - 5$ with cosine scheduling with warm up ratio of 0.1. These values were chosen arbitrarily as they are common in literature and are proven to work well. It was used the same machine as the zero-shot and few-shot experiment, mainly utilizing the PEFT [Han et al. 2024] and HuggingFace Framework [Wolf et al. 2020].

During inference, the model was set to a temperature of 0.0.

6.5. Embedding Similarity Search

For routing with vector similarity multilingual-e5-large-instruct (560M) was used due to better performance with Brazilian Portuguese. This model outputs a vector of dimension 1024 as representation of the input text. The vectors generated with the training dataset are saved to a vector database along with the expected label. It was chosen to use a local vector database for simplicity, as there is no need to create two separate processes. We choose the number of returned embeddings top_k to be 5.

The experiment was run on an Intel® Core™ i3 12th generation with 16GB of RAM.

6.6. BERT based neural model

It was chosen to use the same embedding model as the method above (multilingual-e5-large-instruct), for comparison reasons.

The classification head appended to the pre-trained model is a feed forward neural network with one hidden layer of size 1024 and final layer of size 4. The resulting model was trained with the training dataset and uses Cross-Entropy Loss, batch size of 64, 5 epochs, learning rate of $2e - 5$ and dropout of 0.1 in the encoder output. The experiment was run with the same machine as the generative classification.

7. Ethical Considerations and Risks

While the semantic routing system proposed in this study demonstrates significant operational efficiency and cost reduction, its deployment within the sensitive banking domain introduces specific ethical risks. The reliance on automated classification and the storage of user intent requires a rigorous analysis of algorithmic bias, model safety, and data privacy to ensure the system does not compromise user trust or financial security.

7.1. Algorithmic Bias and Financial Exclusion

The classification of user messages into exclusionary categories such as "Spam" or "Unrelated" relies heavily on the underlying model's ability to interpret diverse linguistic patterns. There is a significant risk that models trained on standard datasets may underperform when processing non-standard dialects, regional slang, or text from users with lower digital literacy. In the banking sector, this technical limitation can manifest as *financial exclusion* [Women's World Banking 2021, Fuster et al. 2022]. For instance, if a fraud report sent by a customer with poor grammar is misclassified as "Spam" and subsequently ignored by the routing layer, that customer is effectively denied critical service. To mitigate this, high-stakes categories such as "Spam" must operate with a human-in-the-loop review process for low-confidence predictions, rather than automated deletion.

7.2. Safety and Hallucination Risks in Smaller Models

Our architecture proposes routing "Chitchat" messages to smaller, cost-effective Large Language Models (LLMs) to preserve the computational budget for complex queries. However, recent empirical studies indicate a strong inverse correlation between model parameter count and hallucination rates reported by previous researches [Wei et al. 2024, van der Heijden et al. 2025]. Smaller models, while efficient, are statistically more prone to confabulation and are often less robust against adversarial prompting or "jailbreaks." A user might initiate a conversation with casual chitchat but subtly shift context to seek financial advice. If the smaller model fails to detect this shift and generates plausible but incorrect financial guidance, it poses a severe reputational and legal risk to the institution. Consequently, these smaller models must be deployed with strict output guardrails that prevent the generation of transaction-related content.

7.3. Privacy Risks in Vector Architectures

The implementation of embedding similarity search involves storing vector representations of customer queries in a database. While high-dimensional vectors are often perceived as opaque or anonymized data, recent research challenges this assumption. [Morris et al. 2023] demonstrated that text embeddings can reveal nearly as much information as the original text through model inversion attacks. An attacker with access to the vector database and the embedding model could theoretically reconstruct sensitive financial queries (e.g., "I lost my card ending in 1234"). Therefore, despite the on-premise advantages discussed in Section 8, vector databases containing financial data must be secured with the same encryption and access rigor as raw text databases, rather than treated as non-sensitive hashes.

7.4. Operational Resilience and Error Handling

Finally, while the embedding similarity search achieved a high accuracy of 98.52%, the remaining error rate implies that in a system processing millions of requests, thousands of users may still be misrouted. In alignment with the "Processes" pillar of the Information Systems framework, the technical solution must be paired with robust operational processes. A deterministic fail-safe mechanism is essential; specifically, the system must allow users to explicitly override the automated routing (e.g., "Speak to an agent") to ensure that algorithmic errors do not result in a dead-end for the customer.

Table 1. Accuracy for each method and label

Method	Accuracy				
	Relevant	**Unrelated**	**Chitchat**	**Spam**	**Total**
GPT Zero-Shot	92.41%	25.42%	16.94%	89.81%	53.78%
GPT Few-Shot	94.94%	86.44%	82.51%	95.54%	89.48%
GPT Fine-Tuned	96.84%	98.31%	**97.28%**	**99.36%**	97.93%
Embedding Similarity Search	98.73%	98.87%	**97.28%**	**99.36%**	98.52%
BERT based neural model	**99.37%**	**100.00%**	97.27%	**99.36%**	**98.96%**

8. Summary of Results

The accuracy for each label and the total accuracy of each of the five used methods is presented in Table 1.

GPT with Zero-Shot presented a 53.78% total accuracy, being particularly worse in labeling unrelated questions and chitchat messages. As shown in Fig. 2, chitchat messages were often classified as relevant or unrelated questions. There were also many cases where chitchat messages were not classified as any valid label (31.15%), most of these happened because the model tried to answer the user back, instead of classifying the text. It is also clear by visualizing the confusion matrix that the model was not able to consistently differentiate relevant and unrelated questions, considering messages more relevant that they actually were and labeling 64.97% of unrelated questions as relevant questions.

With the addition of a single example of each label in few-shot experiment, the result was considerably better. Total accuracy increased to 89.48% and the critical problems observed in zero-shot were mitigated, as observed in the confusion matrix. Chitchat was still invalidly classified and confused with unrelated and relevant questions, but the accuracy increased from 16.94% to 82.51%. Total response invalidity dropped from 9.78% to 2.51% and the incorrect labeling of unrelated questions as relevant decreased to 9.60%.

By fine-tuning the model, it was able to eliminate the problems observed in zero-shot labeling. Cases of invalid label being output were eliminated, as well as the prominent confusion between relevant and unrelated questions. The accuracy of all labels increased to over 96%, resulting in a total accuracy of 97.93%.

Both approaches that explore the embedding of the messages input — embedding similarity search and BERT based neural model — had good results, presenting accuracy over 98.00%. The Confusion Matrix of each method shows that no chronic problem was observed in either method.

Zero-Shot had the worst result in all categories, making it clear that the addition of examples in the prompt and fine-tuning the model are techniques that result in significantly better performance.

Despite the improvement observed with few-shot, it did not compete with the fine-tuned model for the classification problem. The use of more examples in prompt could result in better accuracy with a trade-off in cost added by the extra input tokens.

The fine-tuned model, routing with embedding similarity and BERT based neural

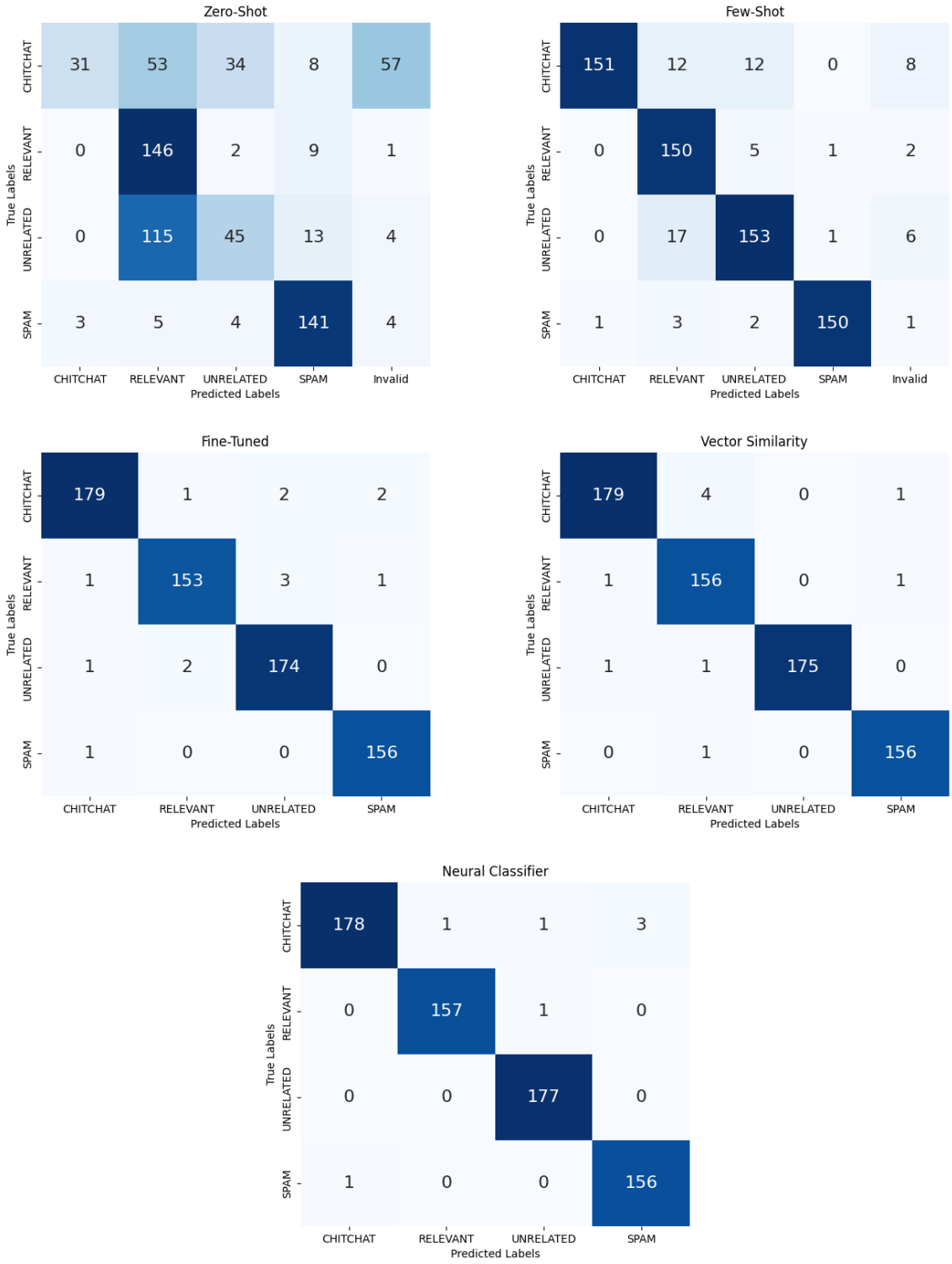

Figure 2. Confusion Matrices of methods used in the routing system.

Table 2. Resource usage for each method

Method	Accuracy	Params	Memory (GB)	Pricing (US$/M)	Extra costs
GPT Zero-Shot	53.78%	7B	6.52	8,52	-
GPT Few-Shot	89.48%	7B	6.52	18,10	-
GPT Fine-Tuned	97.93%	7B	1.63	-	Training
Embedding Similarity Search	98.52%	560M	1.04	0.32	VectorDB
BERT based neural model	98.96%	560M	1.04	-	Training

model achieved very similar accuracy across all labels. Therefore, the analysis of which to choose depends on factors other than precision.

The fine-tuning approach used a model with seven billion parameters, much larger than the 560 million parameter model used by the embedding approaches, making it more expensive to run the inference. However, the usage of a generative model allows a fine-tuning to perform more than one task. By extending the dataset to include expected responses for chitchat messages, the model could be trained to generate structured text that both classifies the text and answers the user in case of chitchat.

Embedding similarity search and BERT based neural model showed up as the cheapest and most reliable solutions. The embedding similarity approach is flexible and allows developers to extend the vector database with critical examples to increase accuracy while the neural approach requires re-training the model with the updated dataset. On the other hand, embedding similarity requires maintaining the vector database and querying it at every user message, possibly increasing the cost when compared to the classification head appended by the neural method.

Table 2 summarizes resource usage in each method. VRAM indicates the GPU memory necessary for loading the model's parameters. In practice, loading the model will consume extra memory depending on the framework used. Pricing indicates the resulting value in dollars paid for a million requests, considering the prompts described in this paper and considering a user query of 40 tokens, approximately 30 words. The zero-shot system prompt contains 115 tokens, while the few-shot prompt contains 289 tokens. Using as reference the price charged by Deepinfra of US$0.055 per million token for Mistral-7B-Instruct-v0.3[5] and US$0.010 for multilingual-e5-large[6] it is possible to estimate the price paid for using the text-classification via an external api. The GPT fine-tuned and the BERT based neural model methods do not have an associated pricing because they used a self-trained model and would require self-hosting it.

As shown in Table 2, Embedding Similarity Search is the cheaper option with equivalent results to the best alternatives, making it the recommended solution. Most vector databases providers offer a free-tier cloud solution for low scale, which wouldn't increase the costs for this method. The fine-tuning of the GPT or BERT based models is not time-consuming, but requires time dedicated to gathering good data for training and

[5]https://deepinfra.com/mistralai/Mistral-7B-Instruct-v0.3/api accesed at 09/13/2024

[6]https://deepinfra.com/intfloat/multilingual-e5-large/api accessed at 09/13/2024

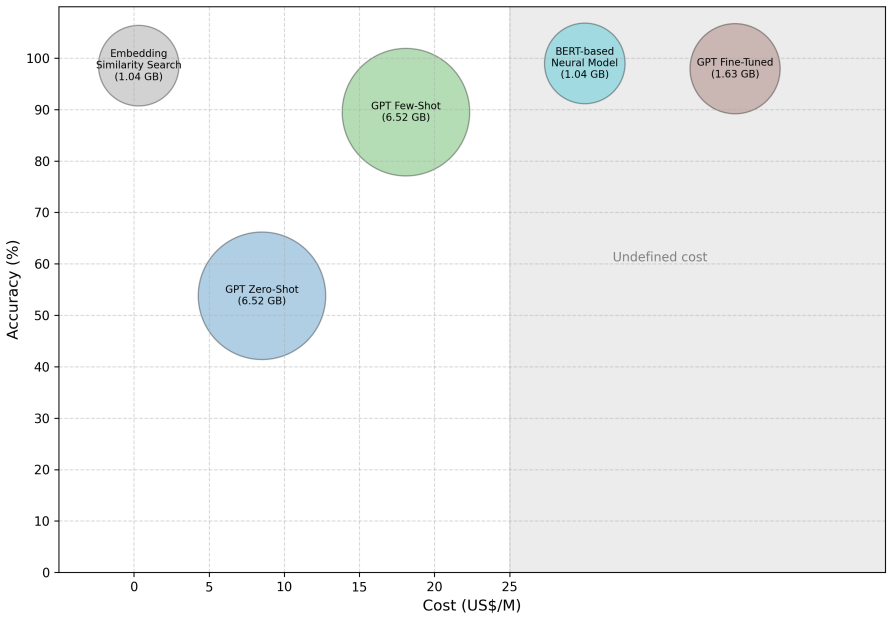

Figure 3. Tradeoff between accuracy, cost and memory footprint. Cost is measured as the estimated price for classifying a million user requests. The sizes of the bubbles represent the required memory for loading the model.

specific hardware with dedicated GPUs with enough VRAM, which increases the costs. After the training, a cloud or self-hosted server would also be needed to run the custom models. Given that the pricing analysis of the self-hosted options would require individual consideration of the load profile of the application, this paper does not estimate the cost of deploying GPT Fine-Tuned and BERT based neural models.

The use of pre-trained GPTs for this application, both zero-shot and few-shot, is not recommended, as they have both the highest costs and the lowest accuracies.

Figure 3 presents the results as a bubble chart for a clear visualization of the trade-offs.

9. Contributions and Impact to IS area

The results highlight how the choice of method for text classification in chatbots affects both accuracy and resource usage. Zero-shot GPT, with an accuracy of 53.78%, proved inadequate, particularly in handling chitchat and differentiating between relevant and unrelated messages. Adding examples in the few-shot approach significantly improved performance to 89.48%, though it still fell short of fine-tuned and embedding-based methods.

The fine-tuned GPT model, while highly accurate at 97.93%, demands more resources due to its larger size and higher inference costs, making it less suitable for scalable deployment. In contrast, embedding similarity search and BERT-based models achieved similarly high accuracies of 98.52% and 98.96%, respectively, while requiring fewer resources.

Among these, embedding similarity search stands out as the most practical solution. It does not require training, and performance can be enhanced by adding examples to the vector database to address edge cases. Additionally, both the embedding model and

vector database can be easily accessed through cloud providers, making it a flexible and cost-effective option for deployment.

The routing system demonstrated in this study proves to be reliable enough to significantly reduce the costs of chatbot systems by minimizing unnecessary reliance on LLMs. By effectively classifying messages and handling common tasks such as chitchat or spam using more cost-efficient methods like embedding similarity search or BERT-based models, the system can selectively route only the most complex queries to LLMs. This approach optimizes resource usage, ensuring high performance while reducing the financial and computational burden associated with running large-scale LLMs for every interaction.

This way, the project acknowledges the Information System tripod, composed of People, Processes and Technologies.

- **People (Social and Organizational impact)**: By correctly routing 96% of the customers' queries through the adequate system, the application reduces the risk of hallucinating and incorrectly guiding the user, reducing the risk of financial injury. Moreover, the overall economic saving of the application allows the use of chatbot customer support for mid-sized banking institutions.
- **Processes**: The deployment of a semantic routing system enables an automatic data curation. By aggregating relevant questions, the bank employees and researchers are able to identify the core components of their entire system from which more questions have been raised and can focus on improving the main LLM bot.
- **Technology**: The use of embedding models (multilingual-e5-large) and vector stores (FAISS) provides a language-agnostic solution that achieved 98.52% accuracy at $0.32 per million requests. This stack enables seamless integration with existing banking APIs while maintaining data compliance through on-premises deployment options. The open-source semantic-router library further allows customization for other domains.

Despite the promising results achieved with the text classification methods evaluated in this study, there are several limitations that should be acknowledged.

First, the classification accuracy, although high, may not fully generalize to all real-world scenarios. The dataset used in our experiments is domain-specific, and the performance of the models might degrade when applied to new domains or languages. This is particularly relevant for applications targeting a broader or multilingual audience, where the classifiers may require additional training or fine-tuning on more diverse data.

Second, the reliance on embedding-based methods, such as vector similarity search, introduces challenges related to latency and storage. As the size of the embedding index grows, the time required for similarity search may increase, potentially impacting the responsiveness of the system. Additionally, maintaining and updating the embedding index for continuous learning or new data can be resource-intensive.

Finally, while the classifiers developed in this study are effective in routing messages based on predefined categories, they may struggle with more nuanced or ambiguous queries. For example, messages that span multiple categories or involve implicit context might not be accurately classified. Further, incorporating user feedback or dynamically

adjusting the system based on evolving user intent would require more sophisticated techniques, which were not explored in this study.

These limitations suggest that future work should focus on improving generalization, optimizing resource usage, and enhancing the system's ability to handle ambiguous or complex queries.

Acknowledgments

This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – (CAPES, Brazil) – Finance Code 001; Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq; and Fundação de Amparo à Pesquisa do Espírito Santo (FAPES, Brazil) - grants 2021-07KJ2.

References

Agrawal, A., Kedia, N., Panwar, A., Mohan, J., Kwatra, N., Gulavani, B. S., Tumanov, A., and Ramjee, R. (2024). Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve. *arXiv preprint arXiv:2403.02310*.

Azharudeen, M. (2024). Beyond basic chatbots: How semantic router is changing the game. `https://medium.com/ai-insights-cobet`. Accessed: June 26, 2024.

Casanueva, I., Temčinas, T., Gerz, D., Henderson, M., and Vulić, I. (2020). Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Chen, L., Zaharia, M., and Zou, J. (2023). FrugalGPT: How to use large language models while reducing cost and improving performance. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Cunningham, P. and Delany, S. J. (2021). k-nearest neighbour classifiers: A tutorial. *ACM Computing Surveys*, 54(6):1–25.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. (2023). GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*.

Fuster, A., Goldsmith-Pinkham, P., Ramadorai, T., and Walther, A. (2022). Predictably unequal? The effects of machine learning on credit markets. *The Journal of Finance*, 77(1):5–47. Discusses how algorithmic maximization can lead to distributional shifts that disadvantage vulnerable groups.

Gasparetto, A., Marcuzzo, M., Zangari, A., and Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. *Information*, 13(2):83.

Gonçalves, A. et al. (2025). Accessibility in banking chatbots: An analysis of portuguese as a second language. In *Brazilian Symposium on Information Systems (SBSI)*.

Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

He, J. and Zhai, J. (2024). FastDecode: High-throughput GPU-efficient LLM serving using heterogeneous pipelines. *arXiv preprint arXiv:2403.11421*.

Horsey, J. (2024). Semantic router superfast decision layer for LLMs and AI agents. `https://www.geeky-gadgets.com`. Accessed: June 26, 2024.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. (2025). Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. (2023). Efficient memory management for large language model serving with PagedAttention. *arXiv preprint arXiv:2309.06180*.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. (2023). AWQ: Activation-aware weight quantization for LLM compression and acceleration. *arXiv preprint arXiv:2306.00978*.

Lindsey, J., Gurnee, W., Ameisen, E., Chen, B., Pearce, A., Turner, N. L., Citro, C., Abrahams, D., Carter, S., Hosmer, B., et al. (2025). On the biology of a large language model. *Transformer Circuits Thread*.

Malkov, Y. A. and Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Manias, D. M., Chouman, A., and Shami, A. (2024a). Semantic routing for enhanced performance of LLM-assisted intent-based 5G core network management and orchestration. *arXiv preprint arXiv:2404.15869*.

Manias, D. M., Chouman, A., and Shami, A. (2024b). Towards intent-based network management: Large language models for intent extraction in 5G core networks. *arXiv preprint arXiv:2403.02238*.

Manik, L. P., Akbar, Z., Mustika, H. F., Indrawati, A., Rini, D. S., Fefirenta, A. D., and Djarwaningsih, T. (2021). Out-of-scope intent detection on a knowledge-based chatbot. *International Journal of Intelligent Engineering and Systems*, 14(5).

Morris, J. X., Kuleshov, V., Shmatikov, V., and Rush, A. M. (2023). Text embeddings reveal (almost) as much as text. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mussmann, S. and Ermon, S. (2016). Learning and inference via maximum inner product search. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48, pages 2587–2596.

Ong, I., Almahairi, A., Wu, V., Chen, W.-L., et al. (2024). RouteLLM: Learning to route LLMs with preference data. *arXiv preprint arXiv:2406.18665*.

Pires, R., Abonizio, H., Almeida, T. S., and Nogueira, R. (2023). Sabiá-65B: A large language model for Portuguese. *arXiv preprint arXiv:2312.11991*.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: Pre-trained BERT models for Brazilian Portuguese. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pages 403–417. Springer.

van der Heijden, N., van der Linde, J., Vossen, P., and Shutova, E. (2025). How much do LLMs hallucinate across languages? On multilingual estimation of LLM hallucination in the wild. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Finds that smaller LLMs exhibit significantly larger hallucination rates than larger models.

Verdecchia, R., Sallou, J., and Cruz, L. (2023). Green AI: A systematic literature review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.

Wang, Z., Pang, Y., and Lin, Y. (2023). Large language models are zero-shot text classifiers. *arXiv preprint arXiv:2312.01044*.

Wei, J., Yang, C., Song, X., Lu, Y., Hu, N., Tran, D., Peng, D., Liu, R., Huang, D., Du, C., et al. (2024). Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Huggingface's transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Women's World Banking (2021). Algorithmic bias, financial inclusion, and gender. `https://www.womensworldbanking.org/insights/algorithmic-bias-financial-inclusion-and-gender/`. Accessed: 2025-01-02.