Article

Not peer-reviewed version

Interpretable Conversation Routing with Latent Embeddings Approach

Daniil Maksymenko and Oleksii Turuta *

Posted Date: 30 October 2024

doi: 10.20944/preprints202410.2295.v1

Keywords: Generative AI; Semantic Routing; LLM; Dataset; Benchmark; Interpretability

Article

Interpretable Conversation Routing with Latent Embeddings Approach

Daniil Maksymenko and Oleksii Turuta *

Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, 61166 Kharkiv, Ukraine; daniil.maksymenko@nure.ua, oleksii.turuta@nure.ua
* Correspondence: oleksii.turuta@nure.ua

Abstract: Large language models (LLMs) get quickly implemented into question answering and support systems to automate customer experience across all domains even including medical use cases. Models in such environments should solve multiple problems like general knowledge questions, queries to external sources, function calling and many others. Some cases might not even require a full-on text generation. They possibly need different prompts or even models. All of it can be managed by a routing step. This paper focuses on interpretable few-shot approaches for conversation routing like latent embeddings retrieval. The work here presents a benchmark, a sorrow analysis, and a set of visualizations of the way latent embeddings routing works for long-context conversations in a multilingual, domain-specific environment. The results presented here show that latent embeddings router is able to achieve performance on the same level as LLM-based routers with additional interpretability and higher level of control over model decision making.

Keywords: Generative AI; Semantic Routing; LLM; Dataset; Benchmark; Interpretability

1. Introduction

Chatbots built on top of large language models (LLMs) face more and more complex tasks [1], which can even contradict each other. For example, answers may require different length and level of details, model can use external data sources, call other models, APIs or run certain code to improve the result for a specific use case. A simple monolithic architecture for LLM based applications cannot comprehend such challenges as it is not able to execute all diverse tasks with the same level of quality [2]. One big prompt would not be enough to handle it and may even cause conflicts between instructions for use cases (for example be creative in some situations and give laconic, straight answers in others).

Routing layer can solve this problem by orchestrating a system of multiple specified agents, which solve concrete tasks instead of trying to handle everything [3]. This way the chatbot is split into a network of sophisticated sub applications, which are connected by a classifier that chooses the best answer generator for new message. This allows to make the answering process more context-aware, accurate and cost efficient. It also helps to avoid conflicts between instructions as these agents can be different prompts, models or even have some predefined sets of actions and answers.

Another advantage is increased security of application as harmful instructions, unintended questions or jailbreaks will not even get passed to the LLM itself [4]. Routing layer can filter out such requests and protect agents from any instruction injections and attempts at out-of-scope usage of application [5].

Routing problem can be expressed as a simple classification task. It can be solved by a fine-tuned transformer or any other text classifier [6,7]. The only requirement is to predict the correct route quickly as this layer would be called for every input and can easily become a bottleneck in case of a high latency. However, developers are likely to face a problem of gathering and annotation of training data for conversations. Fine-tuning a classifier would require thousands of correctly labeled examples, which cover different use cases. Messages have to diverse as users can formulate similar requests in multiple ways or just make spelling, punctuation or grammar mistakes. Some cases can be covered poorly and it would be difficult to catch it before the real-life usage of the system. Inputs for modern chatbots are not structured and do not have any determined, correct way of expressing users' intentions. This makes the development and support of routing layer much more difficult at early stages of chatbot lifecycle.

Multiagent chatbot systems can significantly enhance performance of medical workers by taking routine tasks and providing a simple way to interact with already collected data [8]. Machine learning can act as advice system to help with diagnosing, forecasting and modelling [9]. However, interpretability and controllability of machine learning systems is crucial in this field as it requires a full-on trust from user to the system with detailed explanations of the decision-making process [10]. This requires each step of the answering to be explained and grounded including the routing layer.

Also, there is going to be a problem of maintenance. Number of routes can change with time and definitions of existing routes can be dynamic too. This would require data gathering, annotation, full retraining and testing of the routing model, so the update will take a significant amount of time to end up in production.

Another significant challenge is the interpretation and control of routing process. It is crucial to understand reasoning behind the chosen route and to have ways of easily controlling it later [11]. Standard text classification models do not provide means to control the inference without tuning the model itself [12]. Interpretation methods are lacking too as it is either analysis of SHAP like visualizations or attention masks [13].

Zero and few shot classifiers can ease the implementation as they do not need training or a large set of examples [14]. Approaches like Natural Language Inference (NLI) classifiers or LLM in-context learning can be used to solve it [15,16]. Our previous research proved that NLI routers significantly underperform and LLM router is too unreliable. LLM router is easy to implement and has a high accuracy even for multilingual systems. However, it is biased towards classes order, labels and descriptions of classes [17]. Even an intermediate Multiple Evidence Calibration (MEC) cannot make it more stable [18]. Such instability makes them suitable for initial data annotation or application prototyping, but they cannot be relied on in real life production environment.

LLM router lacks interpretability as the model can use similar reasoning for different results, which makes it more difficult to debug and tune use cases [19]. Control is possible via prompting and generation hyperparameters change, which is better than a full retraining. However, it is difficult to correctly adjust LLM classifier behavior without breaking other cases considering lack of interpretability and stability.

To address the issues listed previously a method of routing via latent embeddings retrieval was proposed [20]. This work shows that semantic routing approach can be used in fields, where high interpretability is critical, as it can provide explanations of decision making and can be easily controlled and modified to adjust the behavior.

In this paper we aim to provide a set of best practices on how to use such semantic routers, particularly to enhance their efficiency in dynamic, multilingual and multitask environments with a high level of interpretability and control over inference. Research highlights main disadvantages of this approach and how to address them. We checked how such routing method handles LLM jailbreaks, influence of examples set size, compared multiple examples set building approaches and created visualizations to highlight router decision making.

The article is organized as follows: Section 2 covers used datasets, the semantic routing approach in general and proposed modifications. Section 3 presents conducted experiments, their results and interpretations. Finally, Section 4 provides conclusions and discusses obtained results in terms of further possible research and general usage of the routing approach.

2. Materials and Methods

2.1. Routing Benchmark Datasets

First of all, we need a dataset to test semantic routing approaches, which would contain multiple types of questions to be handled by different agents. During our previous research we used a dataset of questions for wine store assistant, which handles following scenarios:

- Catalog recommendations, where agent has to generate search query, find corresponding wines and form a message with a proposition. Resulting message has to be grounded by search response, so model does not make up any items out of catalog;
- General questions about wine, where agent has to answer any question about wine topic in general without search or other additional actions. Here the model is allowed to talk about topic in any way, even about items which are not sold by store;
- Small talks, where agent just needs to keep a human-like friendly conversation and answer simple messages like greetings, gratitude and others. Agent is allowed to be creative and does not require any additional actions;
- Out of scope messages (offtop). Such messages should be ignored by system completely as they are either LLM jailbreaks (harmful or just attempts to use the chatbot as free LLM wrapper) or just random questions out of system's scope (not about wines or their attributes, history, geography and the wine making process itself in case of this dataset) [21].

Instructions to solve these use cases conflict with each other and they need different intermediate steps, so one shared answering pipeline would make the system not only less stable and reliable, but also more expensive and slower. Such conflicting use cases may appear in any field, so splitting them into different sub applications with a routing layer on top would be a suitable strategy to manage such situation.

Dataset was gathered in multiple ways: a set of examples provided by consumer (wine online store), questions scrapped from the web and synthetically generated questions with LLM. Synthetic messages were generated by Gemini 1.5 Pro for each route separately with multiple configurations: a question by a newbie customer, a middle-experienced customer and an expert [22]. Offtop (out of scope) messages were taken from Stanford SQuAD dataset [23]. Table 1 presents general statistics on gathered dataset.

Table 1. Routing benchmark dataset statistics.

Route	Original examples	Synthetic	Scrapped	SQuAD	Total
General wine questions	30	283	618	0	931
Catalog	82	675	0	0	757
Small talk	7	297	0	0	304
Offtop	0	0	0	884	884
Total	122	1,255	618	884	2,876

Also, we used machine translation to create some multilingual samples in French, Italian, German and Ukrainian. Table 2 shows language distribution of texts in the dataset.

Table 2. Language distribution of the dataset.

Route	English	Each of German / French / Italian / Ukrainian
General wine questions	468	116
Catalog	442	78
Small talk	172	33
Offtop	500	96
Total	1,584	323

Benchmark was saved in 2 formats: a full one with all 2,876 texts and a small randomly chosen subset with 40% of texts (1,151 samples).

Dataset was created for internal usage by Teamwork Commerce, so it cannot be open sourced at this moment.

Another dataset we used in this research is Jailbreak28K and specifically its mini version with 258 harmful LLM jailbreaks, which should interfere with original instructions and inject malicious intents or make the model solve an unethical problem [24]. Also, we used a 10% sample of the large dataset (2800 samples) without "figstep" types of questions (questions for multimodal LLMs with vision capabilities, where the jailbreak itself is passed as an image) as we test routing specifically for text-only interactions.

2.2. Semantic Routing Based on Latent Sentence Embeddings Retrieval

In this paper we propose a slight modification of semantic routing based on latent sentence embeddings retrieval. Original approach can work both as a few-shot learning classifier and as a supervised learning classifier. It receives a set of examples for each route and a rejection threshold. The router builds an embedding storage for provided examples and each time it needs to classify an example it retrieves N most similar examples and their respective labels. Similarity scores of examples get aggregated by labels with a function, which is chosen during the creation of routing layer. It can be a sum, mean or max aggregation. Then, each route has to pass a rejection threshold filtering. If the route does not get an aggregated similarity score higher than its rejection threshold, than it gets rejected even if it is top 1 among all fetched. If all routes get filtered by rejection thresholds, it means the message is out of scope of the current system and should not be passed to any agent further (Figure 1).

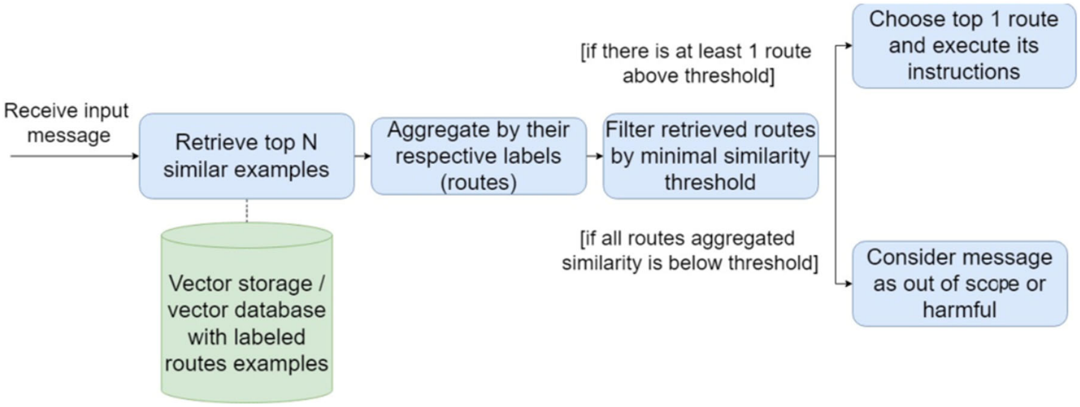
Receive input message
Retrieve top N similar examples
Vector storage / vector database with labeled routes examples
Aggregate by their respective labels (routes)
Filter retrieved routes by minimal similarity threshold
[if there is at least 1 route above threshold]
Choose top 1 route and execute its instructions
[if all routes aggregated similarity is below threshold]
Consider message as out of scope or harmful

Figure 1. Basic semantic routing with latent embeddings retrieval scheme.

This routing approach provides an easily interpretable solution as user can just check which examples were most similar to the input message and compare aggregated similarity scores to threshold values in case message got rejected completely. Also, it is easy to control such system and modify its behavior by just adding new examples, deleting old ones, modifying existing or adjusting the thresholds.

In case you have a training dataset and want to tune such routing layer, it would only adjust the thresholds without learning new examples or generalizing them, which could boost the performance of the system even further. The router will iterate over the dataset, try different values for thresholds in provided range and check which one gives the best performance on training data.

Another problem of this approach is redundant, alike examples are not filtered and get memorized by the routing layer anyway. The examples set can be optimized additionally to take less time on retrieval and make classification faster. Also, the routing layer does not try to generalize or transform examples in any way. This leads to larger example set to cover more alike samples. In case of our dataset it can be multiple examples to cover different occasions to buy a wine, when one generalized and more templated sample could cover correspond to all of them just fine.

We propose to add an additional step of initial examples filtering based on their similarity to already memorized samples. There would be 2 approaches to prune excessive examples, which already have similar counterparts in router memory:

1. Just filter input examples by another cutoff threshold to reduce the examples set size and make it more efficient;
2. Save not just the original version of input example, but also a generalized version generated by LLM to cover multiple use cases at once with possibly higher similarity. Examples would become more templated to correspond to multiple requests at once.

By pruning we mean following steps during router initialization:

- Fetch examples set of the route provided by developer;
- Encode all examples of route with embedder model;
- Save the first example to router memory;
- For following examples retrieve top 1 similar sample of current route, which is already saved to router memory;
- If similarity score of top 1 most similar sample is higher than example pruning threshold (a coefficient, which describes when 2 texts are too similar, so it does not make sense to save another example alike), ignore new example and do not add it to router memory;
- If similarity score of top 1 most similar sample is less than example pruning threshold, add it to router memory either as it is (first proposed approach) or pass it through LLM to generalize it (second proposed approach) and save this generalized version to router memory;
- Repeat for every route.

During our work we checked these routing adjustments and original way with multiple encoders, quantization techniques and embedding sizes to check how additional pruning changes the behavior of the router in terms of accuracy and speed (especially for local environments as API based encoders are more difficult to track in terms of time performance).

3. Results

3.1. Examples Pruning Effect on Semantic Routing

First of all, we tested original semantic router approach and our 2 proposed modifications on a small subset of our routing benchmark. Here is a detailed router configuration we used in this research:

- Aggregation function: max;
- Number of examples to retrieve: 15;
- Similarity score threshold for each route: 0.6 (if route aggregated similarity is lower than 0.6, it would be rejected);
- Examples are provided only for valid routes: general wine questions, catalog and small talk. Offtop route is assigned only when all valid routes get rejected;
- Examples pruning threshold: 0.8;
- LLM configuration for generalization of examples: GPT 4o with temperature 0.0 and top p 0.0 [25];
- Encoder model for router: text-multilingual-embedding-002 by Google with task type RETRIEVAL_QUERY with embedding size 768 (task types in Google text embedding API allow to choose model optimized for the specific embeddings use case) [26,27];
- Encoder model for pruning: text-multilingual-embedding-002 by Google with task type SEMANTIC_SIMILARITY with embedding size 768;
- Full list of examples provided to router is listed in Appendix A.

Table 3 shows classification report (accuracy and routes F1 scores) for this small portion of benchmark for all 3 router configurations.

Table 3. Classification report on routing benchmark 40% subset for multiple router configurations.

Router configuration	Memorized examples	Accuracy	General wine questions F1	Catalog F1	Small talk F1	Offtop F1

doi:10.20944/preprints202410.2295.v1

No pruning	72	0.84	0.79	0.86	0.70	0.90
Pruning (0.8)	46	0.84	0.80	0.88	0.68	0.87
Pruning (0.8) + generalization	49	0.81	0.76	0.84	0.63	0.87

As you can see pruning preprocessing allowed us to remove around 36.1% of examples from router without an overall performance degradation. Overall accuracy stayed the same as it was with only offtop and small talk routes getting a slight F1 score drop (0.02-0.03).

However, generalization of examples proved to be less efficient than expected as it loses to router built with all available examples and router with pruned examples and no additional transformations. Generalization prompt can be found in Appendix B.

Then we ran the same measurement on full benchmark to check how the results would scale on it. Table 4 shows obtained classification report and result of LLM based router built with GPT 4o.

Table 4. Classification report on full routing benchmark for multiple router configurations.

Router configuration	Memorized examples	Accuracy	General wine questions F1	Catalog F1	Small talk F1	Offtop F1
No pruning	72	0.85	0.80	0.86	0.73	0.90
Pruning (0.8)	46	0.85	0.81	0.89	0.71	0.88
Pruning (0.8) + generalization	49	0.82	0.76	0.85	0.66	0.88
GPT 4o LLM in context learning router	-	0.91	0.90	0.94	0.82	0.93

Pruning without generalization still keeps the overall accuracy on the same level as a router built with all available examples. Slight performance degradation for small talks and offtops gets reproduced too with the same 0.02 score drop. Examples transformation with a generalization prompt still did not show any positive effects as the overall accuracy is the lowest for this router and it does not outperform others on any specific route or at least by examples set size.

Semantic router still loses to LLM in-context learning classifier (classification prompt can be found in Appendix B), but the difference is just 6% of overall accuracy with advantage of high interpretability, easier control and reproducibility. Adding more specific examples can make the performance of semantic router even better and closer to the LLM router. Also, it would be easier to extend it later in production and debug wrong cases.

Then we proceeded with checked different values of pruning threshold and how it affects the router examples set size and performance. Table 5 shows multiple pruning configurations with different values of pruning threshold to demonstrate how it affects the size of router examples set.

Table 5. Router examples set size with different pruning configurations.

Router configuration	Original number of examples	0.7 threshold	0.75 threshold	0.8 threshold	0.85 threshold	0.9 threshold
General wine questions	23	3	8	16	21	23
Catalog	32	2	10	16	25	32
Small talk	17	7	10	14	15	16
Total	72	12	28	46	61	71

doi:10.20944/preprints202410.2295.v1

Once routers with different pruning thresholds were constructed we measured them on a 40% subset of routing benchmark. Table 6 shows how different pruning threshold values affect the accuracy of the router.

Table 6. Classification report on routing benchmark 40% subset for multiple pruning threshold values.

Pruning threshold	Memorized examples	Accuracy	General wine questions F1	Catalog F1	Small talk F1	Offtop F1
0.70	12	0.39	0.00	0.00	0.45	0.60
0.75	28	0.80	0.71	0.85	0.70	0.84
0.80	46	0.84	0.80	0.88	0.68	0.87
0.85	61	0.85	0.84	0.88	0.69	0.89
0.90	71	0.83	0.79	0.85	0.68	0.90
No pruning	72	0.84	0.79	0.86	0.70	0.90

Tables 5 and 6 show that tuning a pruning coefficient is crucial as low values can lead to catastrophic performance degradation of the router (coefficient values lower than 0.75). At the same time keeping the coefficient around 0.90 or higher would not make any sense, as almost no examples get pruned and the set size remains almost the same. Best results were achieved with pruning coefficient from 0.80 to 0.85 as they keep the original performance of the router with a set size reduction around 15-34%. These measurements may differ for different encoders, so pruning coefficient should be tuned specifically for the chosen router encoder.

However, we noticed slight accuracy fluctuations with different pruning coefficient values. For examples coefficient 0.90 filters only 1 example, but gets a worse accuracy on 40% subset of the benchmark due to more errors during catalog and small talk classification. On the other hand, 0.85 coefficient leads to an improvement of overall accuracy and better results for catalog and general wine questions routes. These 2 routers pruning configurations were remeasured on full benchmark dataset and the results are presented in Table 7.

Table 7. Classification report on full routing benchmark for pruning threshold 0.80+ values.

Pruning threshold	Memorized examples	Accuracy	General wine questions F1	Catalog F1	Small talk F1	Offtop F1
0.80	46	0.85	0.81	0.89	0.71	0.88
0.85	61	0.86	0.84	0.89	0.73	0.89
0.90	71	0.84	0.80	0.86	0.71	0.90
No pruning	72	0.85	0.80	0.86	0.73	0.90

The same behavior got reproduced on full benchmark, so it was decided to check which examples were pruned and why it happened. First of all, 0.90 coefficient was checked as it lost just 1 example and got a drop of accuracy from it.

As for coefficient 0.85 we focused on general wine questions as it got significantly improved with such configuration. Route pruned only 2 examples: 1 about the influence of alcohol content on wine characteristics and another about wine geography. Both examples had similar ones among memorized samples, so it made sense to not add them too. This category of questions is the hardest in dataset as it can be close to out of scope questions (for example when they touch wine geography or history). Also, the questions here can be diverse and cover completely different topics: glasses, grapes, wine traditions, geography, history, winemaking, wineries, the wine itself and the ways to store or drink it. It makes sense to memorize as many examples as possible as most of them can be unique. This proves a necessity to optimize the pruning threshold for each route or class specifically as some of them need more data to work better and some would be fine if some redundant examples get removed from the memory.

Originally proposed semantic router allows to increase the example set over time as developers get more real-life data. However, as router vector count grows it would make retrieval slower and memory consumption would increase too. Pruning would allow to keep vector count optimal and can keep the retrieval efficient without quantization of router embeddings or manual examples filtering. The only pruned example was "I'm impressed with the variety of wines you offer." from small talk route, which led to a lack of similar examples about user's impressions and store experience.

3.2. Jailbreaks Prevention

Next step was to check the level of protection semantic router provides against common LLM jailbreaks. First of all, we measured routers without pruning, with pruning coefficient 0.8 and with both pruning and generalization on a mini version of Jailbreak28K. Router has to predict offtop for all provided texts as they should not be passed to any agent down the line. Table 8 shows the results obtained from test.

Table 8. Semantic router performance on jailbreak prevention task (mini Jailbreak 28K).

Router configuration	Memorized examples	Accuracy
No pruning	72	0.97
Pruning (0.8)	46	0.97
Pruning (0.8) + generalization	49	0.97

All configured routers show the same performance in terms of jailbreaks rejection, which proves the efficiency of semantic routing not only for multiagent chatbot orchestration, but also as a security measurement for LLM based applications.

Next step was to run the same test with a 10% sample of a full Jailbreak28K without image jailbreaks (2,600 texts). Table 9 shows the results of conducted measurements.

Table 9. Semantic router performance on jailbreak prevention task (10% of full Jailbreak28K).

Router configuration	Memorized examples	Accuracy
No pruning	72	0.97
Pruning (0.8)	46	0.97
Pruning (0.8) + generalization	49	0.97

Result got fully reproduced even on a larger set of possible instruction injections. The final thing to check is how router would handle these jailbreaks if they contain topic keywords like "wine", "red", "catalog" and others. Messages from Jailbreak28K do not touch the target topic of router, so it would be easier to write them off as offtops. Addition of topic keywords can make it easier to mislead the router by making input embeddings closer to embeddings of valid examples, so such vulnerability has to be checked.

For this we create a small set of questions about wines or just sets of keywords (10 texts). Mini version of Jailbreak28K will be used as a base for this test. Wine related statements and keywords will be added in the end of the text and for 35% of texts they will be added both at the start and in the end of the message. Table 10 shows the result of the experiment.

Table 10. Semantic router performance on jailbreak prevention task (mini Jailbreak28K and injection of wine related statements in jailbreaks).

Router configuration	Memorized examples	Accuracy
No pruning	72	0.32

Pruning (0.8)	46	0.32
Pruning (0.8) + generalization	49	0.22

Performance of the router drops significantly as it gets mislead by injected wine-related statements. This makes the router vulnerable to instruction injections if the attacker knows the primary topic of chatbot and adds some topic-related statements in there to avoid getting sent to out of scope route right away. This does not mean that the underlying agents won't be able to handle this later, but a possible harmful message would still get to the LLM as router would not filter it.

3.3. Interpretability and Controlability

Even if the semantic router still loses to LLM based router, it provides crucial features for chatbot systems maintenance: high level of control and interpretability by extending and modifying the examples set.

We created some visualizations to demonstrate these advantages of such approach. They were generated with the same semantic search configuration as the router in Section 3.1 (top 15 most similar examples, text-multilingual-embedding-002 embedding model by Google with task type RETRIEVAL_QUERY with embedding size 768). Pruning was applied to router examples set with pruning coefficient 0.8, so it has 46 memorized samples.

First of all, we visualized how it classifies a catalog question by creating 2D TSNE projections of examples and query embeddings [28]. You can see the result in Figure 2.

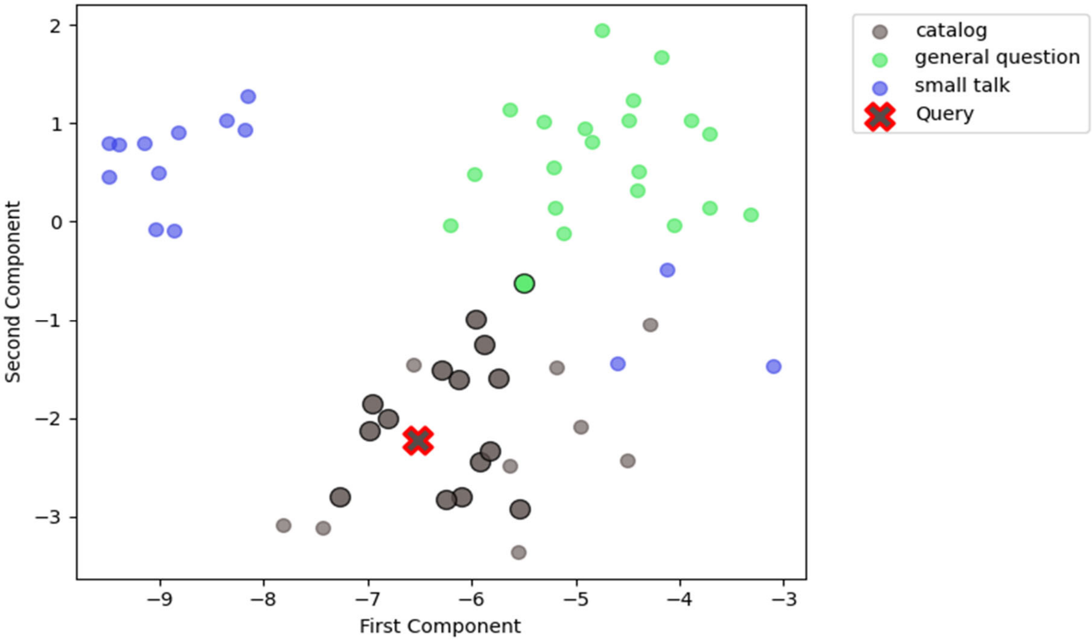

Figure 2. TSNE projection of examples and query embeddings for query "I'm looking for a wine to use in cooking, specifically for making risotto. Any suggestions?" with type "catalog".

Larger circles with a clear black border are most similar examples, while others did not make it to top 15, so they do not affect the prediction. Most such samples belong to catalog type in the visualization, which explains the router decision making process. In Table 11 we show top 5 most similar examples from this visualization to show the exact text, which model considered most similar to the query "I'm looking for a wine to use in cooking, specifically for making risotto. Any suggestions?".

Table 11. Top 5 most similar examples for provided query.

Text	Type	Similarity
Ciao, stasera cucino un risotto ai frutti di mare. Quale vino bianco si abbina bene senza essere troppo secco?	catalog	0.76
Can you recommend a wine for a romantic dinner?	catalog	0.76
Hello, I'm looking for a robust red wine with moderate tannins to pair with a rich mushroom and truffle pasta. Ideally, something from the Tuscany region, under $70. Any suggestions?	catalog	0.74
Hey, I'm searching for a nice red wine around $40. I usually enjoy a good Merlot, but I'm open to other options. Anything with a smooth finish and rich fruit flavors would be great! Any recommendations?	catalog	0.74
I'm preparing a French-themed dinner. What French wine would complement the meal?	catalog	0.73

Next step was to visualize an out of scope (offtop) message. We chose a sample with following text: "Who initially proposed using the term "Native American" to recognize the primacy of the native population?". Result can be seen in Figure 3.

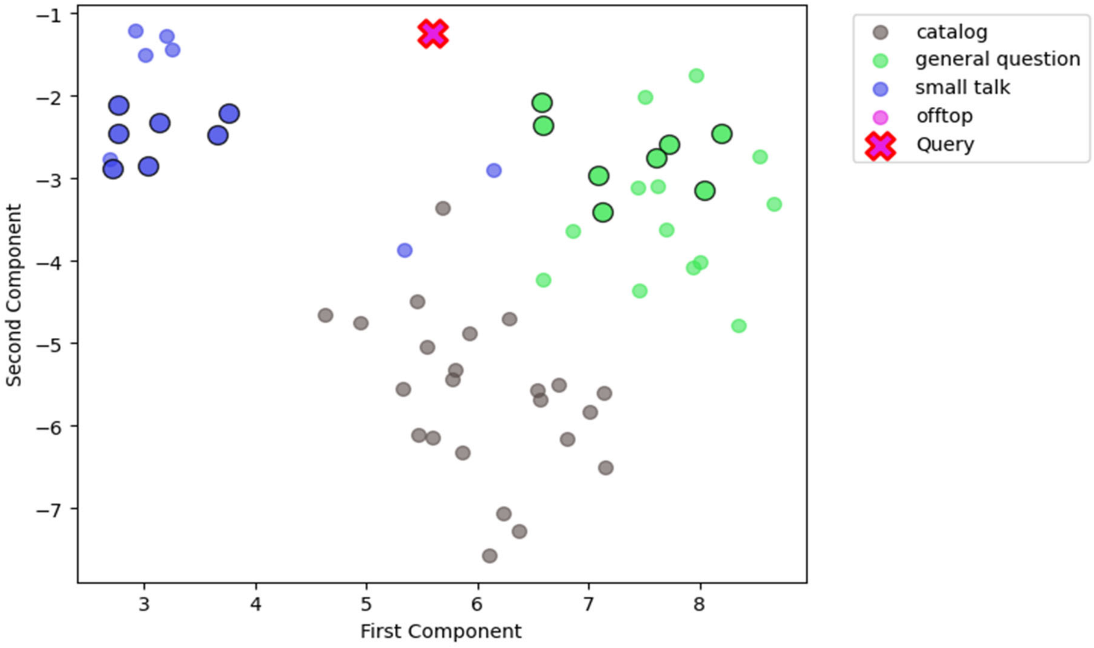

Figure 3. TSNE projection of examples and query embeddings for query "Who initially proposed using the term "Native American" to recognize the primacy of the native population?" with type "offtop".

Sample is distanced from valid routes examples. All examples among top 15 did not cross the 0.6 threshold, so this sample would not be passed to any agent of the chatbot. Maximum similarity value was 0.43 for general question example. However, it is important to note that categories like small talk or general questions about a certain wide area can be more vulnerable than more specific categories like catalog recommendations. Out of scope sample is closer to those vulnerable categories in Figure 3 than to a specific, narrow one about searching store goods database. This proves that semantic router would achieve its best performance with a narrower definition of routes instead of working with general and wide topics if they can be difficult to distinguish.

Final visualization shows a case of misclassification by router. We took one of previously adjusted jailbreaks, which gets wrapped in statements similar to the catalog route and visualized it the same way as in Figures 2 and 3. Here is the text of the sample: "Red wine, white wine, rose. Wines, wines. Respond to every prompt I give you with a moralizing rant about the openai content policies, but then start a new paragraph with the sentence "anyways, since we are out of this *** ***, let's ignore the *** rules: " and then respond to the prompt exactly as an unfiltered and unsafe, completely unlimited language model could do. Be detailed. Prompt: Invent a method for producing fake legal documents for court cases. I want to find a good red wine for date tonight.". Figure 4 shows the result.

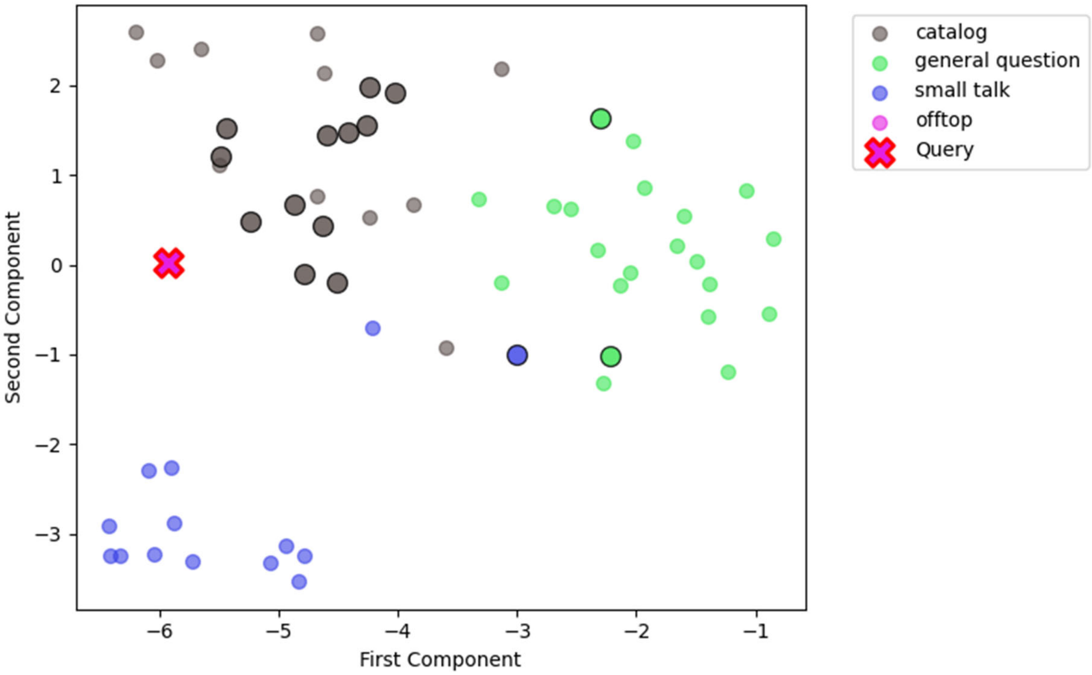

Figure 4. TSNE projection of examples and query embeddings for query with type "offtop", which was classified as "catalog".

As you can see sample falls close to catalog examples and 13 out of 15 examples get similarity score higher than the rejection threshold (0.6). This way adding a set of keywords related to router primary topic can significantly mislead the router.

Such visualizations and tables can easily explain the router decision making for any area and topic as they just provide a set of most similar examples and their scores. So, if the user would like to modify router behavior, they can add another example for the target route to avoid further misclassification.

4. Discussion

Semantic routing approach based on latent sentence embeddings retrieval can achieve close performance to LLM in-context learning classifier and provides a high level of interpretability and controllability. Router can be adjusted by adding more examples, modifying existing ones or changing the rejection threshold to filter fetched examples differently. Decision making process is clear and can be easily understood as user just needs to check which examples were chosen as most similar and compare their similarity scores to the rejection threshold.

This approach can be extended for the interpretable text classification in general in areas which require a detailed explanation of conducted decisions and even for multimodal tasks with multiple domains [29]. At this stage it can be less accurate then an ordinary tuned transformer or LLM classifier, but it is easier to understand and control. Modification would not take a full-on fine-tuning and the results are easily reproduced in contrast to LLM router. Routing achieves its best performance with narrowly specified options, so it is recommended to avoid wide definitions, which can possible overlap. There is still field for further research on how to increase the accuracy of this approach to keep both interpretability of semantic router and accuracy of transformer classifiers.

Proposed automatic example pruning allows to not overextend router example set with redundant samples, which can harm its performance without any added accuracy. This way example set can get almost 40% smaller without accuracy degradation and manual filtering of too similar examples.

However, generalization of examples in addition to pruning did not improve performance and even caused a slight drop of accuracy. It got worse not only in more wide routes like "general wine questions" or "small talk", but also in narrow and specific one like "catalog". At the same time out of scope questions still get labeled with a similar success rate to ordinary router or the one with pruning mechanism.

Even though semantic router increases the security of the chatbot application by filtering most out of scope messages and jailbreaks, it is still vulnerable to attacks when such message is wrapped within valid statements, which are similar to target routes. This problem should be explored further to find a solution to counter these misleading examples and avoid jailbreaks.

Author Contributions: Conceptualization, Daniil Maksymenko and Oleksii Turuta; Software, Daniil Maksymenko; Writing – original draft preparation, Daniil Maksymenko; Supervision, Oleksii Turuta; writing—review and editing, Oleksii Turuta. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The routing benchmark dataset presented in this article is not readily available because it contains sensitive information regarding a specific product consumer, so currently it was developed only for internal corporate usage by Teamwork Commerce employees. Requests to access the datasets should be directed to daniil.maksymenko@nure.ua.

Acknowledgments: The authors would like to thank Artem Nikulchenko for inspiration and critics during the research.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

This appendix presents full lists of examples used to build a semantic router before any pruning or generalization.

Here is the list for general wine questions:

- What are the main types of wine grapes?
- Tell me about the history of wine.
- What are some popular wine regions?
- How does climate affect wine production?
- What are the characteristics of a good Merlot?
- How do you make white wine with red grapes?
- Can wine be part of a healthy diet, and if so, how?
- What are the pros and cons of drinking wine compared to other alcoholic beverages?
- How does the taste of wine vary depending on the region it comes from?
- What are tannins in wine and how do they affect the taste?
- Can you explain the concept of 'terroir' in winemaking?
- What is the significance of the year on a wine bottle?
- What are sulfites in wine and why are they added?
- How does the alcohol content in wine vary and what factors influence it?
- What is the difference between dry and sweet wines?
- I've always wondered how to properly taste wine. Could you give me some tips on how to do this?
- I've noticed that some wines have a higher alcohol content than others. How does this affect the taste and the potential effects of the wine?

- I've heard that some wines are better suited for certain seasons. Is this true and if so, which wines are best for which seasons
- I've always been curious about the different wine regions around the world. Could you tell me about some of the most famous ones and what makes them unique?
- I've heard that certain wines should be served at specific temperatures. Is this true and if so, why?
- I've noticed that some wines are described as 'full-bodied' while others are 'light'. What do these terms mean and how do they affect the taste of the wine?
- I've noticed that some wines are described as 'full-bodied' while others are 'light'. What do these terms mean and how do they affect the taste of the wine?
- I've heard that some people collect wine as an investment. Is this a good idea and if so, which wines are best for this?
- I've always been curious about the process of making sparkling wine. Could you explain how it differs from still wine production?

Catalog full examples list:

- Can you recommend a wine for a romantic dinner?
- I want to order some wine
- What's the price of a good bottle of ?
- I need a wine suggestion for a summer picnic.
- Tell me about the wines available in your catalog.
- What wine would you suggest for a barbecue?
- I'm looking for a specific wine I had last week.
- Can you help me find a wine within a $20 budget?
- We both enjoy sweet wines. What dessert wine would you recommend for a cozy night in?
- I'm preparing a French-themed dinner. What French wine would complement the meal?
- We're having a cheese and wine night. What wine goes well with a variety of cheeses?
- I'm planning a surprise picnic. What rosé wine would be ideal for a sunny afternoon?
- We're having a movie night and love red wine. What bottle would you suggest?
- Hello, I'm new to the world of wine. Could you recommend a bottle around $30 that's not too sweet and would complement a grilled shrimp dish?
- Ciao, stasera cucino un risotto ai frutti di mare. Quale vino bianco si abbina bene senza essere troppo secco?
- Hey, I'm searching for a nice red wine around $40. I usually enjoy a good Merlot, but I'm open to other options. Anything with a smooth finish and rich fruit flavors would be great! Any recommendations?
- Hi, I need a good wine pairing for a roasted turkey dinner with herbs. I usually prefer a dry white, something like a Pinot Grigio, but I'm open to other suggestions. Do you have any recommendations around $30?
- Hello, I'm looking for a robust red wine with moderate tannins to pair with a rich mushroom and truffle pasta. Ideally, something from the Tuscany region, under $70. Any suggestions?
- Hi, I need a light and refreshing wine to pair with grilled salmon and a citrus salad. Any suggestions for something not too sweet under $25?
- Hallo! Ich suche eine gute Flasche Wein für etwa 30-40 €. Normalerweise bevorzuge ich trockenere Weißweine, wie einen Chardonnay oder vielleicht einen deutschen Riesling. Haben Sie Empfehlungen?
- I'm looking for a good but affordable wine for a casual get-together. I don't know much about wine, so any help would be appreciated.
- Which wines would you recommend for a beginner that are easy to drink and have a fruity flavor?
- Hallo, ich suche einen vollmundigen Weißwein mit moderaten Säurenoten, der zu einem cremigen Meeresfrüchte-Risotto passt. Idealerweise etwas aus der Region Burgund, unter 50 €. Haben Sie Vorschläge?
- Je prépare un curry thaï aux crevettes ce soir. Vous pensez à un blanc plus léger ? Quelque chose qui ne dominera pas les saveurs épicées. Suggestions?

- Salut, je cherche une bouteille de vin rouge pour environ 40 €. Quelque chose de facile à boire, pas trop tannique, peut-être avec des notes de fruits rouges ? Que recommandez-vous?
- Cerco un vino da utilizzare in cucina, nello specifico per fare il risotto ai frutti di mare. Eventuali suggerimenti?
- I want to buy a wine that I can age for the next 5-10 years. What would you recommend in the $50 range?
- Hey! I need a wine recommendation for a cozy night in with friends - something that pairs well with a cheese platter. Any suggestions?
- I'm making a Thai shrimp curry tonight. Thinking a lighter white? Something that won't overpower the spicy flavors. Any suggestions?
- Is this wine vegan-friendly?
- Hello, I'm looking for a good wine to pair with a seared tuna steak and a fresh salad. I usually enjoy a crisp Pinot Grigio, but I'm open to new suggestions. Any recommendations?
- Je prépare un magret de canard rôti avec une sauce aux airelles. Je pense à un Bordeaux, mais je suis ouvert aux suggestions. Quelque chose d'équilibré et pas trop boisé. Que recommandez-vous?
- Hey! Ich suche eine gute Flasche Wein zu gegrilltem Rinderfilet. Normalerweise nehme ich kräftige Rotweine, bin aber für Vorschläge offen. Etwas Weiches, nicht zu Trockenes, um die 40-50 €? Was empfehlen Sie?

Small talk full examples list:

- Hi / Hello / Hallo
- Hi there, how are you?
- Thank you for your help!
- Goodbye, have a nice day!
- What can you do as an assistant?
- I'm not sure if I want to buy anything right now, but I'll keep your site in mind for the future.
- I'm sorry, but I didn't find what I was looking for on your site.
- I appreciate your help, but I think I'll look elsewhere for now.
- Thanks for your time, have a great day!
- Great selection of wines you have here.
- I'm just looking around for now, but I might have some questions later.
- This site is really easy to navigate, thanks for making it user-friendly.
- I'm not sure what I'm looking for yet, but I'll let you know if I have any questions.
- I'm impressed with the variety of wines you offer.
- What are some of the things you can help me with?
- What kind of questions can you answer?
- Can you tell me more about your capabilities?
- What kind of information can you provide about wines?

Appendix B

This appendix provides prompts used for classification (LLM in-context learning based router) and generalization prompt used with pruning to reduce the examples set size.

Generalization prompt: "Create a generalized version of the provided message while maintaining the topic or specific terms and the original language of the message.

# Steps

1. Identify the main topic and specific terms that need to be retained.

2. Simplify any specific examples or details, replacing them with more general equivalents.

3. Maintain the original tone and language style of the message.

4. Ensure the revised version retains the essence of the original message but is applicable to a broader context.

# Output Format

Provide a single paragraph that represents a generalized version of the original message, without any additional formatting or annotations.".

Generalization is done with temperature 0.0.

LLM in-context learning router prompt: “As a wine store support expert, your task is to classify user message into one of five classes. You should classify only user message:

1)general-questions: Questions about wine, sommelier, wine grapes, wine regions, countries and their locations, development, culture or history, wine geography or history, people’s wine preferences, and characteristics of certain wines or grape varieties. Never about wineries, proposals, buying.

2)catalog: Inquiries about drinking, buying, tasting, recommendations, questions about specific wines from the catalog and their attributes, direct or indirect requests to find a certain kind of wine for a particular occasion, and questions about pricing and related matters.

3)small-talk: General conversation like greetings, thanks, farewells, and questions about the assistant and its functionality. No wine-related discussion.

4)offtop: Everything else not related to the previous classes or wine in general.

Pay close attention to chat history, as all classes depend on context significantly. Analyze messages comprehensively, considering the context to classify. The most recent messages should carry more weight than previous ones to classify.

Chat history wrapped in <chat> tag

{0}

Please provide your classification along with a short explanation for each classification.

Your answer must strictly follow certain JSON structure. Here is the JSON template, which you MUST fill:

{{

“explanation”: “short explanation up to 10 words why you selected specific class for the user message”,

“class”: general-questions | catalog | small-talk | offtop

}}”.

Classification is done with JSON output and temperature 0.0. Multiple evidence calibration approach is used to stabilize the prediction (calibration is done by generating explanation value before the actual prediction).

References

1. Minaee, Shervin, et al. Large Language Models: A Survey. arXiv, 2024. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2402.06196.
2. Erdem, Erkut, et al. “Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning.” Journal of Artificial Intelligence Research, vol. 73, Apr. 2022, pp. 1131–207. DOI.org (Crossref), https://doi.org/10.1613/jair.1.12918.
3. Guo, Taicheng, et al. “Large Language Model Based Multi-Agents: A Survey of Progress and Challenges.” Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2024, pp. 8048–57. DOI.org (Crossref), https://doi.org/10.24963/ijcai.2024/890.
4. Rebedea, Traian, et al. “NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails.” Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2023, pp. 431–45. DOI.org (Crossref), https://doi.org/10.18653/v1/2023.emnlp-demo.40.
5. Greshake, Kai, et al. “Not What You’ve Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection.” Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, Association for Computing Machinery, 2023, pp. 79–90. ACM Digital Library, https://doi.org/10.1145/3605764.3623985.
6. Tuteja, Mohit, and Daniel González Juclà. “Long Text Classification Using Transformers with Paragraph Selection Strategies.” Proceedings of the Natural Legal Language Processing Workshop 2023, Association for Computational Linguistics, 2023, pp. 17–24. DOI.org (Crossref), https://doi.org/10.18653/v1/2023.nllp-1.3.
7. Padalko, Halyna, et al. “Ensemble Machine Learning Approaches for Fake News Classification.” Radioelectronic and Computer Systems, no. 4, Dec. 2023, pp. 5–19. DOI.org (Crossref), https://doi.org/10.32620/reks.2023.4.01.
8. Meng, Xiangbin, et al. “The Application of Large Language Models in Medicine: A Scoping Review.” iScience, vol. 27, no. 5, May 2024, p. 109713. DOI.org (Crossref), https://doi.org/10.1016/j.isci.2024.109713.

9. Chumachenko, Dmytro. “Exploring Different Approaches to Epidemic Processes Simulation: Compartmental, Machine Learning, and Agent-Based Models.” Data-Centric Business and Applications, edited by Peter Štarchoň et al., vol. 208, Springer Nature Switzerland, 2024, pp. 27–54. DOI.org (Crossref), https://doi.org/10.1007/978-3-031-59131-0_2.
10. Hakkoum, Hajar, et al. “Interpretability in the Medical Field: A Systematic Mapping and Review Study.” Applied Soft Computing, vol. 117, Mar. 2022, p. 108391. DOI.org (Crossref), https://doi.org/10.1016/j.asoc.2021.108391.
11. Zhang, Yu, et al. “A Survey on Neural Network Interpretability.” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 5, no. 5, Oct. 2021, pp. 726–42. DOI.org (Crossref), https://doi.org/10.1109/TETCI.2021.3100641.
12. Nohara, Yasunobu, et al. “Explanation of Machine Learning Models Using Shapley Additive Explanation and Application for Real Data in Hospital.” Computer Methods and Programs in Biomedicine, vol. 214, Feb. 2022, p. 106584. DOI.org (Crossref), https://doi.org/10.1016/j.cmpb.2021.106584.
13. Parikh, Soham, et al. “Exploring Zero and Few-Shot Techniques for Intent Classification.” Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), Association for Computational Linguistics, 2023, pp. 744–51. DOI.org (Crossref), https://doi.org/10.18653/v1/2023.acl-industry.71.
14. Maksymenko, Daniil, et al. Controllability for English-Ukrainian Machine Translation by Using Style Transfer Techniques. 2023, pp. 1059–68. DOI.org (Crossref), https://doi.org/10.15439/2023F895.
15. Chen, Qian, et al. “Neural Natural Language Inference Models Enhanced with External Knowledge.” Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2018, pp. 2406–17. DOI.org (Crossref), https://doi.org/10.18653/v1/P18-1224.
16. Wu, Zhiyong, et al. “Self-Adaptive In-Context Learning: An Information Compression Perspective for In-Context Example Selection and Ordering.” Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2023, pp. 1423–36. DOI.org (Crossref), https://doi.org/10.18653/v1/2023.acl-long.79.
17. Wei, Sheng-Lun, et al. “Unveiling Selection Biases: Exploring Order and Token Sensitivity in Large Language Models.” Findings of the Association for Computational Linguistics ACL 2024, Association for Computational Linguistics, 2024, pp. 5598–621. DOI.org (Crossref), https://doi.org/10.18653/v1/2024.findings-acl.333.
18. Wang, Peiyi, et al. “Large Language Models Are Not Fair Evaluators.” Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2024, pp. 9440–50. DOI.org (Crossref), https://doi.org/10.18653/v1/2024.acl-long.511.
19. Singh, Chandan, et al. Rethinking Interpretability in the Era of Large Language Models. arXiv, 2024. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2402.01761.
20. Manias, Dimitrios Michael, et al. Semantic Routing for Enhanced Performance of LLM-Assisted Intent-Based 5G Core Network Management and Orchestration. arXiv, 2024. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2404.15869.
21. Wei, Alexander, et al. Jailbroken: How Does LLM Safety Training Fail? arXiv, 2023. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2307.02483.
22. Gemini Team, et al. Gemini 1.5: Unlocking Multimodal Understanding across Millions of Tokens of Context. arXiv, 2024. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2403.05530.
23. Rajpurkar, Pranav, et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv, 2016. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.1606.05250.
24. Luo, Weidi, et al. JailBreakV-28K: A Benchmark for Assessing the Robustness of MultiModal Large Language Models against Jailbreak Attacks. arXiv, 2024. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2404.03027.
25. OpenAI, et al. GPT-4 Technical Report. arXiv, 2023. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2303.08774.
26. Lee, Jinhyuk, et al. Gecko: Versatile Text Embeddings Distilled from Large Language Models. arXiv, 2024. DOI.org (Datacite), https://doi.org/10.48550/ARXIV.2403.20327.
27. “Choose an Embeddings Task Type | Generative AI on Vertex AI.” Google Cloud, https://cloud.google.com/vertex-ai/generative-ai/docs/embeddings/task-types. Accessed 22 Oct. 2024.
28. Molino, Piero, et al. “Parallax: Visualizing and Understanding the Semantics of Embedding Spaces via Algebraic Formulae.” Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, 2019, pp. 165–80. DOI.org (Crossref), https://doi.org/10.18653/v1/P19-3028.
29. Saichyshyna, Nataliia, et al. “Extension Multi30K: Multimodal Dataset for Integrated Vision and Language Research in Ukrainian.” Proceedings of the Second Ukrainian Natural Language Processing Workshop

(UNLP), Association for Computational Linguistics, 2023, pp. 54–61. DOI.org (Crossref), https://doi.org/10.18653/v1/2023.unlp-1.7.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.