PAG-LLM: Paraphrase and Aggregate with Large Language Models for Minimizing Intent Classification Errors

Vikas Yadav
University of Arizona, ServiceNow
Tucson, AZ USA
vikasy@email.arizona.edu

Zheng Tang
Samsung Research America
Mountain View, CA, USA
zheng.tang@samsung.com

Vijay Srinivasan
Samsung Research America
Mountain View, CA, USA
v.srinivasan@samsung.com

ABSTRACT

Large language models (LLM) have achieved remarkable success in natural language generation but lesser focus has been given to their applicability in key tasks such as intent-classification. We show that LLMs like LLaMa can achieve high performance on intent classification tasks with large number of classes but still make classification errors and worse, generate out-of-vocabulary intent labels. To address these critical issues, we introduce Paraphrase and AGgregate (PAG)-LLM approach wherein an LLM generates multiple paraphrases of the input query (parallel queries), performs intent classification for the original query and each paraphrase, and at the end aggregate all the predicted intent labels based on their confidence scores. We evaluate PAG-LLM on two large intent classification datasets: CLINC, and Banking and show 22.7% and 15.1% error reduction. We show that PAG-LLM is especially effective for hard examples where LLM is uncertain, and reduces the critical misclassification and hallucinated label generation errors.

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**; *Natural language generation.*

KEYWORDS

Large Langauge Model, Intent Classification, Paraphrasing, Generation confidence, Aggregation

ACM Reference Format:
Vikas Yadav, Zheng Tang, and Vijay Srinivasan. 2024. PAG-LLM: Paraphrase and Aggregate with Large Language Models for Minimizing Intent Classification Errors. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3657959

1 INTRODUCTION

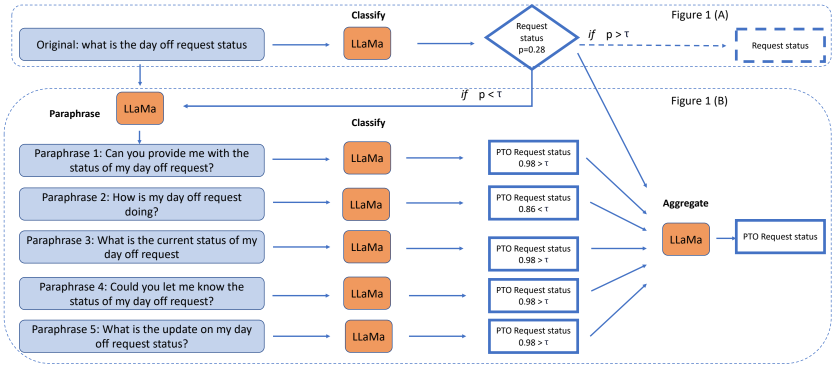
Recent progress on generative AI has had a transformative effect on the field of NLP and ML [2, 6, 21]. Large language models (LLM) have received more spotlight for generative tasks such as question answering, dialogue, summarization, etc [9, 17]. We argue that key NLP tasks such as intent classification is widely utilized in real-world dialogue systems and thus should also be given high emphasis when evaluating LLMs, considering their proven capability to solve a wide range of NLP tasks [9]. In this work, we focus on studying LLMs for large intent classification tasks with two intent classification datasets: CLINC [14] which has 150 classes and Banking [3] which has 77 classes. Intent classification is vital in many real-world NLP systems, where mapping an input query to an in-domain class or rejecting it (if out-of-domain) significantly affects system performance [20]. We demonstrate that LLMs, such as LLaMa, achieve strong performance on both datasets but still make miscalssification and worse, new unseen label generation errors. Thus, our work focuses on addressing such critical errors intent classification from LLMs.

In recent approaches like self-consistency [24], outputs of LLM from multiple reruns on the same input (with different temperatures or sampling techniques) can be aggregated for reducing LLM errors. In table 1, we show that such approaches are less applicable to straightforward classification tasks where the generated text is a single word or phrase representing the class label. In short response generations like single word for class label, tokens with highest generation probability should be selected as the final class label text. Similarly, while sampling techniques or temperature scaling [12, 22] are vital for promoting diversity, they may lesser applicable for fixing errors in classification tasks, where selecting tokens with the highest probabilities is necessary to predict the class label. Hence, as an alternative solution, we propose a (p)araphrasing and (ag)gregating approach (PAG) to fix LLM errors on intent classification task, where input query is paraphrased to perform intent classification on its multiple variations. Our approach is inspired by observations that often user queries are unclear which when rephrased, improve downstream systems [1]. PAG-LLM leverages the versatility of LLMs to perform three tasks: paraphrasing, intent classification, and aggregation. We first generate N paraphrases of the input query, then generate classification predictions for the original query and its N paraphrases and finally, aggregate the classification decisions for generating the final prediction, all using LLMs. Our key findings are as follows:

- We study LLMs on large intent classification task and show substantial error reduction through our proposed PAG-LLM approach. PAG-LLM reduces error by 22.7% on CLINC and 15.1% on Banking intent classification datasets. PAG-LLM also shows improvements in the out-of-domain intent classification setting [30] with 3.2% and 1.5% absolute F1 score improvement in CLINC and Banking respectively.
- We show that PAG-LLM can be selectively applied to low-confidence classification cases to potentially lower the inference cost (section 5). We also present analyses showing distribution of error

This work is licensed under a Creative Commons Attribution-ShareAlike International 4.0 License.

SIGIR '24, July 14–18, 2024, Washington, DC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-XXXX-X/18/06.
https://doi.org/10.1145/3626772.3657959


Classify
Original: what is the day off request status
LLaMa
Request status p=0.28
if p > τ
Figure 1 (A)
Request status
if p < τ
Figure 1 (B)
Paraphrase
LLaMa
Classify
Paraphrase 1: Can you provide me with the status of my day off request?
Paraphrase 2: How is my day off request doing?
Paraphrase 3: What is the current status of my day off request
Paraphrase 4: Could you let me know the status of my day off request?
Paraphrase 5: What is the update on my day off request status?
LLaMa
LLaMa
LLaMa
LLaMa
LLaMa
PTO Request status 0.98 > τ
PTO Request status 0.86 < τ
PTO Request status 0.98 > τ
PTO Request status 0.98 > τ
PTO Request status 0.98 > τ
Aggregate
LLaMa
PTO Request status

	Input	Prediction	Confidence
Original:	what is the day off request status	request_status	0.28
Paraphrase 1:	Can you provide me with the status of my day off request?	pto_request_status	0.98
Paraphrase 2:	How is my day off request doing?	pto_request_status	0.86
Paraphrase 3:	What is the current status of my day off request?	pto_request_status	0.98
Paraphrase 4:	Could you let me know the status of my day off request?	pto_request_status	0.98
Paraphrase 5:	What is the update on my day off request status?	pto_request_status	0.98
	Final Prediction: pto_request_status Aggregation EXAMPLE	Gold Label: pto_request_status	
Original:	what is the reason humans even exist	explain_meaning_of_life	0.11
Paraphrase 1:	What is the purpose of human existence?	meaning_of_life	0.32
Paraphrase 2:	Why do humans exist in the world?	The meaning of life	0.05
Paraphrase 3:	What is the cause of human existence?	*NULL*	0.07
Paraphrase 4:	What is the explanation for the existence of humans?	explain_life	0.21
Paraphrase 5:	What is the rationale behind the existence of human beings?	rational_existence	0.08
	Final Prediction: meaning_of_life Aggregation EXAMPLE	Gold Label: meaning_of_life	

Figure 1: Left figure depicting the flow process of PAG-LLM. On the left fig. 1(A), LLM classifies the original query and only if the classification confidence is lower than τ, original query is given to the LLM for generating paraphrases which are then again given to the LLM for classification. Finally, LLM aggregates the predicted class labels from paraphrases and the original query. In the right figure table, examples from CLINC are shown where LLM classifies incorrect label (top example) and out-of-vocabulory (OOV) class label (bottom example). In the top example, paraphrases generated by PAG-LLM enables correct classification decisions with high confidence scores. Thus, even simple majority voting aggregation leads to the correct class prediction. In the bottom example, only paraphrase2 from PAG-LLM enables correct classification while remaining paraphrases and the original query have OOV class labels. PAG-LLM aggregates texts of input, paraphrases, their labels and confidences to finally predict the correct class label.

corrections with PAG-LLM on misclassification and hallucinated label generations.

2 RELATED WORK

A few recent works have evaluated LLMs on classification tasks such as commonsenseQA, multiple-choice QA, and Boolean QA that have context documents and small number of classes [7, 21, 29] but lesser focus has been given on understanding and fixing their errors. Some of the recent approaches like self-consistency [24] and chain-of-thought (COT) [26] have shown to improve performance but mainly for arithmetic and symbolic reasoning tasks such as GSM8K, ArithmeticQA, StrategyQA [8]. As discussed previously, these approaches are intuitively not promising for (context-free) intent classification tasks because of single word (or phrase) generation for predicting class label [18]. Approaches like our PAG-LLM, where an input query is asked in multiple different ways via paraphrasing can be an effective solution in a wide variety of classification tasks. For example, PAG-LLM via paraphrasing is able to solve both misclassification error and OOV label generation error as shown in right fig. 1 example table.

Paraphrasing is a proven effective solution for evaluating and enhancing the robustness of NLP systems [11]. Previously, paraphrases were generated using a separate encoder-decoder module and then provided to a different neural end-task model, resulting in lower overall performance [5, 10, 11]. Contrary to these, PAG-LLM leverages LLMs' versatility to handle all paraphrasing, classification, and aggregation, showcasing its benefits through significant error reduction and performance improvements of the LLM classifier. Some of the recent work also utilize in-context few shot prompting for intent classification [15]. We compare to these previous works in table 1.

3 APPROACH

We focus and limit our experiments to only open LLMs[1] such as LLaMa [21]. PAG-LLM process flow is formulated in Algorithm 1 and a walkthrough example is shown in left fig. 1. The classifier LLM$_{\text{classify}}$ classifies the input query Q_i to class C_i with p_i confidence. If the classification confidence $p_i > \tau$, C_i is considered as the final label (depicted in left fig. 1(A)) else LLM$_{\text{paraphrase}}$ generates n paraphrases $[PQ_i^1...PQ_i^n]$ of Q_i. Then, LLM$_{\text{classify}}$ classifies each of the paraphrased queries predicting classes Cp_i. LLM$_{\text{aggregate}}$ then aggregates original and its paraphrased queries, their predicted classes along with their confidences to predict the final class C_i. This is depicted in left fig. 1(B)).

Algorithm 1 PAG-LLM algorithm

```
$Q_i \leftarrow$ Input query
$C_i, p_i \leftarrow LLM_{classify}(Q_i)$
**if** $p_i > \tau$ **then**
    return $C_i$                                              ▷ Figure 1(A)
**else**                                                      ▷ If confidence is $< \tau$, invoke PAG-LLM
    $[PQ_i^1...PQ_i^n] \leftarrow LLM_{paraphrase}(Q_i)$
    Predictions = $[C_i, p_i, Q_i]$
    **for** $PQ \leftarrow PQ_i^1$ to $PQ_i^n$ **do**
        $Cp_i, pp_i \leftarrow LLM_{classify}(PQ)$
        Predictions.insert($[Cp_i, pp_i, PQ]$)
    **end for**
    $C_i, p_i \leftarrow LLM_{Aggregate}$(Predictions)
    return $C_i$
**end if**
```

As shown in Figure 1, PAG uses LLM for paraphrasing, classification, and aggregating. We supervised finetune (SFT) LLaMa for each of these 3 tasks until convergence.

Classification: LLM$_{\text{classify}}$ is supervised finetuned (SFT) on training data of CLINC and Banking-50 for the classification task.

Paraphrasing: For paraphrasing, we finetune LLaMa on 419K ChatGPT paraphrasing dataset [23] which has 5 paraphrases of each query from Quora dataset, texts from SQuAD2.0[19] and CNN datasets[4]. LLM$_{\text{paraphrase}}$ generates 5 paraphrases[2] of the original

[1] Our organization prohibits usage of licensed and human feedback learning LLMs such as ChatGPT for privacy reasons.

[2] We limited our experiments to 5 paraphrases because our training data - ChatGPT paraphrasing dataset has only 5 paraphrase outputs for each query.

query (as shown in left fig. 1(B)) which are all fed to the $\text{LLM}_{\text{classify}}$ individually for classification.

Aggregating: In $\text{LLM}_{\text{aggregate}}$ finetuning for aggregation, we first generate predictions on original and paraphrased queries on the validation data. We simply concatenate the original query, all paraphrased queries, their prediction labels (along with their corresponding confidence scores) as input to train LLaMa to predict the final output label.

3.1 Need for finetuning

Intent classification is a critical first task that impacts downstream performance of real-world dialogue systems. Thus, having high accuracy and maximum plausible control for preserving privacy (for e.g., in healthcare, and legal domains) is critical which can be achieved by full SFT of open LLMs [25, 28]. A few shortcomings of open instruction tuned LLMs that necessitate SFT are as follows:

- Parsing generated output: OpenLLMs with zero or few-shot prompts often deviate from instructions, generating relevant but off-target responses [13]. We observed similar issues with open instruction finetuned (IFT) LLMs such as Vicuna and Alpaca where generated response had differnt types of text in addition to the intent class label.
- Intent label informativeness - We observed that intent label text were often confusing and overlapped with other intent classes. Showing examples of how queries are mapped to each intent class may not be feasible for LLM prompting input length in a large-intent classification task (>150 classes). In-context learning (ICL) is an effective solution but LLM performance is greatly dependent on the retrieval accuracy [15]. We show comparison to ICL in table 1.
- Prompt sensitivity - Prompt selection has shown to significantly affect openLLMs performances [16]. To avoid such variations, we fine-tune LLMs in all our experiments.

3.2 Datasets

We present our results on two large multi-class classification datasets: CLINC (with 150 intent classes) and Banking-50 (with 77 intent classes). In the Banking-50 dataset, 50% of the intent labels are masked and labeled as OOD (out-of-domain), while the remaining 50% (i.e., 38 labels) are retained as original. Following previous work [30], we train the LLM exclusively on the in-domain training data (2.1k samples for Banking-50 and 15k for CLINC). We showcase PAG evaluations in two settings:

- In-domain:(ID) is the straightforward multiclass classification task where input query is labeled into one of the predefined classes. Here, we only evaluate on in-domain test inputs (4.5K of CLINC and 1.5K of banking). Results are shown in table 1.
- With Out-of-domain:(ID+OOD) - We show evaluations on the full test dataset (5.5K queries of CLINC and 3.1K of banking) with both ID+OOD queries in table 2. In this setting, a query is labelled as OOD if it does not belong to the predefined ID classes. We consider a prediction to be OOD if the classification confidence is lower than a decision threshold or the generated label is out of label vocabulary list. Predictions above the threshold are mapped to the corresponding ID class. Techniques like PAG-LLM can be crucial for inputs that are closer to such important threshold.

3.3 Hyperparameters

We finetuned LLaMa-7B on 4 A100 GPUs using huggingface library [27] for 4 epochs, learning rate=1e-5, batchsize=4, and gradient accumulation step=4 on Banking-50 and CLINC training sets. For paraphrasing and aggregation, we SFT LLaMa-7B for 3 epochs.

4 RESULTS

We implemented three baselines with SFT LLaMa-7B based on self-consistency (SC) [24] (with top_k sampling and temperature) and training N different LLaMa-7B classifiers with different random seed ("Rand-seed" in table 1). We also show comparison with in-context learning based baselines from previous works in table 1. Our PAG-LLM method with LLaMa-7B are shown in row 5-8. Row 5 and 6 use simple voting for aggregation of the classification from original query and its 5 paraphrases i.e., LLM is utilized only for paraphrasing and intent classification while voting is used for aggregation. PAG-LLM in Row 7 and 8 utilizes LLM for aggregation also. In row 5 and 7, paraphrasing is done for all the input queries whereas in row 6 and 8, paraphrasing+aggregation is done only on low confidence input queries i.e., where LLM intent classification is uncertain. Results of in-domain evaluations are shown in table 1 and overall (ID+OOD) evaluations in table 2.

(1) ID-evaluations: SFT LLaMa-7B achieves high performance in ID classification, outperforming previous ICL based LLM classifiers by subtantial margins (row 1 VS P1-P4 rows). Performance of SFT LLaMa-7B is further improved by our full PAG approach as shown in row 7 and 8. Specifically, as shown in row 8 of table 1, PAG-LLM results in 22.7% and 15.1% error reduction in CLINC and Banking-50 datasets respectively. Aggregation using LLM is always better than voting emphasizing its usefulness within PAG (row 7,8 vs 5,6).

(2) Baseline comparisons: As expected, we observed slightly lower performance from aggregating predictions from top_k sampling (row 2) and temperature=0.9 (row 3) confirming self consistency like approaches may not work in such large (context-free) intent classification tasks. Our PAG-Aggregating predictions from 6 different LLM classifiers (trained with different random seeds) is slightly better than voting aggregation with PAG (row4 vs row6) but arguably having N different LLM classifiers is not practical. On the other hand, PAG-LLM can SFT a single LLM to paraphrase, classify and aggregate all predictions.

(3) Aggregating Lower Confidence: Paraphrasing and aggregating only lower confidence queries (row 6,8) results in higher performance compared to running PAG-LLM on all of the inference queries (row 5,7). This emphasizes the practical potential of PAG-LLM as paraphrasing and aggregation is needed only on 32% of the test inputs that have confidence below 98% in CLINC (thus resulting in 0.32*5 (paraphrased queries) + 1 (original query) = 2.6x number of inference runs).

(4) OOD-evaluations: Aggregation is effective for distinguishing ID from out-of-domain (OOD) queries [30], after which it can be classified into one of the ID classes. In table 2, PAG-LLM shows nearly 3.9% F1 improvement in OOD classification while also improving the ID F1 by 3.5% on CLINC. The improvements in Banking-50 are relatively smaller (1.5% F1) possibly due to its longer class labels compared to CLINC.

#	Exp	Aggregate method	Num runs	CLINC ID F1	CLINC Error Reduct.	Banking-50 ID F1	Banking-50 Error Reduct.
P1	ICL (5 shot) LLaMa-7B [15]	-	-	88.58		84.42	
P2	ICL (10 shot) LLaMa-7B [15]	-	-	91.73		87.63	
P3	ICL (5 shot) OPT-13B [15]	-	-	85.27		81.23	
P4	ICL (10 shot) OPT-13B [15]	-	-	89.24		85.65	
1	SFT LLaMa-7B	-	1x	96.29	-	94.04	-
2	Self-consistency (top_k)[24]	Vote	6x	96.15	+3.8	92.89	+19.3
3	Self-consistency (Temperature)[24]	Vote	6x	96.18	+3.0	93.62	+7.0
4	SFT LLaMa-7B (Rand-Seed)	Vote	6x	96.51	-5.9	94.41	-6.2
5	PAG (All queries)	Vote	6x	96.28	+0.2	93.52	+8.6
6	PAG (low conf < τ)	Vote	2.6x	96.32	-0.8	94.42	-6.4
7	PAG (All queries)	LLM	6x	97.05	-20.4	94.85	13.6
8	PAG (low conf < τ)	LLM	2.6x	**97.13**	**-22.7**	**94.94**	**-15.1**

Table 1: Performance on in-domain (ID) CLINC and Banking(50%) datasets. We use our same SFT LLaMa-7B from row 1 in self-consistency runs also (row2 and row3). Rand-seed denotes ensembling of 6 different SFT LLaMa trained with different random seeds. Vote denotes majority voting strategy to select the final label. For row 6 and 8, we tune the confidence threshold (denoted by τ=0.98, 0.90 for CLINC and Banking) on dev data and only the queries with lower classification confidences are given to PAG-LLM. Best numbers are shown in bold. P1, P2, P3, P4 represent 4 in-context learning (ICL) baselines from previous work.

Exp	Agg. Meth.	CLINC ID F1	CLINC OOD F1	CLINC All F1	CLINC Avg (ID+OOD)	Banking-50 ID F1	Banking-50 OOD F1	Banking-50 All F1	Banking-50 Avg (ID+OOD)
SFT LLaMa-7B	-	88.5	88.85	88.51	88.67	77.04	77.05	77.49	77.26
PAG (All queries)	Vote	91.51	92.56	91.51	92.03	77.14	77.49	77.15	77.31
PAG (conf < τ)	Vote	91.67	**92.58**	91.68	92.13	78.28	77.74	78.28	78.02
PAG (conf < τ)	LLM	**92.04**	92.54	**92.05**	**92.29**	**78.52**	**77.8**	**78.5**	**78.16**

Table 2: Performances on overall test datasets (ID and OOD inputs) of CLINC and Banking(50%). Notations are same as in table 1. All F1 denotes the macro F1 score for all ID + 1 OOD classes (i.e., 151 classes for CLINC and 38 for banking). Avg denotes average of ID and OOD F1 scores.

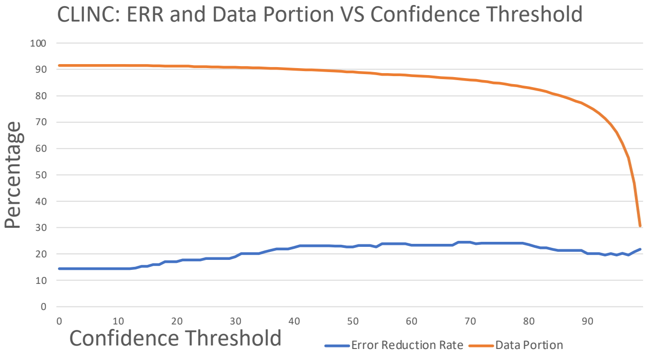


Figure 2: Plot showing portion of inference data and error rate reduction on CLINC with increasing classification threshold (τ).

5 ANALYSIS

Our work highlights high performance of SFT LLMs like LLaMa-7B for intent classification. PAG-LLM further improves the performance by substantially reducing errors of SFT LLaMa-7B model, especially for uncertain input queries with low classification confidence (row 8, table 1). We analyzed error rate reductions (ERR) and test data portion with varying confidence threshold (τ). As shown in fig. 2, ERR remains similar but data proportion drops substantially with higher τ (i.e., $\tau > 0.7$). With the validation data tuned $\tau = 98\%$, only 32% of the queries need to be paraphrased with PAG-LLM achieving 22.7% error reduction in CLINC, (row8 of table 1). We further analyzed the correction distribution of PAG-LLM. Out of 22.7% error correction on CLINC, 14.4% were for out-of-vocabulory (OOV) class label generation and remaining 8.3% for misclassification errors (shown in right fig. 1 example table). Similarly, out of 15.1%, 8.1% and 7.0% error reduction were for OOV and misclassification errors in Banking dataset.

Overall, our work presents a new method to reduce errors of LLM for intent classification. PAG-LLM approach utilizes LLM for each of the three components for intent classification, paraphrasing and aggregation leading to 22.7% and 15.1% error reduction on CLINC and Banking respectively. We show that PAG-LLM is effective in error reduction even for high performing tasks such as intent classification. Such findings are encouraging for future works to apply PAG-LLM to other key NLP tasks.

6 LIMITATIONS

A few limitations of our study are as follows:

- Effect of paraphrasing - We also experimented with randomly selected 210K paraphrasing data out of the 419K ChatGPT paraphrasing dataset [23]. We did not observe any substantial changes in our result findings but in-depth exploration of paraphrasing quality as future work could further enhance PAG-LLM approach.
- Extension to other LLMs - As SFT is a compute expensive process, we showcase our experiments with LLaMa-7B. PAG-LLM can be applied to any other open LLM and we will provide our github codes for easily extending to any LLM available in Huggingface model repositories.

REFERENCES

[1] Quentin Brabant, Gwénolé Lecorvé, and Lina M Rojas Barahona. 2022. CoQAR: Question Rewriting on CoQA. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 119–126.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[3] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. 38–45.

[4] Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2358–2367.

[5] Eunah Cho, He Xie, and William M Campbell. 2019. Paraphrase generation for semi-supervised learning in NLU. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. 45–54.

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[7] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2924–2936.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168* (2021).

[9] Nathan Lambert Nazneen Rajani Omar Sanseviero Lewis Tunstall Thomas Wolf Edward Beeching, Sheon Han. 2023. Open LLM Leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

[10] Tobias Falke, Markus Boese, Daniil Sorokin, Caglar Tirkaz, and Patrick Lehnen. 2020. Leveraging User Paraphrasing Behavior In Dialog Systems To Automatically Collect Annotations For Long-Tail Utterances. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*. 21–32.

[11] Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6065–6075.

[12] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. [n. d.]. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

[13] Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2022. A Good Prompt Is Worth Millions of Parameters: Low-resource Prompt-based Learning for Vision-Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2763–2775.

[14] Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 1311–1316.

[15] Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-Context Learning for Text Classification with Many Labels. In *GenBench: The first workshop on generalisation (benchmarking) in NLP*. 173.

[16] Hammond Pearce, Benjamin Tan, Baleegh Ahmad, Ramesh Karri, and Brendan Dolan-Gavitt. 2023. Examining zero-shot vulnerability repair with large language models. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2339–2356.

[17] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277* (2023).

[18] Silviu Pitis, Michael R Zhang, Andrew Wang, and Jimmy Ba. 2023. Boosted Prompt Ensembles for Large Language Models. *arXiv preprint arXiv:2304.05970* (2023).

[19] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 784–789.

[20] Yilin Shen, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. Enhancing the generalization for Intent Classification and Out-of-Domain Detection in SLU. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2443–2453.

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[22] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424* (2016).

[23] Maxim Kuznetsov Vladimir Vorobev. 2023. ChatGPT paraphrases dataset.

[24] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).

[25] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966* (2023).

[26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. [n. d.]. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*.

[27] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771* (2019).

[28] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792* (2023).

[29] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

[30] Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 2022. KNN-contrastive learning for out-of-domain intent classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5129–5141.