Improving Unsupervised Strict Zero-shot Intent Classification with Candidate Selection

Anonymous ACL submission

Abstract

Task-oriented dialogue systems allow users to interact through natural language with a variety of digital devices in order to accomplish some goal, within which intent classification is an integral component in ensuring the satisfaction of a user's request. Applications of Large Language Models (LLMs) in this domain can suffer from prohibitively high computation requirements and costs owing to the number of input tokens scaling with the number of intents. We propose a framework using *candidate selection*, aimed at refining a model's selection of candidate intents to reduce inference costs. We validate our approach through extensive evaluation on four commonly-used intent classification datasets and show that our candidate selection approach can improve zero-shot intent classification performance (between +2.08% to +14.67%) over naive zero-shot across a range of model parameters, while significantly reducing both the number of input tokens (up to 88% reduction) and inference time (up to 53% reduction). All the while accomplishing this without any additional fine-tuning.

1 Introduction

Intent classification (Larson et al., 2019) is an integral part of Task-Oriented Dialogue Systems (TODS) in determining the correct intent of the user through a given utterance. Combined with slot filling (Chen et al., 2015; Goo et al., 2018), it is critical in enabling a TODS to determine the appropriate functions to service the user's request. In recent years, the capabilities of Large Language Models (LLMs) in generalising to a large number of unseen tasks have improved significantly (Achiam et al., 2023; Dubey et al., 2024; Team et al., 2024b), with such models having already been shown to improve intent classification through generating synthetic training data (Cegin et al., 2023; Liu et al., 2024). However, supervised approaches using such models (Zhang et al., 2024; Gretz et al., 2023), while demonstrating impressive performance, can suffer from issues stemming from the cost associated with fine-tuning LLMs (Li et al., 2023a; Lin et al., 2024) that limit their practical application, compounded by the inherent limitations of training data requirement per intent and the necessity for further training if an intent is added. To mitigate the limitations mentioned above, recent work (Hong et al., 2024; Milios et al., 2023) has shown promising results for the ability of LLMs to perform *zero shot intent classification* in the absence of any task-specific fine-tuning. Yet, such approaches typically require the inclusion of the full list of supported intents within the model prompt, significantly increasing the number of input tokens and consequently cost —computationally and monetarily (Chen et al., 2023; Bang, 2023), which can limit the scaling of such methods to large numbers of intents (Larson et al., 2019).

In this work, we seek to explore approaches to address the aforementioned problems with the inference overhead associated with using LLMs[1]. We compare our approaches on a number of recently released models in a strict zero-shot, or *dataless*, intent classification setting in which we forego any model training. We perform extensive evaluation of our approaches in a number of different task settings with varying numbers of supported intents. Our contributions can be summarised as follows:

- We propose a framework using a dataless candidate selector to filter candidate intents for strict zero-shot intent classification.
- We show our approach can significantly improve classification performance over naive zero-shot on tested models (up to $+14.67\%$).
- We show our approach significantly reduces the number of input tokens (up to 88% reduc-

[1]All of our evaluation code and datasets will be made available at [GITHUB LINK]

tion) and average inference time[2] (up to 53% reduction).

- We evaluate our approach extensively on four commonly used Task-Oriented Dialogue datasets and report on the results.
- We provide analysis into the behaviours of the models on this task setting to encourage and guide future work in this domain.

2 Related Works

2.1 Zero-shot Intent Classification

Intent classification (IC) (Larson et al., 2019) refers to the task of assigning a given utterance to one of a list of supported intents within task-oriented dialogue systems to properly service the user's request. Zero-shot intent classification (0SHOT-IC) (Xia et al., 2018) focuses on systems that perform intent classification without training on labelled, task-specific data (Yin et al., 2019). (Fan et al., 2020) uses capsule networks (Liu et al., 2019) along with an outlier detector to leverage training on 'seen' classes to discriminate against unseen classes. (Zhang et al., 2022; Parikh et al., 2023; Kulkarni et al., 2024; Liu et al., 2024) all demonstrate the potential for learning conducted on synthetic examples in conjunction with supervised training to transfer to unseen contexts and domains. However, supervised approaches to 0SHOT-IC can suffer from issues stemming from their reliance on the quality and quantity of the training data (Yin et al., 2023; Xu et al., 2024).

Dataless classification (Chang et al., 2008; Song and Roth, 2014; Chen et al., 2015) is a stricter form of 0SHOT-IC defined by a total absence of training on *any* labelled data. This is typically achieved by mapping semantic representations of an utterance to a class label using their respective embedding distances (Chang et al., 2008). Recent approaches such as (Lamanov et al., 2022; Hu et al., 2024) have yielded promising results by leveraging embeddings of intent descriptions. However, (Hu et al., 2024) also identified an issue with the overlaps within the embedding space between different intent classes, particularly those pertaining to similar concepts (i.e. PLAYMUSIC and ADDTOPLAYLIST, AIR_FARE and GROUND_FARE), that was first noted by (Chang et al., 2008) in reference to the necessity of a large enough margin between semantic representations of classes for dataless classification.

2.2 LLM Zero-shot Intent Classification

Recent developments in LLMs have demonstrated the capabilities of such models in generalising to a wide range of tasks in a zero-shot manner (Achiam et al., 2023; Dubey et al., 2024; Team et al., 2024a), with a number of the aforementioned models having been used in generating synthetic examples for training (Cegin et al., 2023; Liu et al., 2024) and intent detection (Song et al., 2023). Supervised approaches using LLMs such as (Gretz et al., 2023; Zhang et al., 2024) have demonstrated the potential for such models to be fine-tuned to perform 0SHOT-IC. However, the cost of fine-tuning models of such size can prove prohibitive. LLM-based approaches also typically include the full list of intents and descriptions within the prompt (Hong et al., 2024; Milios et al., 2023), which can significantly increase the number of input tokens for a given utterance, forming a bottleneck towards scaling to a large number of intents (Larson et al., 2019).

2.3 LLM Re-ranking and Filtering

The problem of selecting the 'most-relevant' label from a large list of candidates based on some metric of quality has been studied extensively for re-ranking tasks in Information Retrieval (Nogueira and Cho, 2019; Azar et al., 2009). Recent works have explored the potential for LLMs to perform the task of ranking in multistage re-ranking frameworks, with most existing work in this domain focusing on fine-tuned models (Luo et al., 2024; Yue et al., 2023) or models accessible through commercially available APIs (Nouriinanloo and Lamothe, 2024; Rashid et al., 2024). These approaches can incur significant costs when scaling to tasks with a large number of labels. While approaches have explored the use of a filtering stage prior to re-ranking to reduce the number of inputs (Nouriinanloo and Lamothe, 2024; Rashid et al., 2024), these approaches typically make use of LLMs in the filtering mechanism, necessitating the inclusion of all labels within the prompt at least once within the framework.

[2]All inference times are extracted from experiments run on a single Nvidia Quadro RTX6000 GPU. In total, all experiments took an estimated total of 2400 GPU hours.

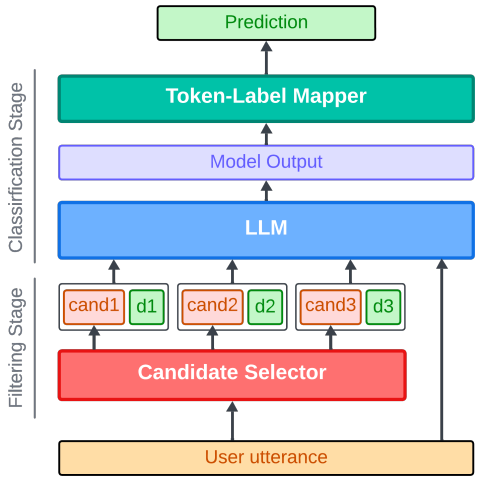

Figure 1: Illustration of our two-stage model architecture. For each model input, intent candidates (i.e. cand1) are added alongside their respective intent descriptions (i.e. d1) and the original user utterance into the prompt of the LLM.

```
Given the user said "UTTERANCE"

Please give the 'intent' that best reflect what the user is
saying/asking for, based on which of the following
intents has a description best matching the user's
utterance:

intent: INTENT1
description: DESCRIPTION1

...

Please give the intent name only, do not provide
reasoning.

The intent is:
```

Figure 2: An example prompt used to give the list of candidate intents and corresponding descriptions to the LLM.

3 Methodology

3.1 Problem Definition

For a given task-oriented dialogue system, we define $\mathcal{C}$ as the set of intents supported by the system. For each intent $c \in \mathcal{C}$, we define l_c as the description of the intent. We consider only the case where an utterance u is associated with a single intent c as previous works such as (Wan et al., 2024) have shown such approaches can be iteratively applied to tackle utterances with multiple labels and thresholding of relevance scores can be used to detect out-of-domain intents (Hou et al., 2021). We therefore leave further exploration in that domain for future work.

3.2 Our Approach

3.2.1 Candidate Selection

Previous work carried out by (Hu et al., 2024) produced a dataless intent classifier that leverages the cosine similarity $s(\cdot,\cdot)$ between the embeddings of a user utterance $\mathbf{h}(u)$ and an intent description $\mathbf{h}(l_c)$ to select the label $\hat{y} = \arg\max\ s(\mathbf{h}(u), \mathbf{h}(l_c))$ as the prediction. We instead sort the list of similarity scores to produce $\boldsymbol{s}$, where $\boldsymbol{s}_1$ is the highest similarity score.

In order to reduce the computational requirements posed by passing all intents within the input to our LLM (Nouriinanloo and Lamothe, 2024; Rashid et al., 2024), for each model prompt, we select the intents with the top-k highest similarity scores (Yang et al., 2012) as *candidate* intents. The list of candidate intents is then combined with their corresponding descriptions before being passed to the LLM to generate a prediction. Our approach is illustrated in Figure 1. As the focus of this work is on the use of a dataless classifier and the impact it has on zero-shot classification performance, we do not perform extensive prompt engineering and instead opt for a basic prompt template similar to previous works (Rashid et al., 2024). Figure 2 contains an example of our model prompt template. We provide further analysis of the effects of our candidate selector further on in Sections 6 and 7.

3.2.2 Token-Label Mapping

We follow previous work (Hong et al., 2024) in including instructions within the model prompt to return only the intent label. However, we observed that models did not always follow this particular instruction, we therefore implement a lightweight post-processing mechanism to tackle such cases. For a given utterance u consisting of tokens $x_1, \ldots, x_n$, we first extract a contiguous sequence of tokens $w = x_i, \ldots, x_j$ that forms the last 'word' in the sequence. We then map w to an intent class $\hat{l}$:

$$\hat{l} = \arg\min_{c \in \mathcal{C}}\ d_{\text{Lev}}(w, l_c) \quad (1)$$

where d_{Lev} is the Levenshtein distance between two strings and l_c is the intent label for class $c \in \mathcal{C}$. We provide further analysis of the effects of this mechanism in Section 6.2.

4 Experiments

4.1 Datasets

Following previous work of a similar nature and for the sake of comparison, we choose to evaluate our approach on four commonly used English TOD datasets: ATIS (Hemphill et al., 1990), SNIPS-NLU (Coucke et al., 2018), CLINC150 (Larson

Dataset	Uttr.	Intents	$\bar{u}$
ATIS (Hemphill et al., 1990)	5.8k	18	11.18
SNIPS-NLU (Coucke et al., 2018)	14.5k	7	8.97
CLINC150 (Larson et al., 2019)	22.5k	150	8.31
MASSIVE (FitzGerald et al., 2023)	16.5k	60	6.90
Total	59.3k	235	

Table 1: Dataset statistics for the four evaluation datasets used. $\bar{u}$ denotes the average sequence length for each dataset.

et al., 2019) and MASSIVE (FitzGerald et al., 2023). In each case, as our approach does not require fine-tuning, we use the full dataset as the evaluation data. A statistical breakdown for each dataset is shown in Table 1.

4.2 Baselines

Zero-shot Language Model (LM) Baseline We establish an LM-only baseline by implementing a basic zero-shot setup, providing each LM with the full list of intents and corresponding descriptions of each intent in a similar way to previous works (Gretz et al., 2023; Milios et al., 2023). We use the same prompt template previously outlined in Section 3.2.1 across all of our experiments. The results are referred to as 'LM Only' in Table 2.

Candidate Selector Baseline To evaluate the addition of the LM in conjunction with the candidate selector, we establish the results reported in (Hu et al., 2024) as our candidate selector baseline, referred to as 'Encoder only' in Section 5. We note the authors of (Hu et al., 2024) reported only macro-F1 for the MASSIVE dataset for the purpose of comparing against previous work. We, however, report both the accuracy and macro-F1 scores in our results (Section 5).

4.3 Models

Large Language Models We experiment with the following models as our LLM for producing an intent prediction given a set of candidates: **Llama-3.1-7B-Instruct** (Dubey et al., 2024), **gemma-2-9b-it** (Team et al., 2024b), **phi-3-medium-4k-instruct** (14B parameters) (Abdin et al., 2024) and **Mistral-7B-Instruct-v0.3** (Jiang et al., 2023). Our model selection was conducted based on a desire to capture a range of model performance and the availability of compute resources to us at the time of experimentation. It is by no means comprehensive and we invite future work to explore a wider range of models in application to this domain. All model weights were sourced from their respective repositories on Huggingface (Wolf et al., 2019) with default hyperparameters being used.

Model Quantization Due to a limitation in compute resources, we experiment with quantization of our selected models at 4-bit, 8-bit, 16-bit and 'full' (32-bit) precisions using the `bitsandbytes` library (Dettmers et al., 2024). Results of these experiments are elaborated upon in more detail in Section 6.4. We select the best-performing setup for each model by the consistency between the scores at each quantization precision and full precision.

Candidate Selector Models We experiment with using **BGE-large** (Xiao et al., 2024) and **GTE-large** (Li et al., 2023b) as our candidate selector models as both models have been shown to perform well in dataless contexts for intent classification (Hu et al., 2024). We analyse the impact of both models within our framework in Section 6.3.

5 Results

5.1 Metrics

Following on from previous works of a similar nature (Gritta et al., 2022; Hu et al., 2024), we report both Accuracy and Macro-F1 scores[3] for all models and datasets in our experiments. Where applicable, we report the average of Accuracy and Macro-F1 across all evaluation datasets as 'Overall' (Tables 2 and 4).

5.2 Zero-shot LM Baseline

The naive zero-shot baseline underperforms the encoder-only approach on Llama-3.1-8B-Instruct (-4.13% Overall), Phi-3-medium-4k-instruct (-3.81% Overall), Mistral-7B-Instruct (-13.41% Overall) models and outperforms the encoder-only approach on Gemma-2-9b-it (+5.03% Overall). On inspection of the model outputs, we note that the Phi-3 model and Mistral-7B models fail to produce a valid model prediction at much higher rates (9.61% and 9.71% respectively) compared to the Llama-3.1 (1.23%) and Gemma-2 (1.09%). In such instances, the model typically outputs a reasonable-looking intent label that is not of a valid intent given to the model. We attribute such outputs to hallucinations caused by the amount of intent labels and descriptions available to the models.

[3]Accuracy and macro-F1 are computed using the `scikit-learn` library (v1.3.2). Correlation is measured using the `numpy` library (v1.24.4).

	Model	Top-k	ATIS Acc.	ATIS F1	SNIPS Acc.	SNIPS F1	CLINC150 Acc.	CLINC150 F1	MASSIVE Acc.	MASSIVE F1	Overall
	Encoder only		69.57	52.51	92.81	92.33	81.95	81.09	65.49	65.76	75.19
All intents	Llama-3.1-8B-Instruct	*All*	67.53	38.13	71.94	71.78	86.64	86.03	73.87	72.60	71.06
	Gemma-2-9B-it		80.45	48.31	92.25	92.53	88.88	88.36	75.81	75.21	80.22
	Phi-3-medium-4k-instruct		75.94	48.15	92.45	92.70	60.89	57.71	73.26	69.97	71.38
	Mistral-7B-Instruct-v0.3		49.43	28.45	66.41	68.41	83.14	82.50	60.61	55.32	61.78
Candidate Selection (Ours)	Llama-3.1-8B-Instruct	k=3	76.08	52.65	79.23	78.60	87.89	87.34	72.91	72.19	75.86
		k=5	73.71	49.56	75.72	75.65	88.52	88.13	74.42	73.29	74.88
		k=10	72.23	45.11	72.38	72.27	88.88	88.46	75.00	74.07	73.55
	Gemma-2-9B-it	k=3	**86.48**	**57.60**	93.83	93.94	89.89	89.45	75.00	74.51	82.59
		k=5	85.69	57.34	92.74	93.04	**90.80**	**90.45**	77.07	**76.02**	**82.89**
		k=10	85.52	53.78	92.65	92.93	90.76	90.42	76.99	75.86	82.36
	Phi-3-medium-4k-instruct	k=3	75.97	56.17	**94.00**	**94.05**	89.01	88.65	74.63	72.88	80.67
		k=5	77.47	55.86	93.54	93.69	89.87	89.50	77.23	74.71	81.48
		k=10	78.31	54.82	92.85	93.08	90.01	89.72	**77.33**	75.10	81.40
	Mistral-7B-Instruct-v0.3	k=3	73.26	51.58	83.55	84.16	87.03	86.68	72.76	72.55	76.45
		k=5	65.66	45.32	76.22	77.95	87.35	87.08	74.14	73.48	73.40
		k=10	53.90	38.86	71.09	73.67	85.45	85.21	73.24	72.35	69.22

Table 2: Results of our approach on 4 intent classification datasets compared to LM performance. We report both Accuracy and Macro-F1 scores. **Overall** denotes the average of all metrics across all datasets.

5.3 Methods with Candidate Selection

Our approach using a dataless candidate selector on all selected models quantized at 8-bit precision (Section 6.4) yielded significant improvements across all tested models (Table 2) compared to the naive LLM baseline without candidate selection. For each model, the best-performing setup achieves: Llama-3.1-8B-Instruct (+4.80% Overall), Gemma-2-9b-it (+2.67% Overall), Phi-3-medium-4k-instruct (+10.10% Overall) and Mistral-7B-Instruct-v0.3 (+14.67% Overall). The increase in model performance can also be seen in the significant reduction in average model failure rate in producing a valid intent prediction at $k = 3$ compared to the naive zero-shot LM approach, which had access to the full list of intents and descriptions (1.96% vs 5.41%).

6 Ablations

6.1 Repeated Experiments

In order to observe the effect of randomisation on our results, we repeat our experiments for 4 independent runs and compute the average result. Due to the high number of experiments requiring a prohibitively high amount of time and compute resources, we choose to evaluate only the best-performing setups on the gemma-2-9b-it and llama-3.1-8b-instruct models to rerun across all selected intent classification datasets. We obtained a mean Overall score of 82.88 ± 0.01 for gemma-2-9b-it and 76.38 ± 0.08 for llama-3.1-8b-instruct. As $\sigma \ll 0.01\%$ in both instances, we conclude that our approach is consistent across random initialisations. This setup is used for all further experiments with the two models unless stated otherwise.

Model	Pred_{LM}	+Map	Δ
Llama-3.1-8B-Instruct	75.07	75.42	0.36
Gemma-2-9b-it	81.96	82.26	0.31
Phi-3-medium-4k-instruct	78.38	80.94	2.57
Mistral-7B-Instruct-v0.3	68.78	71.96	3.18

Table 3: Average model performance across all intent classification tasks, with and without our intent label mapping mechanism.

6.2 Effect of token-label mapping

We investigate the effect of our token-label mapping procedure on model performance by comparing results with and without our postprocessing step for all tested models at 4-bit and 8-bit precisions. Results are averaged across all setups per model and shown in Table 3, full results are shown in Appendix A. It can be observed that the inclusion of our mapping procedure improves model performance across all models tested, with

Model	Q	ATIS		SNIPS		CLINC150		MASSIVE		Overall
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	
Llama-3.1-8B-Instruct	4-bit	74.76	53.30	81.66	81.92	87.38	86.85	73.00	72.51	76.42
	8-bit	76.08	52.65	79.23	78.60	87.89	87.34	72.91	72.19	75.86
	full	78.48	53.00	79.89	79.34	88.05	87.52	72.93	72.14	76.42
gemma-2-9b-it	4-bit	86.32	55.78	93.08	93.24	89.59	89.13	74.78	74.12	82.00
	8-bit	**86.48**	57.60	93.83	93.94	89.89	89.45	**75.00**	**74.51**	82.59
	full	86.41	57.56	93.97	94.07	**89.94**	**89.49**	74.96	74.50	**82.61**
Phi-3-medium-4k-instruct	4-bit	76.04	**57.71**	93.74	93.80	88.41	88.09	73.92	72.06	80.47
	8-bit	75.97	56.17	94.00	94.05	89.01	88.65	74.63	72.88	80.67
	16-bit	76.30	57.06	**94.38**	**94.42**	89.03	88.64	74.60	72.89	80.92
Mistral-7B-Instruct-v0.3	4-bit	65.90	51.69	80.53	81.27	86.72	86.32	72.66	72.18	74.66
	8-bit	73.26	51.58	83.55	84.16	87.03	86.68	72.76	72.55	76.45
	full	72.68	51.27	82.43	83.02	87.18	86.82	72.65	72.39	76.05

Table 4: Model performance across various quantization precisions. Due to memory constraints, we report the 16-bit quantization of Phi-3-medium-4k-instruct and full (32-bit) for all other models.

greater improvements seen in the Phi-3-medium-4k-instruct (+2.57) and Mistral-7B-Instruct-v0.3 models (+3.18). We conduct a basic error attribution and broadly summarise the two main groups of errors that were eliminated by token-label mapping:

- *Output verbosity* - Our prompt included a specifier that the model should output only the proposed intent label. Nonetheless, it was observed across all models, particularly at 4-bit precision, that there were instances where the model would disregard this instruction and generate more text after generating the intent label.
- *Lexical errors* - Results from our experiments with Phi-3-medium-4k-instruct at 4-bit precision yielded a significant number of instances where the generated text would be similar to an intent label but would contain lexical errors, where a number of letters are incorrect. We note this error likely arises from quantization to 4-bit precision as it is not seen at 8-bit or 16-bit precisions.

In both instances, traditional regex post-processing would fail to correctly identify the model prediction, leading to misclassifications. Both of these issues are effectively eliminated with token-label mapping.

6.3 Effect of choice of candidate selector

We conduct a second set of experiments with a GTE-large model (Li et al., 2023b) using the same models and setups as in our experiments in Section 6.1. Table 5 shows the results for gemma-2-9b-it and llama-3.1-8b-instruct in setups using only the sentence embedding similarities as candidate selection metrics and setups using the additional paraphrasing and masking components proposed by (Hu et al., 2024). We observe a significant increase in performance in all setup permutations compared to LM-only. Figure 3a illustrates the success rates of the candidate selector models, showing the addition of paraphrasing and masking to yield a higher model success rate. Upon inspection of candidate selector predictions, we note that the addition of paraphrasing and masking improves the ranking of the correct label in 8.31% of examples on average. However, we also note that on average in 0.94% of examples, the correct label was no longer within the top-5 candidates. Further work should investigate improvements to the candidate selector to reduce the introduction of new errors. A detailed breakdown of results by dataset is available in Appendix B.

Model	BGE		GTE	
	E	+P+M	E	+P+M
Llama-3.1-8B-Instruct	4.88	5.41	4.57	5.55
Gemma-2-9b-it	1.98	2.61	1.65	2.14

Table 5: Changes in the overall score when candidate selector is used. E denotes setups using *encoder only* candidate selector and +P+M denotes setups with paraphrasing and masking components from (Hu et al., 2024).

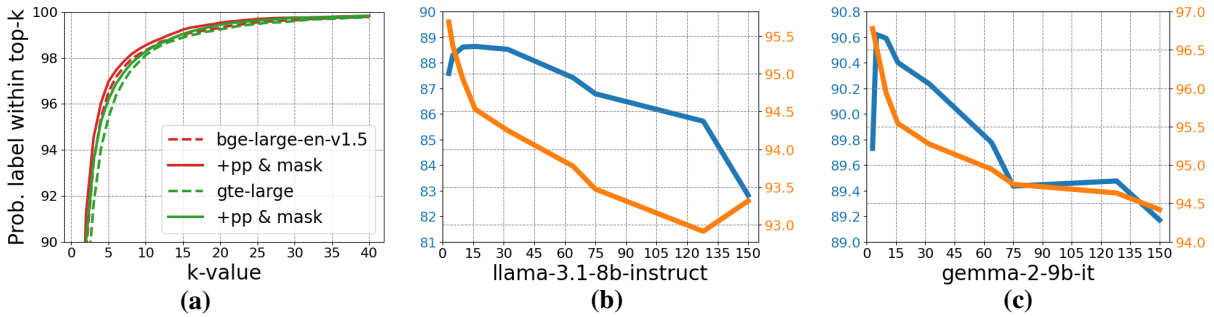

Figure 3: (**a**) Probability of the correct label being within top-k candidates given to the LM for different candidate selector setups. (**b**) and (**c**) **Blue**: Mean of Accuracy and Macro-F1 on the CLINC150 dataset. **Orange**: Model success rate when the label is within top-k candidates given to the LM. Note that the success rate decreases as k increases. Full results are shown in Appendix D.

6.4 Effect of model quantization precision on model performance

Table 4 shows the model performance for all selected models and datasets for $k = 3$ at various quantization precisions. We note that some models appear generally more robust across different quantization precisions (Llama-3.1 $\sigma = 0.32$, Gemma-2 $\sigma = 0.34$, Phi-3 $\sigma = 0.22$) while some yield more variant results (Mistral $\sigma = 0.94$). Table 6 shows the results of our investigation into comparing prediction behaviours across different precisions. Our results showed the average model failure rate in producing a valid prediction to be significantly higher at 4-bit precision than at 8-bit or full-precision (3.71% vs 1.96% vs 1.91%). We observed the largest difference in Phi-3-medium-4k-instruct (8.34% vs 0.69%), which began to produce misspellings of intent labels at 4-bit precision. We note that both Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3 had a lower failure rate at 4-bit precision than 8-bit (0.49 vs 0.57 and 4.47 vs 5.49), though for Mistral, it was closer to the failure rate of the full-precision model (5.49 vs 5.47). Table 6 also shows that all tested models showed greater correlation between predictions made at 8-bit precision (average Pearson's $r = 0.972$) than at 4-bit precision (average Pearson's $r = 0.921$), implying the 8-bit precision model to be more similar to the original, full-precision model. In consideration of this in addition to our compute resource constraints, and our early experiments showing an average of 54.56% reduction in inference time between the quantized and full-precision models (Table 10 in Appendix C), we opt to experiment with models quantized to 4-bit and 8-bit precision.

Model	LLM Failure (%)			Corr. Pred.	
	4-bit	8-bit	full	4-bit	8-bit
Llama-3.1-8B-Instruct	0.49	0.57	0.45	0.936	0.953
Gemma-2-9b-it	1.26	1.07	1.03	0.975	0.992
Phi-3-Medium-4k-Instruct	8.34	0.69	0.68	0.848	0.977
Mistral-7b-Instruct-v0.3	4.47	5.49	5.47	0.924	0.966
Mean	**3.71**	**1.96**	**1.91**	**0.921**	**0.972**

Table 6: Comparison of model predictions across quantization precisions. **LLM Failure** denotes the rate at which the LLM fails to produce the intent label without label-intent mapping. **Corr. Pred.** denotes the correlation of predictions between quantized models at lower precisions and the model at full/16-bit precision.

7 Analysis

7.1 Effect of k-value on model performance

As the choice of k has a direct impact on whether the correct label is presented to the LLM, we repeat our experiments on the CLINC150 dataset with an increasing number of candidates starting from $k = 3$ to a maximum of $k = 150$. Figures 3b and 3c show the performance of the models. Full results can be found in Appendix D. We observe on both tested models a peak in performance around $k = 10$ similar to those observed by (Hong et al., 2024) in their work on the same dataset, after which the model performance steadily decreases. On inspecting the model performance, given the correct label is provided to the LLM, we note that the success rate is continually decreasing for both models tested, suggesting the overall model performance is a balance between the correctness of the candidate selector in ensuring the correct label is provided as a candidate and the LLM selecting the correct label in turn.

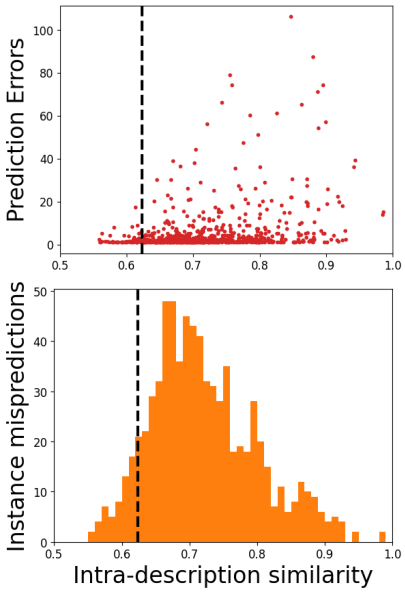

Figure 4: Distribution of prediction errors by the intra-description similarity of the ground-truth intent and the predicted intent on the CLINC150 dataset. **(Top)** The number of errors per relation **(Bottom)** The number of relations with erroneous predictions. Dashed line denotes the mean intra-description similarity across the entire dataset.

7.2 Effect of description similarity on model performance

We attempt to examine whether there exists any relation between classification errors in the model output and similarities between intent descriptions. Intuitively, this is the assumption that similar definitions of intents may lead to a greater degree of confusion, thus resulting in more prediction errors. Figure 4 illustrates the distribution of prediction errors by the intra-description similarity between the ground-truth intent and the predicted intent. We note a significant skewness of mispredictions towards higher intra-description similarities (Fisher-Pearson $G_1 = 0.798$). Computing the correlation between prediction errors and cosine similarity of description embeddings yielded a weak positive correlation (Pearson's $r = 0.213 \pm 0.015$). Conversely, much weaker correlation was obtained between the prediction errors and the intra-description Levenshtein distance (Pearson's $r = 0.07 \pm 0.01$), implying that errors arising from high description similarity or overlapping descriptions are more likely to be focused on the semantic similarities rather than lexical similarity. We also observe from error attribution that on average, mispredictions arising from the LLM accounted for 75-82% of classification errors across all of our experiments. We suggest future work to explore means to tackle prediction errors arising from such circumstances, such as through generating semantically diverse descriptions.

Model	**Inference time / it. (s)**		
	$k = 3$	$k = 16$	$k = 150$
Llama-3.1-8B-Instruct	1.69	2.59	4.29
Gemma-2-9b-it	5.05	5.61	11.26
Phi-3-Medium-4k-Instruct	2.68	3.89	5.64
Mistral-7B-Instruct-v0.3	3.09	3.60	5.71
% Reduction	*53%*	*38%*	-
Model	**Input Tokens / Prompt**		
	$k = 3$	$k = 16$	$k = 150$
Avg. Num. Tokens	128.55	312.21	1084.16
% Reduction	*88%*	*71%*	-

Table 7: Average inference time and number of input tokens per example for varying values of k. *% Reduction* indicates the mean percentage-decrease for lower values of k from $k = 150$.

7.3 Effect of candidate selection on model inference

A key conceptual benefit to filtering through candidate selection is a reduction in computational overhead and cost when applying to classification tasks with a high number of classes. We therefore analyse the benefits of our approach in these areas. We collect inference times and the number of input tokens per prompt for all selected models and datasets and analyse the average inference overhead, results are shown in Table 7. We observe from Table 7 that our approach significantly reduced the average inference time (up to 53% reduction) and the number of input tokens (up to 88% reduction). We note the latter in particular can significantly reduce the cost in using the growing suite of commercial LLMs-as-a-service solutions (Sun et al., 2022; Chen et al., 2023).

8 Conclusion

In this paper, we utilise a dataless candidate selector component and demonstrate its potential to improve model intent classification performance over encoder-only and naive zero-shot LLM approaches, while significantly reducing the number of input tokens and inference time, all without requiring any further fine-tuning of the models. We performed extensive experiments with a range of models and provide analysis into model behaviour and failure conditions to guide future work.

8.1 Limitations

Our work focuses on the potential for our proposed framework to leverage dataless candidate selection to mitigate computational overheads in a strict zero-shot intent classification setting. The utilisation of intent label descriptions in candidate selection creates the potential for such approaches to be applied to other such tasks that can be defined using label descriptions, such as emotion classification (Rashkin et al., 2019; Canales and Martínez-Barco, 2014) and genre prediction (Hoang, 2018) which we leave for future work to pursue. Additionally, we experimented with Levenshtein distance in our token-label mapping postprocessing procedure, future work could explore semantic distance metrics such as embedding similarity or WordNet-based methods.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yossi Azar, Iftah Gamzu, and Xiaoxin Yin. 2009. Multiple intents re-ranking. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 669–678, New York, NY, USA. Association for Computing Machinery.

Fu Bang. 2023. GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 212–218, Singapore. Association for Computational Linguistics.

Lea Canales and Patricio Martínez-Barco. 2014. Emotion detection from text: A survey. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43, Quito, Ecuador. Association for Computational Linguistics.

Jan Cegin, Jakub Simko, and Peter Brusilovsky. 2023. ChatGPT to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1889–1905, Singapore. Association for Computational Linguistics.

Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 830–835. AAAI Press.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. 2015. Dataless text classification with descriptive lda. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2224–2231. AAAI Press.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Lu Fan, Guangfeng Yan, Qimai Li, Han Liu, Xiaotong Zhang, Albert Y.S. Lam, and Xiao-Ming Wu. 2020. Unknown intent detection using Gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1050–1060, Online. Association for Computational Linguistics.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, Volume 2 (Short Papers), pages 753–757.

Shai Gretz, Alon Halfon, Ilya Shnayderman, Orith Toledo-Ronen, Artem Spector, Lena Dankin, Yannis Katsis, Ofir Arviv, Yoav Katz, Noam Slonim, and Liat Ein-Dor. 2023. Zero-shot topical text classification with LLMs - an experimental study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9647–9676, Singapore. Association for Computational Linguistics.

Milan Gritta, Ruoyu Hu, and Ignacio Iacobacci. 2022. CrossAligner & co: Zero-shot transfer methods for task-oriented cross-lingual natural language understanding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4048–4061, Dublin, Ireland. Association for Computational Linguistics.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The atis spoken language systems pilot corpus. *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania*.

Quan Hoang. 2018. Predicting movie genres based on plot summaries. *arXiv preprint arXiv:1801.04813*.

Taesuk Hong, Youbin Ahn, Dongkyu Lee, Joongbo Shin, Seungpil Won, Janghoon Han, Stanley Jungkyu Choi, and Jungyun Seo. 2024. Exploring the use of natural language descriptions of intents for large language models in zero-shot intent classification. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 458–465, Kyoto, Japan. Association for Computational Linguistics.

Yutai Hou, Yongkui Lai, Yushan Wu, Wanxiang Che, and Ting Liu. 2021. Few-shot learning for multi-label intent detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13036–13044.

Ruoyu Hu, Foaad Khosmood, and Abbas Edalat. 2024. Exploring description-augmented dataless intent classification. In *Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024)*, pages 13–36, Bangkok, Thailand. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Atharva Kulkarni, Bo-Hsiang Tseng, Joel Ruben Antony Moniz, Dhivya Piraviperumal, Hong Yu, and Shruti Bhargava. 2024. SynthDST: Synthetic data is all you need for few-shot dialog state tracking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1988–2001, St. Julian's, Malta. Association for Computational Linguistics.

Dmitry Lamanov, Pavel Burnyshev, Ekaterina Artemova, Valentin Malykh, Andrey Bout, and Irina Piontkovskaya. 2022. Template-based approach to zero-shot intent recognition. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 15–28, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 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)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Xuying Meng, Siqi Fan, Peng Han, Jing Li, Li Du, Bowen Qin, et al. 2023a. Flm-101b: An open llm and how to train it with $100kbudget$. *arXiv preprint arXiv:2309.03852*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 365–374, New York, NY, USA. Association for Computing Machinery.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y.S. Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. 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)*, pages 4799–4809, Hong Kong, China. Association for Computational Linguistics.

Han Liu, Siyang Zhao, Xiaotong Zhang, Feng Zhang, Wei Wang, Fenglong Ma, Hongyang Chen, Hong Yu, and Xianchao Zhang. 2024. Liberating seen classes: Boosting few-shot and zero-shot text classification via anchor generation and classification reframing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18644–18652.

Sichun Luo, Bowei He, Haohan Zhao, Wei Shao, Yanlin Qi, Yinya Huang, Aojun Zhou, Yuxuan Yao, Zongpeng Li, Yuanzhang Xiao, Mingjie Zhan, and Linqi Song. 2024. Recranker: Instruction tuning large language model as ranker for top-k recommendation. *ACM Trans. Inf. Syst.* Just Accepted.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 173–184, Singapore. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Baharan Nouriinanloo and Maxime Lamothe. 2024. Reranking step by step: Investigating pre-filtering for re-ranking with large language models. *arXiv preprint arXiv:2406.18740*, arXiv:2406.18740.

Soham Parikh, Mitul Tiwari, Prashil Tumbade, and Quaizar Vohra. 2023. Exploring zero and few-shot techniques for intent classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 744–751, Toronto, Canada. Association for Computational Linguistics.

Muhammad Rashid, Jannat Meem, Yue Dong, and Vagelis Hristidis. 2024. EcoRank: Budget-constrained text re-ranking using large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13049–13063, Bangkok, Thailand. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Xiaoshuai Song, Keqing He, Pei Wang, Guanting Dong, Yutao Mou, Jingang Wang, Yunsen Xian, Xunliang Cai, and Weiran Xu. 2023. Large language models meet open-world intent discovery and recognition: An evaluation of ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10291–10304, Singapore. Association for Computational Linguistics.

Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1579–1585. AAAI Press.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. 2024. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5836–5847, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. arxiv. *arXiv preprint arXiv:1910.03771*.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099, Brussels, Belgium. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Hanzi Xu, Muhao Chen, Lifu Huang, Slobodan Vucetic, and Wenpeng Yin. 2024. X-shot: A unified system to handle frequent, few-shot and zero-shot learning simultaneously in classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4652–4665, Bangkok, Thailand. Association for Computational Linguistics.

Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. 2012. On top-k recommendation using social networks. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, page 67–74, New York, NY, USA. Association for Computing Machinery.

Wenpeng Yin, Muhao Chen, Ben Zhou, Qiang Ning, Kai-Wei Chang, and Dan Roth. 2023. Indirectly supervised natural language processing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 32–40, Toronto, Canada. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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), pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. Llamarec: Two-stage recommendation using large language models for ranking. *arXiv preprint arXiv:2311.02089*, arXiv:2311.02089.

Feng Zhang, Wei Chen, Fei Ding, Meng Gao, Tengjiao Wang, Jiahui Yao, and Jiabin Zheng. 2024. From discrimination to generation: Low-resource intent detection with language model instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10167–10183, Bangkok, Thailand. Association for Computational Linguistics.

Yiwen Zhang, Caixia Yuan, Xiaojie Wang, Ziwei Bai, and Yongbin Liu. 2022. Learn to adapt for generalized zero-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 517–527, Dublin, Ireland. Association for Computational Linguistics.

A Results of token-label mapping

Table 8 contains the full results for model performance with and without token label mapping, used to compute the values for Table 3.

B Results of choice of candidate selector

Table 9 shows the full results for our experiments using different candidate selectors, from which we get the results in Table 5.

C Model inference times at full/16-bit precision

Table 10 shows the inference time of all tested models at full/16-bit quantization precision.

D Results of increasing k-value

Table 11 shows the full list of results from our experiments on increasing k-value using Gemma-2-9b-it and Llama-3.1-8B-Instruct that was used to produce Figures 3b and 3c.

Model	Q	k-value	ATIS		SNIPS		CLINC150		MASSIVE		Overall	
			Before	After	Before	After	Before	After	Before	After	Before	After
Llama-3.1-8B-Instruct	4-bit	k=3	63.97	64.03	81.77	81.79	86.78	87.11	72.56	72.75	76.27	76.42
		k=5	62.22	62.28	81.55	81.57	87.43	87.87	74.00	74.18	76.30	76.48
		k=10	61.18	61.39	76.95	76.96	88.29	88.71	74.30	74.41	75.18	75.37
	8-bit	k=3	63.50	64.37	78.86	78.91	87.50	87.62	72.32	72.55	75.54	75.86
		k=5	59.89	61.64	75.67	75.69	88.24	88.33	73.63	73.85	74.36	74.88
		k=10	55.86	58.67	72.31	72.32	88.49	88.67	74.41	74.53	72.77	73.55
Gemma-2-9b-it	4-bit	k=3	70.97	71.05	92.97	93.16	89.27	89.36	73.90	74.45	81.78	82.01
		k=5	69.83	70.30	90.93	91.50	90.07	90.15	75.67	75.99	81.62	81.98
		k=10	68.72	69.17	90.65	91.88	89.94	90.06	75.69	75.90	81.25	81.75
	8-bit	k=3	**71.73**	**72.04**	93.76	93.88	89.67	89.67	74.29	74.76	82.36	82.59
		k=5	71.21	71.52	92.74	92.89	**90.55**	**90.63**	76.23	**76.54**	**82.68**	**82.89**
		k=10	69.11	69.65	92.25	92.79	90.54	90.59	**76.27**	76.42	82.04	82.36
Phi-3-medium-4k-instruct	4-bit	k=3	64.74	66.88	84.26	93.77	74.82	88.25	67.64	72.99	72.87	80.47
		k=5	65.36	66.54	85.05	93.13	75.87	88.75	70.12	74.59	74.10	80.75
		k=10	63.79	64.48	92.79	92.97	89.43	89.87	75.88	76.21	80.47	80.88
	8-bit	k=3	65.93	66.07	**93.85**	**94.02**	88.72	88.83	73.42	73.75	80.48	80.67
		k=5	66.38	66.67	93.49	93.61	89.47	89.68	75.59	75.97	81.23	81.48
		k=10	66.30	66.57	92.79	92.97	89.44	89.87	75.88	76.21	81.10	81.40
Mistral-7B-Instruct-v0.3	4-bit	k=3	58.40	58.80	74.94	80.90	85.38	86.52	70.66	72.42	72.34	74.66
		k=5	50.42	50.56	68.80	73.95	86.03	86.98	72.53	74.06	69.44	71.39
		k=10	40.70	40.79	62.48	67.85	84.12	84.91	71.07	73.14	64.59	66.67
	8-bit	k=3	61.88	62.42	78.31	83.86	85.20	86.85	68.79	72.65	73.55	76.45
		k=5	55.34	55.49	71.35	77.09	85.40	87.22	67.61	73.81	69.92	73.40
		k=10	45.10	46.38	66.49	72.38	79.52	85.33	60.23	72.80	62.84	69.22

Table 8: Results of each tested model on all four evaluation datasets with and without token-label mapping. **Before** - Prediction made directly from model output. **After** - Prediction made using token-label mapping.

CS	LM	Setup	ATIS		SNIPS		CLINC150		MASSIVE		Overall
			Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	
Gemma-2-9b-it	BGE	Embedding Only	85.14	54.69	92.68	92.99	90.62	90.27	76.62	75.02	82.25
		+ Parap. + Mask.	85.69	57.34	92.74	93.04	90.80	90.45	77.07	76.02	82.89
	GTE	Embedding Only	86.32	54.84	93.09	93.34	90.01	89.58	74.86	73.39	81.93
		+ Parap. + Mask.	86.25	54.99	93.00	93.24	90.33	89.93	76.55	75.08	82.42
Llama-3.1-8B-Instruct	BGE	Embedding Only	74.43	48.91	80.30	81.32	88.10	87.49	73.83	73.13	75.94
		+ Parap. + Mask.	73.45	51.11	81.04	82.10	88.10	87.64	74.44	73.93	76.48
	GTE	Embedding Only	76.27	48.07	80.64	81.47	87.21	86.56	72.95	71.92	75.64
		+ Parap. + Mask.	76.03	49.39	82.71	83.44	87.69	87.19	73.71	72.74	76.61

Table 9: Performance when using different candidate selectors within our framework. BGE - bge-large-en-v1.5, GTE - gte-large. Paraphrasing and Masking refers to the additional components proposed by (Hu et al., 2024) to improve the candidate selector.

Model ($k = 3$)	Inference Time / it. (s)
Llama-3.1-8B-Instruct	5.76
Gemma-2-9b-it	13.82
Phi-3-medium-4k-instruct*	4.05
Mistral-7B-Instruct-v0.3	3.90
Average	6.88

Table 10: Inference times for each model at the highest quantization precision tested. *Due to GPU memory limits, we test Phi-3-medium-4k-instruct at 16-bit precision.

k-value	Gemma-2-9b-it			Llama-3.1-8B-Instruct		
	Acc.	F1	(LM %)	Acc.	F1	(LM %)
k=3	89.93	89.53	**96.78**	87.86	87.30	**95.69**
k=5	**90.80**	**90.44**	96.54	88.49	88.09	95.37
k=10	90.76	90.42	95.93	88.82	88.40	94.93
k=16	90.57	90.23	95.54	**88.85**	**88.42**	94.53
k=32	90.40	90.07	95.27	88.73	88.31	94.25
k=64	89.95	89.60	94.94	87.69	87.14	93.78
k=75	89.63	89.24	94.75	87.08	86.49	93.47
k=128	89.66	89.29	94.64	85.98	85.44	92.91
k=150	89.36	88.98	94.42	83.14	82.50	93.32

Table 11: Results of tested models on the CLINC150 dataset for varying k-values. (LM %) denotes the LM's success rate given the correct intent label is within the list of candidates.