Selective In-Context Data Augmentation for Intent Detection using Pointwise V-Information

Yen-Ting Lin∗ Alexandros Papangelis† Seokhwan Kim† Sungjin Lee† Devamanyu Hazarika† Mahdi Namazifar† Di Jin† Yang Liu† Dilek Hakkani-Tur†

∗National Taiwan University
†Amazon Alexa AI

Abstract

This work focuses on in-context data augmentation for intent detection. Having found that augmentation via in-context prompting of large pre-trained language models (PLMs) alone does not improve performance, we introduce a novel approach based on PLMs and pointwise V-information (PVI), a metric that can measure the usefulness of a datapoint for training a model. Our method first fine-tunes a PLM on a small seed of training data and then synthesizes new datapoints – utterances that correspond to given intents. It then employs intent-aware filtering, based on PVI, to remove datapoints that are not helpful to the downstream intent classifier. Our method is thus able to leverage the expressive power of large language models to produce diverse training data. Empirical results demonstrate that our method can produce synthetic training data that achieve state-of-the-art performance on three challenging intent detection datasets under few-shot settings (1.28% absolute improvement in 5-shot and 1.18% absolute in 10-shot, on average) and perform on par with the state-of-the-art in full-shot settings (within 0.01% absolute, on average).

1 Introduction

Intent detection, defined as the identification of a user’s intent given an utterance, is a fundamental element in task-oriented dialogue systems, usually occurring within the Natural Language Understanding (NLU) component. One of the practical challenges of training and deploying NLU modules is data scarcity, due to various reasons, such as under-represented languages, privacy and ethical concerns, or simply the cost of collecting and annotating sufficiently large amounts of data for new intents. Consequently, accurately identifying intents in limited-resource scenarios has drawn attention from the community (Papangelis et al., 2021; Mehri and Eric, 2021; Zhang et al., 2021b, for example).

There are three main families of approaches that address the challenge of limited data for intent detection: data augmentation (Peng et al., 2021; Li et al., 2021), focusing on generating high-quality synthetic training and evaluation data; few-shot learning (Zhang et al., 2020, 2021b), focusing on creating learning algorithms that can cope with limited amounts of data; and transfer learning (Namazifar et al., 2021), focusing on learning algorithms that can generalize across domains (therefore not requiring in-domain data). In this work, we follow the data augmentation approach, which is a general method that attempts to augment a human-authored dataset with a large set of synthetically-generated instances. Most recent work has suggested using Pre-trained Language Models (PLMs) for data augmentations under various setups, e.g., (Peng et al., 2021), showing great improvements in performance. However, simply generating a large number of synthetic data points is not enough; we need to consider the quality of each data point, i.e., how beneficial it would be to the model’s performance if that synthetic data point is added to the training set. This is an important issue since the model might learn to overfit to synthetic datapoints (which may be low quality, represent specific use cases, etc.) and thus under-perform on real data.

In this work, we propose to apply Pointwise V-Information (PVI) (Ethayarajh et al., 2022) for data augmentation, in a way that leverages a PLM to generate synthetic examples that are relevant and beneficial for training the downstream model, which in our case is an intent classifier. Our contributions are as follows:

- We propose a novel filtering method based on PVI (Ethayarajh et al., 2022) to filter out examples that are not relevant or helpful to the desired intent.
- We conduct experiments on three challenging intent detection datasets and show that our
method achieves state-of-the-art performance.

• We conduct an in-depth study and present a comprehensive analysis of the factors that influence performance, including ablation studies and comparisons with alternative methods.

The rest of the paper is organized as follows: In Section 2 we present relevant work and in Section 3 we introduce our method. In sections 4 and 5 we discuss training details, experiments, and results. In section 6, we present our analysis and discuss alternative approaches we investigated. In section 7 we conclude, and in the following sections we discuss limitations and ethical considerations.

2 Related Work

Intent Detection Intent detection is the task of identifying the user’s intent by mapping the user’s natural language utterance into one of several pre-defined classes (Hemphill et al., 1990; Coucke et al., 2018). It is a critical component in the pipeline of task-oriented dialogue systems, as it is used to determine the user’s goal and to trigger an appropriate system action (Raux et al., 2005; Young et al., 2013). Several datasets have been proposed to evaluate the performance of intent detection models (Casanueva et al., 2020; Liu et al., 2019a; Larson et al., 2019, for some recent examples). With the availability of such datasets, intent detection has been extensively studied in the literature. Recently, pre-trained language models (e.g., BERT (Devlin et al., 2019)) have been shown to be effective in intent detection (Bunk et al., 2020; Zhang et al., 2020, 2021a,b; Mehri and Eric, 2021).

Data Augmentation Data augmentation is a widely-used technique to address the problem of data scarcity. Paraphrasing the data is one of the ways frequently used for augmentation and can produce more diverse synthetic text with different word choices and sentence structures while preserving the meaning of the original text. Paraphrasing methods have been shown to be effective in many natural language processing tasks (Gupta et al., 2018; Edunov et al., 2018; Iyyer et al., 2018; Wei and Zou, 2019; Cai et al., 2020; Okur et al., 2022; Panda et al., 2021; Jolly et al., 2020). However, such methods often fail to generate more challenging and semantically diverse sentences that are important for the robustness of the downstream models.

Recently, conditional generation – using a PLM to produce text conditioned on some label – has become the dominant paradigm of data augmentation (Bowman et al., 2016; Kumar et al., 2019; Anaby-Tavor et al., 2020; Kumar et al., 2020; Yang et al., 2020a; Lee et al., 2021). This is usually achieved by fine-tuning a language model to produce the original text given the label.

In the field of intent detection, previous work has proposed using data augmentation techniques to generate synthetic training data (Sahu et al., 2022; Papangelis et al., 2021). Sahu et al. (2022) also used PLMs to generate augmented examples, but they require human effort for labeling. This is a challenging task since it is expensive to annotate large amounts of data.

Our approach involves data valuation, similar to the concepts of Ghorbani and Zou (2019); Mindermann et al. (2022). However, our approach differs from such previous work in two key ways. First, Ghorbani and Zou (2019) only evaluated the quality of the training set after training them, whereas we evaluate the synthetic examples before training the task model. Second, Mindermann et al. (2022) selected points that minimize the loss on a holdout set, whereas we select synthetic examples that are reasonably challenging to the task model. Our approach aims to address the problem of data scarcity by evaluating the synthetic examples generated by PLMs and selecting the most valuable examples to augment the training data.

In-context Learning Large language models such as GPT-3 (Brown et al., 2020) and OPT (Zhang et al., 2022) have shown to be able to perform many natural language processing tasks with in-context learning. In this paradigm, the model is provided with a few exemplars based on which it performs the respective task.

In-context learning is a promising solution for few-shot learning. Because of the effectiveness in few-shot performance, in-context learning has been applied to a wide range of NLP tasks. For dialogue tasks, in-context learning has been applied to intent classification (Yu et al., 2021), semantic parsing (Shin and Durme, 2022), and dialogue state tracking (Hu et al., 2022).

However, PLMs require a large amount of computational resources and the limitation on input length restricts the application of PLMs to intent detection tasks with large numbers of intents (e.g., 150 intents in CLINC (Larson et al., 2019)), where
we cannot fit examples for each intent in the input. One solution would be to call the model multiple times, each time with a subset of the possible intents. This would lead to increased inference time and may also impact performance. Consequently, Yoo et al. (2021); Sahu et al. (2022) leveraged in-context learning and PLMs to generate synthetic examples for intent detection, instead of directly deploying the PLM. However, they did not consider the quality of the generated examples, which may lead to the model overfitting on examples that are not relevant to the desired intent.

3 In-Context Data Augmentation

In the following section, we describe our proposed two-stage method for data augmentation, which we refer to as In-Context Data Augmentation (ICDA). The overall procedure is summarized in Algorithm 1. We apply ICDA to the task of few-shot intent detection, which involves classifying a user utterance \( x \) into an intent label \( y \in Y \). ICDA aims to generate synthetic examples \( x' \) such that they would belong to a given intent \( y \).

3.1 Synthesizing Examples

The core idea is to use a large pre-trained language model such as GPT-3 (Brown et al., 2020) or OPT (Zhang et al., 2022) to generate synthetic data in the context of the training set. In particular, for each intent class, we create a natural language text (prompt) that contains the intent class name, a set of real training examples under the same intent class, and an incomplete example. For instance, the prompt for the intent class \textit{refund not showing up} is shown in Figure 1. We feed the prompt to the language model and obtain a set of synthetic examples as outputs. In this work, we use OPT-66B (Zhang et al., 2022) as the language model to generate a set of examples for each intent class. We adopt typical decoding with \( \tau = 0.9 \) (Meister et al., 2022) and set repetition penalty to 1.1 following Keskar et al. (2019) to generate the synthetic examples.\(^1\) Due to the fine-grained nature of intents, and the sampling-based generation aiming to produce a set of diverse datapoints, we expect some of the generated utterances to not match the given intent.

Note that our method leverages PLMs in a way that is orthogonal to the intent detection model. Unlike other methods that use the same model to directly predict the intent class of a user utterance, we use a PLM to generate synthetic training instances. These instances are then used to augment the actual training data and train a smaller intent detection model. This approach leverages the power of PLMs while preserving the independence of the intent detection model design.

3.2 PVI Filtering

As mentioned above, given the stochastic nature of synthetic data generation, we expect some of the synthetic utterances not to match the given intent. To address this phenomenon, we filter generated instances and retain only those that are relevant and helpful to the desired intent classes.

Specifically, we apply Pointwise V-Information (Ethayarajh et al., 2022) - an idea originally suggested for understanding how difficult a dataset is - as a filter to discard unhelpful datapoints. PVI of an utterance \( x \) with respect to its corresponding intent class, \( y \), is defined as:

\[
PVI(x \rightarrow y) = - \log_2 g^* [\emptyset](y) + \log_2 g'(x)(y)
\]

where, in this work, \( g' \) and \( g^* \) are the intent detection models finetuned with and without the input \( x \), respectively. \( \emptyset \) is a special token that is used to indicate the absence of an input utterance.

Intuitively, PVI measures the amount of information that the input \( x \) provides to the intent detection

\(^1\)Implementation details are available from https://huggingface.co/docs/transformers/main_classes/text_generation
Algorithm 1: In-Context Data Augmentation with PVI Filtering

**Input:** Task Model $V$, Language Model $PLM$, Data Multiplier $m$, PVI Threshold Function $\epsilon$

**Output:** Task Model $g$

**Data:** Seed Data $D_{\text{train}} = \{(\text{input } x_i, \text{gold label } y_i)\}_{i=1}^{n}$

1. $g' \leftarrow$ Finetune $V$ on $D_{\text{train}}$
2. $\emptyset \leftarrow$ empty string
3. $g^* \leftarrow$ Finetune $V$ on $\{((\emptyset, y_i)| (x_i, y_i) \in D_{\text{train}}\}$
4. $D_{\text{synthetic}} \leftarrow$ Prompt($PLM, D_{\text{train}}$)
5. for $(x_i, y_i) \in D_{\text{synthetic}}$ do
6. \[ \text{PVI}(x_i \rightarrow y_i) \leftarrow -\log_2 g^*[\emptyset](y_i) + \log_2 g'[x_i](y_i) \]
7. $D_{\text{synthetic}} \leftarrow \{(x_i, y_i)|(x_i, y_i) \in D_{\text{synthetic}} \& \text{PVI}(x_i \rightarrow y_i) > \epsilon(y_i)\}$
8. $g \leftarrow$ Finetune $V$ on $D_{\text{train}} \cup D_{\text{synthetic}}$

---

Table 1: To assess the impact of the synthetic data size on performance, we experiment with several data multipliers (synthetic data size = source data size $\times$ mult.).

<table>
<thead>
<tr>
<th>Full-shot mult.</th>
<th>Few-shot mult.</th>
</tr>
</thead>
<tbody>
<tr>
<td>XS</td>
<td>1x</td>
</tr>
<tr>
<td>S</td>
<td>4x</td>
</tr>
<tr>
<td>M</td>
<td>16x</td>
</tr>
<tr>
<td>L</td>
<td>64x</td>
</tr>
<tr>
<td>XL</td>
<td>128x</td>
</tr>
</tbody>
</table>

---

4.2 Training

In our experiments, we use RoBERTa-LARGE (Liu et al., 2019b) as the intent detection model $V$ in Algorithm 1. We use OPT-66B² (Zhang et al., 2022) as the language model $PLM$ to generate synthetic examples and set the data multiplier $m$ to be 128³. We set the PVI threshold function $\epsilon$ to be the average PVI under each intent class in the validation set, where the PVI is computed using the same models as in Algorithm 1. We train RoBERTa-LARGE for 40 epochs with a batch size of 16, a learning rate of $1e-5$, and the AdamW optimizer (Loshchilov and Hutter, 2019). We use the HuggingFace Transformers library (Wolf et al., 2020) for all experiments.

4.3 Baseline Models

We compare our proposed method with the following baselines:

- **RoBERTa-BASE + Classifier** is a baseline that uses RoBERTa-BASE (Liu et al., 2019b) with a linear classifier on top (Zhang et al., 2020).
- **USE** is a universal sentence encoder pre-trained on 16 languages supporting multiple down-stream tasks (Yang et al., 2020b).
- **CONVERT** is an intent detection model finetuned from dual encoder models, which is pre-trained on (input, response) pairs from Reddit (Henderson et al., 2020).
- **CONVBERT** fine-tunes BERT on a large open-domain dialogue corpus with 700 million conversations (Mehri et al., 2020).
- **CONVBERT + Combined** is an intent detection model based on CONVBERT, with example-driven training based on similarity matching and observers for transformer attentions. It also conducts task-adaptive self-supervised learning with masked language modeling (MLM) on the intent detection

²We used p3dn.24xlarge AWS EC2 instances for our experiments.
³This means that we generate $m$ times the available training data, e.g. $(5 \times 77) \times m$ in the 5-shot BANKING case.
datasets. Here, “Combined” represents the best MLM+Example+Observers setting in the referenced paper (Mehri and Eric, 2021).

**DNNC** (Discriminative Nearest-Neighbor Classification) is a discriminative nearest-neighbor model, which finds the best-matched example from the training set through similarity matching. The model conducts data augmentation during training and boosts performance by pre-training on three natural language inference tasks (Zhang et al., 2020).

**CPFT** (Contrastive Pre-training and Fine-Tuning) is the current state-of-the-art in few-shot intent detection on the selected datasets. It is pre-trained on multiple intent detection datasets in a self-supervised contrastive manner and then fine-tuned with supervised contrastive learning (Zhang et al., 2021b).

### 5 Experimental Results

We conduct experiments on three benchmark datasets to validate the effectiveness of our proposed method. We first use OPT-66B to generate augmentation examples and then apply our method to enhance a RoBERTa-Large model trained on three datasets. We repeat all experiments with 5 random seeds and report the average performance in Full-shot and Few-shot settings. To investigate the effect of the synthetic data size, we experiment with a variety of multipliers (see Table 1 for notations). Results are shown in Table 2.

**Full-shot settings.** In this setting, we use the entire training set for each domain. The proposed method achieves the best performance on BANKING and comparable results on HWU and CLINC. In particular, on BANKING, we improve the CONVBERT + Combined baseline (Mehri and Eric, 2021) by 0.59% (absolute) and the RoBERTa-Large baseline by 0.72% (absolute). Compared with the CONVBERT + Combined, which is pre-trained on intent detection datasets in a self-supervised fashion and adds examples-driven training and specific model architectural design, our method achieves similar results with much simpler model design. Furthermore, our method is orthogonal to model architectures and can be integrated with any other approach for further improvement.

Moreover, state-of-the-art performance on BANKING with the proposed method and RoBERTa-Large shows that our method is capable of generating high-quality augmentation examples to enhance the RoBERTa-Large model on the most fine-grained intent detection task.

**Few-shot settings.** In this setting we only use a small number of instances (datapoints) per class. We evaluate our method in both 5-shot and 10-shot settings and compare it with several strong baselines. Our proposed method outperforms all baselines on all datasets in both 5-shot and 10-shot settings. ICDA-M achieves the best performance in 5-shot settings on BANKING dataset and ICDA-XL achieves the best performance on HWU and CLINC datasets in 5-shot settings and on all datasets in 10-shot settings. All configurations of our method significantly improve the performance of a RoBERTa-Large model trained on any of the three datasets. Compared with CPFT (Zhang et al., 2021b), which utilizes contrastive learning for few-shot intent detection with extra data, our method achieves better performance without any additional human-annotated data. This showcases the advantage of our method for few-shot intent detection.

We also observe that our method consistently improves the performance of the baseline model as the number of synthetic datapoints increases from XS to XL. This indicates that the generated instances from our method can gradually cover more and more information of real instances and are capable of providing more useful information for model training.

### 6 Analysis and Discussion

In this section, we analyze the performance of ICDA and other approaches we tried. We first identify several factors that affect performance, and then present evidence that ICDA works by transferring knowledge from the pretrained generator to the task model. We then discuss a data-relabelling experiment and an experiment using uncertainty measures or data cartography (Swayamdipta et al., 2020) as filters.

#### 6.1 Factors that Affect ICDA Performance

**ICDA is effective at various training sizes.** Throughout this work, we conduct experiments with different seed data sizes\(^4\) to study the effect of
We believe that the success of ICDA is because PVI filtering discards mislabeled examples. To verify this hypothesis, we observe that our proposed method consistently improves the accuracy of the downstream model in all training sizes. Also, as the training size decreases, we see that the ICDA improvement increases significantly. For example, on BANKING, the improvement goes from 42% in the full shot setting to 5.02% as the training size decreases to 5-shot. This indicates that ICDA is more effective when we have few training data available.

**PVI filtering threshold.** To study the effect of the threshold function $\epsilon$, we conduct experiments with two different threshold functions: *Global*, and *Per-Intent*. *Global* means that the PVI threshold is the same for all intent classes, which is the average PVI value in the validation set. *Per-Intent* means that the PVI threshold is different for each intent class, which is the average PVI value under each intent class in the validation set. As a sanity check, we also conduct experiments using synthetic instances with PVI values lower than the threshold (*Low PVI*) as opposed to the normal (*High PVI*) instances.

We show the results in Table 3 (bottom half), where we see that *Per-Intent High PVI* filtering performs the best. Compared to using all synthetic training data without filtering (referred to as *All*), we see that *High PVI* filtering in general helps in improving accuracy. In BANKING, for example, when PVI filtering is applied with *Per-Intent High PVI*, the accuracy is 88.64% with 10-shot training size, which is significantly better than the result without PVI filtering (84.19%) – the same holds for the other two datasets. For the *Low PVI* conditions, we observe that performance drops significantly. This indicates that the model overfits on those examples that are not relevant to the desired intent. We discuss the *All w/ relabelling* condition in Section 6.3.

In Figure 2, we plot the F1 score against the PVI score of the test set instances grouped by intent, showing that some classes are harder than others, further supporting why we need a threshold per class rather than a global one.

### Table 2: Intent Detection Accuracy (in %) in few-/full-shot settings with augmented data from OPT-66B. Numbers in bold are the best results and numbers with * are statistically significant by t-test ($p < 0.05$) compared to the baselines (5 / 10 examples per intent).

<table>
<thead>
<tr>
<th>Model</th>
<th>BANKING</th>
<th>HWU</th>
<th>CLINC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>10 Full</td>
<td>5</td>
</tr>
<tr>
<td>RoBERTa-Base + Class</td>
<td>74.04</td>
<td>84.27</td>
<td>75.56</td>
</tr>
<tr>
<td>USE</td>
<td>76.29</td>
<td>84.23</td>
<td>92.81</td>
</tr>
<tr>
<td>CONVERT</td>
<td>75.52</td>
<td>83.32</td>
<td>93.01</td>
</tr>
<tr>
<td>USE+CONVERT</td>
<td>77.75</td>
<td>85.19</td>
<td>93.36</td>
</tr>
<tr>
<td>CONVBERT</td>
<td>-</td>
<td>83.63</td>
<td>92.95</td>
</tr>
<tr>
<td>+ MLM</td>
<td>-</td>
<td>83.99</td>
<td>93.44</td>
</tr>
<tr>
<td>+ MLM + Example</td>
<td>-</td>
<td>84.09</td>
<td>94.06</td>
</tr>
<tr>
<td>+ Combined</td>
<td>-</td>
<td>85.95</td>
<td>93.83</td>
</tr>
<tr>
<td>DNNC</td>
<td>80.40</td>
<td>86.71</td>
<td>-</td>
</tr>
<tr>
<td>CPFT</td>
<td>80.86</td>
<td>87.20</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RoBERTa-Large + Class</td>
<td>78.99</td>
<td>86.08</td>
<td>93.70</td>
</tr>
<tr>
<td>+ ICDA-XS</td>
<td>80.29</td>
<td>86.72</td>
<td>-</td>
</tr>
<tr>
<td>+ ICDA-S</td>
<td>81.95</td>
<td>87.37</td>
<td>93.66</td>
</tr>
<tr>
<td>+ ICDA-M</td>
<td>84.01</td>
<td>88.64</td>
<td>93.73</td>
</tr>
<tr>
<td>+ ICDA-L</td>
<td>83.90</td>
<td>89.12</td>
<td>94.42</td>
</tr>
<tr>
<td>+ ICDA-XL</td>
<td>83.90</td>
<td>89.79</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 3: Intent Detection Accuracy (in %) for RoBERTa-Large model in 10-shot settings with ICDA-M synthetic instances from OPT-66B. Numbers in bold are statistically significant by t-test ($p < 0.05$).

<table>
<thead>
<tr>
<th>Model</th>
<th>BANKING</th>
<th>HWU</th>
<th>CLINC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RoBERTa-Large</td>
<td>86.08</td>
<td>84.11</td>
<td>93.56</td>
</tr>
<tr>
<td>All</td>
<td>84.19</td>
<td>84.57</td>
<td>94.24</td>
</tr>
<tr>
<td>All w/ relabelling</td>
<td>87.05</td>
<td>89.22</td>
<td>93.02</td>
</tr>
<tr>
<td>Global Low PVI</td>
<td>73.99</td>
<td>89.61</td>
<td>85.82</td>
</tr>
<tr>
<td>Global High PVI</td>
<td>87.38</td>
<td>86.27</td>
<td>94.27</td>
</tr>
<tr>
<td>Per-Intent Low PVI</td>
<td>76.49</td>
<td>71.84</td>
<td>89.33</td>
</tr>
<tr>
<td>Per-Intent High PVI</td>
<td><strong>88.64</strong></td>
<td><strong>87.36</strong></td>
<td><strong>94.71</strong></td>
</tr>
</tbody>
</table>
we randomly sample several synthetic instances from the OPT-66B generator and manually assess if each instance follows the same intent as the prompt label. We show some examples in Table 4. We observe that instances that are relevant to the desired intent are assigned high PVI values, and instances that are not relevant to the desired intent are assigned low PVI values. This further indicates that the per-intent threshold function provides an effective indicator of relevance. For example, in the BANKING dataset, most relevant instances have PVI values greater than 5.79, and most non-relevant instances have PVI values less than 5.79. This indicates that PVI filtering is an effective method for discarding mislabeled data points.

**ICDA produces fluent and diverse utterances.** We hypothesize that our proposed method is effective because it introduces more fluent and diverse utterances. We therefore compare synthetic data under the 10-shot XS condition (i.e., we generate 10 synthetic datapoints) with the original 10-shot datapoints taken from the training data. Then we use a GPT2 model trained on the test set of each benchmark dataset to calculate the perplexity of the generated utterances. We also use the same synthetic set to calculate the distinct-1, distinct-2, self-BLEU, and perplexity (PPL) metrics. We report the results in Table 5 and observe that our proposed method generates more diverse utterances as shown by distinct-1, distinct-2, and self-BLEU. This indicates that our proposed method harnesses the generation power of the OPT-66B generator. Additionally, the perplexity of synthetic utterances is slightly higher than the human-annotated training set. These results suggest that our proposed method generates more diverse utterances, which can help the task model to learn a better representation.
6.3 Data Relabelling

Following Sahu et al. (2022), we wanted to see if it is effective to use the available data to train an intent classifier and then use it to relabel the synthetic data. Intuitively, such a method would correct mistakes in the generation process. To test the feasibility of this approach, we train an oracle classifier using the entire training data of each dataset and use this as an upper bound. The results are shown in Table 3 (“All w/ relabeling”), where we see that while promising, this approach underperforms ICDA.

7 Conclusion

We introduced In-Context Data Augmentation, a novel data augmentation framework to generate synthetic training data, preserving quality and diversity. We demonstrate that ICDA is effective on multiple intent detection benchmarks, with state-of-the-art few-shot performance. Our analysis shows that ICDA tends to perform better in low-resource settings and that our PVI filtering strategy is important for performance. Future work includes applying ICDA to other conversational understanding tasks such as slot filling and dialogue state tracking, and incorporating other filtering or data selection strategies for further performance gains.

Limitations

In this section we take BANKING as a case study to motivate PVI and discuss some of the limitations of our approach. Figure 3 shows how much we gain (or lose) in F1 score when we use a custom threshold for each class vs. a fixed threshold. While most classes benefit, there are clearly many that show performance degradation. Another limitation is the size of the model we use to generate synthetic instances (OPT-66B); in general the larger the model is, the better the generated data is.

Ethical Considerations

As with any work involving PLMs (or foundation models), due to the data and training methods, there is inherent risk of generating biased, toxic, harmful, or otherwise unwanted output. Regarding our work in particular, as we show in Figure 3, the model’s performance on some of the classes can degrade. More analysis needs to be done before deploying our approach, since it is unclear whether it will introduce a bias towards certain types of classes.

References


Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021b. Few-shot intent


A Data Cartography and Uncertainty

Apart from relabelling, we investigated two additional approaches to rank synthetic instances as easy or hard to classify. We used data cartography (Swayamdipta et al., 2020) and classification uncertainty to guide our filtering.

Data cartography classifies the training data in four categories: Easy-to-learn, Low-Correctness, Ambiguous, Hard-to-Learn using training dynamics (i.e. the model’s confidence in the true class, and the variability of this confidence across epochs).

For uncertainty modeling, we assign uncertainty scores to each training instance in a cross-validation manner. We first split the training set into 5 folds, hold one fold out as validation, and predict on the validation with the classifier trained on the remaining 4 folds. We tried the following uncertainty measures: Contrastive Active Learning (AL) (Margatina et al., 2021), Least Confidence (Culotta and McCallum, 2005), Prediction Entropy (Schöhn and Cohn, 2000; Roy and McCallum, 2001), and Breaking Ties (Scheffer et al., 2001; Luo et al., 2004).

We conducted experiments using the above approaches to select data that amounts to one third of the total training data in BANKING (i.e., we select the top 33% hardest examples, etc.). As an additional baseline, we include a random filter, i.e., a randomly sampled 33% portion of BANKING. Table 6 shows the results, where we see that the performance actually degrades when compared to using the entirety of the data. We experimented with a few more variations in the filtering thresholds but no combination improved performance and we do not report those results here. See Figures 5 and 6 in the Appendix B for a visualization of the BANKING data map.

<table>
<thead>
<tr>
<th>Cartography</th>
<th>Uncertainty</th>
<th>100% Train</th>
<th>33% Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to Learn</td>
<td>Contrastive AL</td>
<td>90.44</td>
<td>88.54</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>Least Confidence</td>
<td>90.94</td>
<td>89.08</td>
</tr>
<tr>
<td>Low Correctness</td>
<td>Breaking Ties</td>
<td>91.00</td>
<td>89.20</td>
</tr>
<tr>
<td>Hard to Learn</td>
<td>Prediction Entropy</td>
<td>91.26</td>
<td>89.23</td>
</tr>
</tbody>
</table>

Table 6: Intent Detection Accuracy (in %) for ConvBERT model, trained on different selections of BANKING77 under full-shot settings.

B Figures
Figure 4: This figure shows the difference in F1 score for each intent, if we have a PVI threshold per-class VS having a fixed PVI threshold (Enlarged Figure 3).
Figure 5: Data map for BANKING (10-shot).

Figure 6: Data map for BANKING (full-shot).