Abstract

Data augmentation is a widely employed technique to alleviate the problem of data scarcity. In this work, we propose a prompting-based approach to generate labelled training data for intent classification with off-the-shelf language models (LMs) such as GPT-3. An advantage of this method is that no task-specific LM-fine-tuning for data generation is required; hence the method requires no hyper-parameter tuning and is applicable even when the available training data is very scarce. We evaluate the proposed method in a few-shot setting on four diverse intent classification tasks. We find that GPT-generated data significantly boosts the performance of intent classifiers when intents in consideration are sufficiently distinct from each other. In tasks with semantically close intents, we observe that the generated data is less helpful. Our analysis shows that this is because GPT often generates utterances that belong to a closely-related intent instead of the desired one. We present preliminary evidence that a prompting-based GPT classifier could be helpful in filtering the generated data to enhance its quality.\footnote{Work done during an internship at ServiceNow Research}

1 Introduction

A key challenge in creating task-oriented conversational agents is gathering and labelling training data. Standard data gathering options include manual authoring and crowd-sourcing. Unfortunately, both of these options are tedious and expensive. Data augmentation is a widely used strategy to alleviate this problem of data acquisition.

There are two particularly promising paradigms for data augmentation in natural language processing that use pretrained language models (LMs) (Peters et al., 2018; Devlin et al., 2018). The first family of methods fine-tunes an LM on task-specific data and generates new examples using the fine-tuned LM (Wu et al., 2018; Kumar et al., 2019, 2021; Anaby-Tavor et al., 2020; Lee et al., 2021). A limitation of these methods is that, in a real-world scenario, task-specific data is scarce and fine-tuning an LM can quickly become the bottleneck. The second family of methods sidesteps this bot-
tlenec by employing off-the-shelf pretrained LMs such as GPT-3 (Brown et al., 2020) to directly generate text without any task-specific fine-tuning. In particular, data generation by the GPT3Mix approach by Yoo et al. (2021) boosts performance on multiple classification tasks; however, they only consider tasks with few (up to 6) classes and easy-to-grasp class boundaries (e.g., positive and negative).

This work studies the applicability of massive off-the-shelf LMs, such as GPT-3 and GPT-J (Wang and Komatsuzaki, 2021) to perform effective data augmentation for intent classification (IC) tasks. In IC, the end goal is to predict a user’s intent given an utterance, i.e., what the user of a task-oriented chatbot wants to accomplish. Data augmentation for IC is particularly challenging because the generative model must distinguish between a large number (in practice up to several hundreds) of fine-grained intents that can be semantically very close to each other. Prior methods such as GPT3Mix prompt the model with the names of all classes as well as a few examples from randomly chosen classes. We test GPT3Mix for one and observe that such approaches are poorly suitable for intent classification tasks with tens or hundreds of possible intents. Instead, in this study, we use a simple prompt structure that focuses on a single seed intent (see Figure 1) as it combines the intent’s name and all available examples.

Our experiments primarily focus on few-shot IC on four prominent datasets: CLINC150 (Larson et al., 2019), HWU64 (Xingkun Liu and Rieser, 2019), Banking77 (Casanueva et al., 2020), and SNIPS (Coucke et al., 2018). We also consider a partial few-shot setup to compare to the Example Extrapolation (Ex2) approach by Lee et al. (2021) who use a similar prompt but fine-tune the LM instead of using it as is. The main findings of our experiments are as follows: (1) GPT-generated samples boost classification accuracy when the considered intents are well-distinguished from each other (like in CLINC150, SNIPS). (2) On more granular datasets (namely HWU64 and Banking77), we find that GPT struggles in distinguishing between different confounding intents. (3) A small-scale study to further understand this behaviour suggests that GPT could be used as a classifier to filter out unfaithful examples and enhance the quality of the generated training set. Additionally, we investigate how valuable the generated data could be if relabelled by a human. Using an oracle model, we show that (4) the human labelling of GPT-generated examples can further improve the performance of intent classifiers, and that (5) LM-generated data has a higher relabelling potential compared to edit-based augmentation techniques, such as Easy Data Augmentation (EDA) (Wei and Zou, 2019).

2 Method

We consider training an intent classifier, where an intent is a type of request that the conversational agent supports; e.g. the user may want to change the language of the conversation, play a song, transfer money between accounts, etc. However, collecting many example utterances that express the same intent is difficult and expensive. Therefore, this paper experiments with a straightforward method to augment the training data available for an intent: creating prompts from the available examples and feeding them to a large language model such as GPT-3 (Brown et al., 2020). Figure 1 illustrates the process of data generation for an intent with \( K \) available examples.

3 Experimental Setup

3.1 Datasets

We use four intent classification datasets in our experiments with varying levels of granularity among intents. CLINC150 (Larson et al., 2019), HWU64 (Xingkun Liu and Rieser, 2019) are multi-domain datasets, each covering a wide range of typical task-oriented chatbot domains, such as playing music and setting up alarms. Importantly, the CLINC150 task also contains examples of out-of-scope (OOS) utterances that do not correspond to any of CLINC’s 150 intents. Banking77 (Casanueva et al., 2020) is a single domain dataset with very fine-grained banking-related intents. Finally, the SNIPS (Coucke et al., 2018) dataset contains 7 intents typical for the smart speaker usecase. We refer the reader to Table 1 for exact statistics of all used datasets.

3.2 Setup

The main data-scarce setup that we consider in this work is the few-shot setup, where only \( K = 10 \) training examples are available for every intent of interest. Additionally, to compare to example extrapolation with fine-tuned language models as proposed by Lee et al. (2021), we consider a partial...
When data augmentation is performed, we augment data with four different-sized GPT-3 models: Ada, Babbage, Curie, and Davinci. In order, Ada is the smallest model and Davinci is the largest. Model sizes of GPT-3 engines are not known precisely but are estimated by Eleuther AI to be between 300M and 175B. In all setups we use the original validation set for tuning the classifier’s training hyperparameters. We chose to use the full validation set as opposed to a few-shot one to avoid issues with unstable hyperparameter tuning and focus on assessing the quality of the generated data.

### Full few-shot
In this setup, we treat all the intents as few-shot and evaluate our method on the following three scenarios: (i) **Baseline**: all the intents are truncated to \( K = 10 \) samples per intent, (ii) **Augmented**: \( D_{\{F,train\}} \) is generated using GPT and models are trained on \( D_{\{F,train\}} \cup D_{\{F,train\}} \) and (iii) **EDA-baseline**: same as above, but \( D_{\{F,train\}} \) is generated using Easy Data Augmentation (EDA) by Wei and Zou (2019). For each scenario, we report the 1) overall in-scope accuracy on the complete test set \( D_{test} \), i.e. intent classification accuracy excluding OOS samples in the test set, and 2) few-shot classification accuracy of the models on \( D_{\{F,train\}} \). For CLINC150, we also report out-of-scope recall (OOS recall) on \( D_{test} \) that we compute as the percentage of OOS examples that the model correctly labelled as such.

The purpose of this setting is to estimate what further gains can be achieved if the data generated by GPT were labelled by a human. We train an oracle \( O \) on the full training data \( D_{train} \). We also use \( O \) to assess the quality of the generated data. Namely, we compute fidelity of the generated data as the ratio of generated utterances that the oracle labels as indeed belonging to the intended seed intent. A higher fidelity value means that the generated samples are more faithful to original data distribution.

### Partial few-shot
In this setup, we train \( S \) intent classifiers, choosing different few-shot intents every time to obtain \( D_{F} \). We then average the metrics across these \( S \) runs. For CLINC150, \( S = 10 \) corresponding to the 10 different domains, whereas for SNIPS, \( S = 7 \) corresponding to the 7 different intents. We evaluate our method on the following three scenarios introduced by Lee et al. (2021): (i) **Baseline**: models are trained without data augmentation on \( D_{\{F,train\}} \cup D_{\{M,train\}} \). (ii) **Upsampled**: \( D_{\{F,train\}} \) is upsampled to have \( N \) examples per intent. Then models are trained on upsampled data.
Figure 2: Partial few-shot validation performance for different GPT-3 models and temperatures. (a) few-shot accuracy, (b) OOS recall of intent classifiers trained on augmented sets, and (c) fidelity measured as the accuracy of the oracle on the augmented sets.

\[ D_{\{F,\text{train}\}} \cup D_{\{M,\text{train}\}}. \]

(ii) Augmented: models are trained on \( D_{\{F,\text{train}\}} \cup \tilde{D}_{\{F,\text{train}\}} \cup D_{\{M,\text{train}\}}. \)

For each scenario in this setup, we report the overall in-scope classification accuracy (and OOS Recall for CLINC150).

For both partial few-shot and full few-shot settings, we report means and standard deviations over 10 repetitions of each experiment.

4 Experimental Results

Full few-shot. Table 2 shows the results of our few-shot experiments. For CLINC150 and SNIPS, data augmentation with GPT-3 is very effective as it leads to respective accuracy improvements of up to approximately 3.7% and 6% on these tasks. These improvements are larger than what the baseline EDA method brings, namely 2.4% and 2.9% for CLINC150 and SNIPS. Importantly, using larger GPT models for data augmentation brings significantly bigger gains. Data augmentation results on Banking77 and HWU64 are, however, much worse, with no or little improvement upon the plain few-shot baseline. We present a thorough investigation of this behaviour in Section 4.1. One can also see that data augmentation with GPT models lowers the OOS recall.

Next, we observe that relabelling EDA and GPT-generated sentences by the oracle gives a significant boost to accuracies across the board, confirming our hypothesis that human inspection of generated data could be fruitful. Importantly, we note that the magnitude of improvement for EDA is less than for GPT models. This suggests that GPT models generate more diverse data that can eventually be more useful after careful human inspection. Lastly, relabelling also improves OOS recall on CLINC150, which is due to the fact that much of the generated data was labelled as OOS by the oracle.

Partial few-shot. Table 3 shows the results of our partial few-shot experiments on CLINC150 and SNIPS. By augmenting the dataset with GPT-
### Table 2: Full few-shot results on CLINC150, HWU64, Banking77, and SNIPS datasets. IA: Inscope Accuracy (mean (std)). OR: OOS-Recall (mean (std)). Towards the top of the table, we also report the test set performance (enclosed in parentheses) when all examples are used for fine-tuning (without any augmentation.)

<table>
<thead>
<tr>
<th>Model</th>
<th>CLINC150 IA (96.93)</th>
<th>CLINC150 OR (42.9)</th>
<th>HWU64 IA (92.75)</th>
<th>Banking77 IA (93.73)</th>
<th>SNIPS IA (98.57)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDA</td>
<td>92.66 (0.40)</td>
<td>43.81 (2.03)</td>
<td>83.67 (0.48)</td>
<td>83.96 (0.66)</td>
<td>92.50 (1.61)</td>
</tr>
<tr>
<td>Baseline (Ours)</td>
<td>90.28 (0.49)</td>
<td>50.18 (1.14)</td>
<td>81.43 (0.57)</td>
<td>83.35 (0.59)</td>
<td>89.69 (1.63)</td>
</tr>
</tbody>
</table>

#### Augmented

<table>
<thead>
<tr>
<th>Model</th>
<th>CLINC150 IA (96.93)</th>
<th>CLINC150 OR (42.9)</th>
<th>HWU64 IA (92.75)</th>
<th>Banking77 IA (93.73)</th>
<th>SNIPS IA (98.57)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada (Ours)</td>
<td>91.31 (0.34)</td>
<td>21.69 (1.57)</td>
<td>79.68 (0.83)</td>
<td>79.30 (0.42)</td>
<td>94.27 (0.52)</td>
</tr>
<tr>
<td>Babbage (Ours)</td>
<td>92.72 (0.33)</td>
<td>22.99 (2.39)</td>
<td>81.86 (0.78)</td>
<td>80.31 (0.41)</td>
<td>94.74 (0.67)</td>
</tr>
<tr>
<td>Curie (Ours)</td>
<td>93.37 (0.21)</td>
<td>25.85 (1.49)</td>
<td>82.85 (0.70)</td>
<td>83.50 (0.44)</td>
<td>94.73 (0.62)</td>
</tr>
<tr>
<td>GPT-J (Ours)</td>
<td>93.25 (0.19)</td>
<td>24.02 (1.45)</td>
<td>81.78 (0.56)</td>
<td>82.32 (0.90)</td>
<td>95.19 (0.61)</td>
</tr>
<tr>
<td>Davinci (Ours)</td>
<td>94.07 (0.18)</td>
<td>27.36 (1.08)</td>
<td>82.79 (0.93)</td>
<td>83.60 (0.45)</td>
<td>95.77 (0.86)</td>
</tr>
</tbody>
</table>

#### Augmented + Relabelled

<table>
<thead>
<tr>
<th>Model</th>
<th>CLINC150 IA (96.93)</th>
<th>CLINC150 OR (42.9)</th>
<th>HWU64 IA (92.75)</th>
<th>Banking77 IA (93.73)</th>
<th>SNIPS IA (98.57)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada (Ours)</td>
<td>93.43 (0.22)</td>
<td>48.56 (1.84)</td>
<td>85.58 (0.73)</td>
<td>84.82 (0.57)</td>
<td>94.91 (0.66)</td>
</tr>
<tr>
<td>Babbage (Ours)</td>
<td>95.09 (0.16)</td>
<td>41.38 (1.77)</td>
<td>88.53 (0.61)</td>
<td>88.45 (0.19)</td>
<td>97.03 (0.18)</td>
</tr>
<tr>
<td>Curie (Ours)</td>
<td>95.39 (0.17)</td>
<td>40.58 (1.63)</td>
<td>89.49 (0.32)</td>
<td>88.86 (0.26)</td>
<td>96.89 (0.49)</td>
</tr>
<tr>
<td>GPT-J (Ours)</td>
<td>95.08 (0.19)</td>
<td>40.09 (2.38)</td>
<td>89.78 (0.47)</td>
<td>88.30 (4.64)</td>
<td>96.86 (0.31)</td>
</tr>
<tr>
<td>Davinci (Ours)</td>
<td>95.08 (0.13)</td>
<td>40.76 (1.37)</td>
<td>89.53 (0.45)</td>
<td>88.89 (0.31)</td>
<td>97.03 (0.38)</td>
</tr>
</tbody>
</table>

### Table 3: Partial few-shot results on CLINC150 and SNIPS datasets. Refer to Section 3.3 for more details. IA: Inscope accuracy (mean (std)). OR: OOS Recall (mean (std)). A: Accuracy (mean (std)). ♠ (Lee et al., 2021).

<table>
<thead>
<tr>
<th>Classifier</th>
<th>CLINC150 Overall</th>
<th>CLINC150 Few-shot</th>
<th>SNIPS Overall</th>
<th>SNIPS Few-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline ♠</td>
<td>T5 97.4</td>
<td>93.7</td>
<td>95.2</td>
<td>74.0</td>
</tr>
<tr>
<td>Upsampled ♠</td>
<td>T5 97.4</td>
<td>94.4</td>
<td>95.9</td>
<td>80.0</td>
</tr>
<tr>
<td>Augmented (Ex2)♠</td>
<td>T5 97.4</td>
<td>95.6</td>
<td>97.8</td>
<td>94.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classifier</th>
<th>CLINC150 Overall</th>
<th>CLINC150 Few-shot</th>
<th>SNIPS Overall</th>
<th>SNIPS Few-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline ♠</td>
<td>BERT 96.28 (0.06)</td>
<td>39.14 (0.82)</td>
<td>91.36 (0.47)</td>
<td>95.47 (0.45)</td>
</tr>
<tr>
<td>Upsampled ♠</td>
<td>BERT 96.20 (0.05)</td>
<td>40.21 (0.59)</td>
<td>90.93 (0.19)</td>
<td>95.29 (0.37)</td>
</tr>
<tr>
<td>Augmented (Ada)♠</td>
<td>BERT 96.16 (0.05)</td>
<td>34.37 (0.27)</td>
<td>92.60 (0.15)</td>
<td>97.30 (0.24)</td>
</tr>
<tr>
<td>Augmented (Babbage)♠</td>
<td>BERT 96.39 (0.06)</td>
<td>35.71 (0.46)</td>
<td>93.66 (0.21)</td>
<td>97.46 (0.25)</td>
</tr>
<tr>
<td>Augmented (Curie)♠</td>
<td>BERT 96.41 (0.06)</td>
<td>36.77 (0.93)</td>
<td>93.90 (0.21)</td>
<td>97.37 (0.19)</td>
</tr>
<tr>
<td>Augmented (GPT-J)♠</td>
<td>BERT 96.38 (0.05)</td>
<td>35.91 (0.94)</td>
<td>93.85 (0.25)</td>
<td>97.59 (0.21)</td>
</tr>
<tr>
<td>Augmented (Davinci)♠</td>
<td>BERT 96.45 (0.03)</td>
<td>37.52 (0.54)</td>
<td>94.28 (0.24)</td>
<td>97.66 (0.21)</td>
</tr>
</tbody>
</table>

generated samples, the few-shot accuracy improves by up to 2.92% on CLINC150 and 18.14% on SNIPS compared to the baseline setting. Our method achieves competitive results compared to Ex2 (Lee et al., 2021), both in terms of absolute accuracies and the relative gains brought by data augmentation. Note that Ex2 uses T5-XL (Roberts et al., 2020) with nearly 3 billion parameters as its base intent classifier, while our method uses BERT-large with only 340 million parameters.

### 4.1 Analysis

**Effect of GPT sampling temperature.** We investigate the impact of generation temperature on the quality and fidelity of generated data. We perform this investigation on the CLINC150 dataset using the partial few-shot setup. Results in Figure 2 show that, for all engines, the generated data leads to the
highest classification accuracy when the generation temperature is around 1.0, although lower temperatures result in higher OOS recall. We also observe that the fidelity of the generated samples decreases as we increase the temperature (i.e., higher diversity, see Figure 2c). This suggests that higher fidelity does not always imply better quality samples as the language model may simply copy or produce less diverse utterances at lower temperatures. In Appendix A, we perform a human evaluation, reaching similar conclusions as when using an oracle to approximate fidelity.

Fidelity on different datasets. Our results in Section 4 show that data augmentation gains are much higher on CLINC150 and SNIPS than on HWU64 and Banking77. To contextualize these results, we report the fidelity of GPT-J-generated data for all these tasks in Figure 3b. Across all generation temperatures, the fidelity of the generated data is higher for CLINC150 and SNIPS than for HWU64 and Banking77. For all datasets, the fidelity is higher when the generation temperature is lower; however, Figure 3a shows that low-temperature data also does improve the model’s performance.

Data generation for close intents. To better understand the lower fidelity and accuracy on HWU64 and Banking77 datasets, we focus on intents with the lowest fidelities. Here, by intent fidelity, we mean the percentage of the intent’s generated data that the oracle classified as indeed belonging to the seed intent. In the Banking77 dataset, the lowest-fidelity intent is “topping_up_by_card.” For this intent, only 33% of the Davinci-generated sentences were labelled as “topping_up_by_card”
Table 5: The impact and the accuracy of using GPT-3 as a 3-way classifier on close intent triplets from HWU64 and Banking77 datasets. For fidelity, generated examples are rejected if the GPT-3 classifier labels them as not belonging to the seed intent. Classification accuracies are reported on the reduced validation+test sets where we only consider examples from the three confounding intents.

<table>
<thead>
<tr>
<th>Fidelity (3 intents)</th>
<th>HWU64</th>
<th>Banking77</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o filtering (468)</td>
<td>60.26</td>
<td>57.31</td>
</tr>
<tr>
<td>w/ filtering (371)</td>
<td>72.51</td>
<td>65.54</td>
</tr>
<tr>
<td>3-way accuracy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Davinci</td>
<td>86.36</td>
<td>78.75</td>
</tr>
<tr>
<td>10-shot BERT-large</td>
<td>82.95</td>
<td>65.54</td>
</tr>
<tr>
<td>Full data BERT-large</td>
<td>94.32</td>
<td>95.00</td>
</tr>
</tbody>
</table>

Table 4 presents generated sentences for confounding intents in the HWU64 and Banking77 datasets. There are clear indications of mix-up of intents, e.g., Davinci generates, “play a song with the word honey,” which should belong to “play_music” rather than “music_likeness.”

4.2 Can GPT Models Understand Close Intents?

We perform extra investigations to better understand what limits GPT-3’s ability to generate data accurately. We hypothesize that one limiting factor can be GPT-3’s inability to understand fine-grained differences in the meanings of utterances. To verify this hypothesis, we evaluate how accurate GPT-3 is at classifying given utterances as opposed to generating new ones. Due to the limited prompt size of 2048 tokens, we can not prompt GPT-3 to predict all the intents in the considered datasets. We thus focus on the close intent triplets from HWU64 and Banking77 datasets that we use in Table 4. We compare the 3-way accuracy of a prompted GPT-3 classifier to the similarly-measured 3-way performance of conventional BERT-large classifiers. We prompt GPT-3 with 10 examples per intent (see Figure 5). For comparison, we train BERT-large classifiers on either the same 10 examples or the full training set. Table 5 shows that the Davinci version of GPT-3 performs in between the 10-shot and the full-data conventional classifiers. This suggests that while GPT-3’s understanding of nuanced intent differences is imperfect, it could still be sufficient to improve the performance of the downstream few-shot model. Inspired by this finding, we experiment with using GPT-3’s classification abilities to improve the quality of generated data. Namely, we...
reject the generated utterances that GPT-3 classifies as not belonging to the seed intent. For both HWU64 and Banking77, this filtering method significantly improves the fidelity of the generated data for the chosen close intent triplets.

4.3 Comparison with GPT3Mix

To test our initial hypothesis that prior methods such as GPT3Mix are not suitable for intent classification, we experiment with the said method on the CLINC150 dataset using Curie. Specifically, we include an enumeration of the 150 intent names in the prompt and randomly select one example for K intents. We observe a poor in-scope accuracy of 86.33\% in the Augmented scenario\(^6\). Furthermore, the generated samples have low fidelity (27.96\%).

We also test a mixture of GPT3Mix prompt and our prompt where we include all the K examples for the seed intent instead of 1 example per K randomly sampled intents. This mixed variant also performs poorly on CLINC150 and only achieves an in-scope accuracy of 86.05\%\(^7\) and a fidelity of 33.56\%. Our interpretation of this result is that GPT cannot handle the long list of 150 intent names in the prompt.

5 Related Work

The natural language processing literature features diverse data augmentation methods. Edit-based methods such as Easy Data Augmentation apply rule-based changes to the original utterances to produce new ones (Wei and Zou, 2019). In back-translation methods (Sennrich et al., 2016) available examples are translated to another language and back. Recently, data augmentation with fine-tuned LMs has become the dominant paradigm (Wu et al., 2018; Kumar et al., 2019, 2021; Anaby-Tavor et al., 2020; Lee et al., 2021). Our simpler method sidesteps LM-fine-tuning and directly uses off-the-shelf LMs as is.

The data augmentation approach that is closest to the one we use here is GPT3Mix by Yoo et al. (2021). A key part of the GPT3Mix prompt is a list of names of all possible classes (e.g. “The sentiment is one of ‘positive’ or ‘negative’”). The LM is then expected to pick a random class from the list and generate a new example as well as the corresponding label. However, this approach does not scale to intent classification setups, which often feature hundreds of intents (see Section 4.3). Therefore, we choose a different prompt that encourages the model to extrapolate between examples of a seed intent similarly to (Lee et al., 2021).

Other work on few-shot intent classification explores fine-tuning dialogue-specific LMs as classifiers as well as using similarity-based classifiers instead of MLP-based ones on top of BERT (Vulić et al., 2021). We believe that improvements brought by data augmentation would be complementary to the gains brought by these methods.

Lastly, our method to filter out unfaithful GPT generations is related to the recent work by Wang et al. (2021) that proposes using GPT3 for data labelling. A crucial difference with respect to our work, however, is that we use GPT-3 for rejecting mislabelled samples rather than proposing labels for unlabelled samples.

6 Conclusion

We propose a prompt-based method to generate intent classification data with large pretrained lan-

---

\(^6\)Average of 10 runs with a standard deviation of 1.17

\(^7\)Average of 10 runs with a standard deviation of 0.59
guage models. Our experiments show that generated data can be helpful as additional labelled data for some tasks, whereas, for other tasks, the generated data needs to be either relabelled or filtered to be helpful. We show that a filtering method that recasts the same GPT model as a classifier can be effective. Our filtering method, however, requires knowing the other intents that the generated data is likely to belong to instead of the seed intent. Future work can experiment with heuristics for approximately identifying the most likely actual intents for the generated utterances. This would complete a data generation and filtering pipeline that, according to our preliminary results in Section 4.2 here, could be effective. Other filtering methods could also be applied, such as looking at the likeliness of the generated utterances as explored in a concurrent work by Meng et al. (2022). Lastly, an interesting future work direction is identifying which generated utterances most likely need a human inspection.

7 Ethical Considerations

As discussed for the GPT3Mix method in Yoo et al. (2021), using large language models for data augmentation presents several challenges: they exhibit social biases and are prone to generating toxic content. Therefore, samples generated using our prompting-based approach need to be considered carefully.

To address such ethical concerns, human inspection would be the most reliable way to identify and filter out problematic generations. The practitioners who apply our method may also consider debiasing the language model before using it for generation (Schick and Schütze, 2021).

References


Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke


Appendix

A Human Evaluation

In Figure 2 we evaluate the fidelity of the samples generated by GPT-3 with respect to the original set of sentences used to prompt it. Fidelity is approximated by the classification performance of an "oracle" intent classifier trained on the whole dataset ($D_{train} \cup D_{test}$) and evaluated over the generated samples. In order assess whether the oracle predictions are comparable to those of a human, we perform a human evaluation study.

![Human Identification Performance](image)

**Figure 6:** Human evaluation. Error rate of human evaluators at the task of finding whether any sentence in a group of 5 was generated by GPT-3 or not. Each color represents a different GPT-3 engine. Higher error rate indicates that humans could not correctly identify generated samples and thus it also indicates higher fidelity. The standard error is displayed as a vertical line on top of each bar.

Help us decide which sentences are generated by a human or a model. Note, that there could be either one or zero sentences generated by a model. Thanks for your time!

![Human evaluation tool](image)

**Figure 7:** Human evaluation tool. Example of a question for the human evaluators. Human evaluators are asked to flag which example is GPT-3 generated if any among the 5 presented ones.

We consider that a model produces sentences with high fidelity if a human is unable to distinguish them from a set of human-generated sentences belonging to the same intent. Therefore, for each intent in the CLINC150 dataset, we sample five random examples and we randomly choose whether to replace one of them by a GPT-3 generated sentence from the same intent. We generate sentences with each of the four GPT-3 models considered in the main text with two different temperatures (0.8 and 1.0). The sentence to replace is randomly selected. Finally, the five sentences are displayed to a human who has to choose which of the sentences is generated by GPT-3, if any.

The task is presented to human evaluators in the form of a web application (see Figure 7). We placed a button next to each sentence in order to force human evaluators to individually consider each of the examples. Once annotated, the evaluator can either submit, discard, or leave the task to label later. We used a set of 15 voluntary evaluators from multiple backgrounds, nationalities, and genders. Each evaluator annotated an average of 35 examples, reaching a total of 500 evaluated tasks.

For each model and temperature, we report the error rate of humans evaluating whether a task contains a GPT-generated sample. We consider that evaluators succeeds at a given task when they correctly find the sentence that was generated by GPT or when they identify that none of them was generated. Thus, the error rate for a given model and temperature is calculated as #failed / total_evaluated.

Results are displayed in Figure 6. We find that human evaluators tend to make more mistakes when the temperature used to sample sentences from GPT-3 is smaller. This result is expected since lowering the temperature results in sentences closer to those prompted to GPT-3, which are human-made. We also observe that models with higher capacity such as Davinci tend to generate more indistinguishable sentences than lower-capacity models such as Ada, even for higher temperatures. These results are in agreement with the "oracle" fidelity results introduced in Figure 2.