Improving Dialogue Act Recognition with Augmented Data

Khyati Mahajan*[1], Soham Parikh[2], Quaizar Vohra[2], Mitul Tiwari[2], and Samira Shaikh[1]

[1]UNC Charlotte
[2]ServiceNow, Inc.
{kmahaja2,samirashaikh}@uncc.edu
{soham.parikh,quaizar.vohra,mitul.tiwari}@servicenow.com

Abstract

We present our work on augmenting dialogue act recognition capabilities utilizing synthetically generated data. Our work is motivated by the limitations of current dialogue act datasets, and the need to adapt for new domains as well as ambiguity in utterances written by humans. We list our observations and findings towards how synthetically generated data can contribute meaningfully towards more robust dialogue act recognition models extending to new domains. Our major finding shows that synthetic data, which is linguistically varied, can be very useful towards this goal and increase the performance from $0.39, 0.16$ to $0.85, 0.88$ for AFFIRM and NEGATE dialogue acts respectively.

1 Introduction

Virtual assistants have been deployed towards helping users perform various tasks, such as setting up a credit card. Behind the scenes, most dialogue systems powering these virtual assistants are built of various components which facilitate Natural Language Understanding (NLU). One such critical component is dialogue state tracking (DST), which helps systems recognize the current state and intent of the user in the conversation. DST often consists of three main sub-components - intent classification, slot filling and dialogue act recognition (DAR). Dialogue acts describe how the dialogue state should be modified from a system perspective, whereas the intents and slots help identify the user's intent in an utterance. These sub-components are usually built separately for industrial applications, since DAR could be generalizable, while intents and slots vary with the intended task or service.

Since DST can be subjective, large-scale industrial applications need to rise to many challenges, including supporting heterogeneous services and APIs. The Schema-Guided Dialogue (SGD) State Tracking task at the Eighth Dialogue System Technology Challenge (DSTC8) (Rastogi et al., 2020) introduced a dataset which could help handle these challenges, towards being able to handle multiple services and APIs while not requiring the collection of new data or retraining models. The SGD dataset includes various dialogue acts as well as intents, one of the first to allow multiple APIs with overlapping functionality in each domain.

Out of the 3 sub-components for DST, we observe that training models separately towards dialogue act recognition allows better internal utilization, since dialogue acts are similar across virtual assistant tasks and customers, whereas intent recognition and slot filling can vary across customers as well as customer specific tasks. Keeping this in mind, we focus our research towards developing robust, generalizable DAR models.

Since SGD is one of the most dialogue act-rich datasets, we explore its applications towards training dialogue act recognition models for confidential internal data. However, during our experiments, we observe that the performance drops significantly (from 0.98 to 0.39 F1 for 'AFFIRM'). Digging deeper, we observe that the form of responses for certain dialogue acts could be improved with adding variations. For example, majority of 'AFFIRM' utterances include or start with the word 'yes'. We conduct more experiments for 'AFFIRM' and 'NEGATE', and present our observations further details in Section 5.

We study the limitations of the dialogue act recognition models trained on SGD and tested on confidential internal data. We focus our study on understanding the performance for AFFIRM and NEGATE, and looking for the existence of similar patterns in existing data which could lead to overfitting. To bolster the generalizability of the model to new domains, we explore and implement data augmentation strategies which help add more variety to the form of the utterances in the dataset.

*Work conducted during an internship at ServiceNow, Inc.

Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 471 - 479
December 7, 2022 ©2022 Association for Computational Linguistics

We present all our findings in this paper, focusing mainly on our data augmentation techniques which utilize synthetic text generation methods. Overall, our main contributions thus focus around the following studies:

1. We observe shortcomings of the variation of forms in the utterances within the Schema Guided Dialogue (SGD) dataset.
2. We study the limitations of dialogue act recognition models trained on SGD and their poor generalization on internal data (generated by linguists).
3. We present synthetic data generation techniques employed towards overcoming the aforementioned shortcomings, built with OpenAI's GPT-3 (Brown et al., 2020). We also showcase their effectiveness towards better generalization for new domains with the aforementioned dialogue act models.

2 Related Work

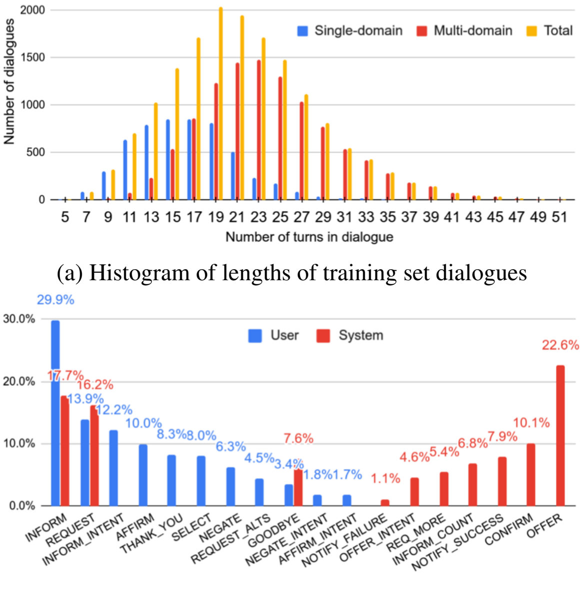
Dialogue state tracking, and consequently dialogue act recognition, are integral components of task-oriented dialogue systems. Recent research often focuses on utilizing neural methods towards approaching these tasks (Balaraman et al., 2021; Jacqmin et al., 2022), and both surveys find that generalizability in dialogue state tracking is understudied. They both also present various strategies towards data augmentation, including but not limited to training on resource-rich domains and applying to unseen domains (similar to SGD), using weak supervision to identify slots, reformulating dialogue state tracking as dialogue summarization to leverage external annotated data, using reinforcement learning towards generating relevant data, and prompting generative models to address unseen domains. Many of the aforementioned methods focus on the intent slot values, and not the dialogue acts. Since our work focuses mainly on dialogue act recognition, we include relevant work towards this task in this section.

2.1 Dialogue Act Recognition

Research in dialogue act modeling and recognition (DAR) has employed both statistical methods such as Bayesian classification (Grau et al., 2004), Conditional Random Fields (CRFs) (Stolcke et al., 2000), Hidden Markov Models (HMMs) (Boyer et al., 2010), and Support Vector Machines (SVMs) (Tavafi et al., 2013). Recent studies utilize neural methods such as LSTMs (Kumar et al., 2018) and structured attention network (Chen et al., 2018) with a CRF classification layer. Since most research utilizes different corpora and dialogue acts, it is difficult to compare performance across literature. However, recent research has moved further into utilizing neural methods, showing their viability of adapting to a wide range of dialogue acts as well as corpora, such as seq2seq models with attention (Colombo et al., 2020; Raheja and Tetreault, 2019).

Most recent research utilizes contextual models towards DAR (Ahmadvand et al., 2019; Saha et al., 2019), moving further towards utilizing neural methods. More recently, Noble and Maraev (2021) experiment with BERT towards DAR, and find that while pre-trained models like BERT perform well, the performance is much better with fine-tuning.

Learning from these studies as well as drawing takeaways from the Dialogue State Tracking Challenge 8 (DSTC 8) (Rastogi et al., 2020), we implement a BERT-based model in our DAR, which is also fine-tuned on the training dataset. We also keep in mind the class imbalances involved since DAR is a multi-class classification task, and describe our experiments and results in Section 5.



(a) Histogram of lengths of training set dialogues

(b) Distribution of dialogue acts in training set

Figure 1: Statistics for SGD (Rastogi et al., 2020)

2.2 Synthetic Data Generation

We utilize synthetic data generation to boost the capabilities of our DAR model. We follow this method since it has been shown in the past to augment the performance of NLP classification in varied applications (Whitfield, 2021; Bartolo et al., 2021; Bonifacio et al., 2022), and since it can also help boost the performance on our private, confidential data towards better DAR. Towards this goal, we look at various strategies for data generation.

Apart from learning from the OpenAI Completions guidelines, we also draw from findings of recent work on synthetic data generation (Reynolds and McDonell, 2021). We choose to work with GPT-3 mainly since it is the current state-of-the-art for off-the-shelf text generation, and we aim to generate synthetic data with varied linguistic forms which GPT3 is highly suitable for. Thus, GPT-3 provides us a method to generate relevant and robust synthetic data without the need to fine-tune a text generation model.

We refer to current literature for further guidelines and useful strategies. There exist various paradigms which intend to help with text generation based on the generations goals, such as utilizing Reinforcement Learning (RL) techniques or Q-learning (Guo et al., 2021), and AutoPrompt (Shin et al., 2020) which uses a gradient-based search. However, these lack the interpretability which applies to our goal, and secondly they require a specified goal towards which to tune the generations. Thus we focus more on manual experimentation which could provide us with clearer takeaways for future, more subjective generations (presented in Section 4.3).

	SGD	Internal data	Test set
# of dialogues	16142	161	296
# of utterances	164982	1980	1629
# of AFFIRM	25054	1375	808
# of NEGATE	16715	605	821

Table 1: Details for each dialogue act in each dataset used for training the dialogue act recognition model. The test set is hand written by linguists.

3 Datasets

This section details the datasets we utilize in our experiments, and provides aggregate details for each without revealing private and protected information for confidential internal data. For the purposes of our research, we focus mainly on a few dialogue acts relevant towards confidential internal applications - namely AFFIRM and NEGATE. Each section describes how we build the utterance and associated dialogue act pairs towards dialogue act recognition. The final statistics for each dataset is presented in Table 1.

3.1 Schema Guided Dialogue (SGD)

The Schema Guided Dialogue (SGD) dataset consists of over 20k annotated multi-domain, task-oriented conversations between a human and a virtual assistant, spanning 20 domains such as banks, events, media, calendar, travel, and weather (Shah et al., 2018; Rastogi et al., 2020). Figure 1a shows the distribution of dialogue lengths across single-domain (average 15.3 turns) and multi-domain dialogues (average 23 turns). Figure 1b shows the frequency of the different dialogue acts contained in the dataset. The dataset also contains a significant number of unseen domains/APIs in the dev and test sets. 77% of the dialogue turns in the test set and 45% of the turns in dev set contain at least one service not present in the training set.

Each utterance in the SGD dataset comes with relevant information including a breakdown of all the dialogue acts and slots present in the utterance. Our model predicts all the dialogue acts associated with each frame. The final statistics for the dataset thus built is presented in Table 1.

3.2 Confidential Internal Data

The confidential internal data we utilize for researching the transferability and generalizability of the SGD dataset follows the same structure as the SGD data. This data reflects the services built on our virtual assistant which are similar to SGD but are specific to our customer domains.

The training set for the internal data is synthetically generated using GPT-3. We used different prompts to generate synthetic data with a large variety of dialogue act patterns. We expect that this variety will help our dialogue act models generalize well to unseen domains and use cases. There is evidence of this as shown by experiments described in 5. Our test set, modeling real user traffic, is created separately and annotated by Subject Matter Expert (SME) linguists. The statistics for this dataset are presented in Table 1.

This dataset has a much greater variety on the dialogue acts we experiment with in this paper compared to SGD. For example, almost all AFFIRMs in the SGD dataset have a ‘yes’ or a very similar strong affirmative word. In our test set, we have a rich variety of patterns, including explicit affirmatives, implicit affirmatives using context from the

	Prompt Type	Prompt Prefix	Prompt Sub-type	Prompt Affix	Generated Text
Few-shot	Full context	Example 1: Bot: Here are the available rooms near you. Do you want to book selected conference room? User: [PROMPT AFFIX] *[2 more examples]* Example 4: *[conversation context]* Bot: Do you have the account number?	[with context]	Yes, please book the room	yes i do
			[without context]	Yes, please go ahead.	yes
			[only context]	Book selected room.	i have the number.
	Immediate context	Example 1: Question: Here are the available rooms near you. Do you want to book selected conference room? Affirm: [PROMPT AFFIX] *[2 more examples]* Example 4: Question: Do you have the account number?	[with context]	Yes, please book the room	yes
			[without context]	Yes, please go ahead.	yes
			[only context]	Book selected room.	yeah
Zero-shot	Full context	Generate multiple positive responses to the question using only words from the question. *[conversation context]* Bot: [QUESTION] User:	-	-	• Yes I do. • Yes, I have the account number. • Yes, I can give you the account number. • The account number is ___ .
	Immediate context	Generate multiple positive responses to the question using only words from the question. Bot: [QUESTION] User:	-	-	• Yes, I have the account number. • Great, what is the account number? • The account number is ___. • Thank you for providing the account number.

Table 2: Prompt experiments, listing all the types of prompts we used and samples from the text GPT3 generated

conversation with a virtual agent as well as a mix of the two. This is further described in 4.

3.3 Observations

As discussed earlier, we observe that there exist a few shortcomings in the SGD data, mainly related to the variety in the form of utterances in each dialogue act. Out of $15k$ utterances with AFFIRM or AFFIRM_INTENT as the sole dialogue acts, only $4k$ of them are unique. Moreover, over 70% of all AFFIRM utterances contain the word ‘yes’ or its variations like ‘yup’, ‘yep’, ‘yeah’ or start with ‘sure’. Similarly, more than 80% of all NEGATE utterances start with ‘no’ or ‘nope’ and out of $2.7k$ utterances in NEGATE or NEGATE_INTENT, only $1.2k$ are unique. We see a similar distribution in test and validation sets as well, leading us to believe that the existence of this predictable pattern is what contributes to the strong performance baselines.

Acting on our findings, we experiment with adding synthetically generated data to our dataset. We choose this augmentation method since it allows us to contribute relevant yet original data, while generating varied forms and structures for each utterance. We first experiment with an SGD-fine-tuned data (Section 5.1, and find that this lack of variety does indeed lead to worse predictions on our rich SME linguist generated test set. We present methods and evaluation techniques used to overcome these shortcomings by generating synthetic data (Section 4).

4 Synthetic Data Generation

We aim to augment our dataset using synthetic data generated by prompting GPT-3, as described in Section 2. We detail our experiments and their results in this section.

4.1 Experiments

We utilize OpenAI’s GPT-3 Completions API to generate synthetic data which could be useful towards mitigating the effects of the presence of patterns in the training data. We experiment with different kinds of prompts, following guidelines laid out by OpenAI[1] for text generation (detailed in Table 2).

The main prompt types we experiment with include few-shot and zero-shot. In the few-shot setting, the prompt consists of a few examples (3 to 5 examples) which can help demonstrate the completions we expect. In the zero-shot setting, the prompt includes an instruction along with the question. Additionally, we frame prompts so as to generate different kinds of responses for both AFFIRM and NEGATE. For example, each yes/no question (such as “Would you like to continue?") can be answered by a human in 3 different ways,

[1] https://beta.openai.com/docs/guides/completion

1) with-context (such as “Yes, I would like that"), 2) without-context (such as “Yes please"), and 3) only-context (such as “Please continue").

In addition to various framing, we also experiment with both the *Curie* and *Davinci* engines, although we conclusively find that *Davinci* performs better in initial experiments. Thus, the results included in this paper are all generations using the *Davinci* engine.

We find that some prompts perform better than others for different kinds of expected generations. We discuss our evaluation strategies next, and present our findings and takeaways in Section 4.3.

4.2 Evaluation

We employ multiple strategies for evaluating the generated synthetic data, consisting of both automatic and human evaluation methods. We employ custom automatic evaluation metrics, such as the presence of key words, to ensure that we generate different kinds of variations. For human evaluations, we work with subject matter experts (SMEs), who hand annotate each synthetic generation as good, alright, or bad generations, depending on our generation goal with a specific prompt. Further details for our evaluation strategies is presented in Table 3.

Tables 4 and 5 list the performance of our major experiments. We required fewer experiments for NEGATE since we were able to learn from our takeaways stemming from our experiments with AFFIRM.

For AFFIRM, the few-shot examples are listed in Table 2. With zero-shot prompts, Type 1 consisted of the instruction “Generate multiple positive responses as a human would to the following question asked by a bot:", Type 2 consisted of “Imagine a conversation between a bot and a human. Generate multiple positive responses to the following question asked by the bot:", Type 3 consisted of “Generate multiple positive responses as a human would to the following question asked by a bot:", Type 4 consisted of “Generate multiple affirmative responses as a human would to the following question asked by a bot:", and Type 5 consisted of “Generate multiple responses that agree with the question using only words from the question. Do not use the word “yes".". Evaluations for how well each of the generations perform are shown in Table 3. Few-shot, full context prompts do perform the best, however these require a heavy payload to the API and thus cost more (compared to zero-shot prompts). Thus, we focus more on improving the zero-shot prompts, and find that instructions which include multiple yet simple asks perform best. Further takeaways are discussed in the next section.

For NEGATE, Type 1 consisted of “Generate at least 5 negative responses as a human would to the following question asked by a bot. Do not generate positive responses:", while Type 2 consisted of “Generate at least 5 responses that disagree with the following question asked by a bot. Do not generate positive responses:". As shown, the performance is comparable for both prompts.

Combining the automatic and human evaluation metrics allows us to better gauge the effectiveness of our prompts. The SME linguists also provided us with deeper insights into patterns associated with prompt wording. In general, we find that utilizing both instructions and examples can help generate more relevant data.

4.3 Discussion

We observe that many data points in the SGD dataset consist of utterances which contain a keyword like ‘yes’ or ‘yeah’, which can become a pattern that signifies the dialogue act for AFFIRM, and similarly for NEGATE. Therefore, our prompts aim to generate data points which could serve as utterances which have the context of the preceding utterance (generally a REQUEST). Through our experiments, we observe a number of relevant takeaways for generation with GPT-3.

Firstly, we discover that if a REQUEST is posed as a statement and not a question, ie if the REQUEST does not end with a question mark, then the Completions API tends to hallucinate wildly, even if relevant contextual information (such as the preceding conversation) is available. For example, if a prompt asking for a laptop replacement ends with “requesting a loaner”, the output first hallucinates and generates completions such as “vehicle of make and model?”. We also experiment with removing question marks in a few cases where we observe good synthetic generations, and find that we are able to replicate this problem. Therefore, there is a need to ensure that prompts end with a question mark if the objective is to generate relevant responses.

Secondly, it is always better to show the completions API that a question was spoken by a bot,

Dialogue Act	Automatic	Human
Both	1. Word count 2. Jaccard similarity with REQUEST 3. GUSE similarity with REQUEST 4. All scores averaged	1. Grammaticality & Fluency 2. Follows dialogue constraints (ex, conversation flow) 3. Follows cooperative principle (effective communication)
AFFIRM	Presence of ‘yes’ and related words	Variety in form and linguistic features
NEGATE	Presence of ‘no’ and related words	Variety in form and linguistic features

Table 3: Evaluation metrics used for evaluating synthetically generated data

Prompt Type	Generation Type	Good Generations
Few-shot, full context	with-context	7
	without-context	37
	only-context	**75**
Zero-shot	Type 1	48
	Type 2	49
	Type 3	56
	Type 4	58
	Type 5	**67**

Table 4: AFFIRM prompts and performance for a total of 81 data points - bold text shows best performance

Prompt Type	Generation Type	Good Generations
Zero-shot	Type 1	48
	Type 2	47

Table 5: NEGATE prompts and performance for a total of 67 data points

rather than instruct the API to generate completions for questions posed by a bot. This becomes important for our use-case since we are aiming to generate responses that sound like they are coming from a user who is interacting with the bot. Therefore, we want succinct yet easy to understand responses which can be easily understood by a dialogue system, coming from the user's point of view. Thus, it is useful to have prompts such as “Generate responses to the following question. Bot: Would you like me to proceed? User:” as compared to “Generate responses for the following question asked by a bot. Would you like me to proceed?”.

Lastly, we observe that using simple, small but multiple instructions works better than using long and complex instructions. For example, the prompt “Generate at least 5 negative response to the following question. Be polite. Do not use `no`” works better than “Generate multiple polite negative responses to the following question without saying `no`”.

Overall, our findings echo many of the guidelines suggested by OpenAI while also showcasing that prompt design requires experimentation to fit into specific use-cases. Especially in scenarios where there is a need to report on which prompts worked better and to understand why, as well as a subjective view of which synthetic generations would be the best addition to training data, soft prompting and prompt tuning become difficult to implement. We therefore focus our efforts on understanding the underlying conditions and guidelines under which we are able to generate synthetic data which eventually can boost dialogue act recognition.

5 Dialogue Act Recognition

We show how our synthetically generated data can boost the capabilities of dialogue act recognition models in this section. We detail each step in our experiments as well as our findings.

5.1 Experimental Setup

We utilize the SGD training set, synthetic generations using OpenAI and a fraction of hand-written generations produced by our SME linguists as training data. For the synthetic data, we generate AFFIRM and NEGATE utterances using various prompts which are then filtered by human experts as relevant or not. Unless otherwise mentioned, we only use the relevant synthetic generations for training.

We evaluate the model on both SGD test set and gold standard SME linguist generated test data, specifically written to include several forms of utterances for each dialogue act, making it difficult to achieve perfect performance on them. We report the F1 score computed separately for each dialogue act, averaged across 3 training runs with different random seeds.

For all our experiments, we fine-tune the BERT-small model to predict dialogue acts. We train the model for 4 epochs with a learning rate of $1e$-5 and a batch size of 64. As input to the model, we concatenate the previous and current utterance with a [SEP] token. Predicting dialogue acts is a multi-label task and hence we use a sigmoid activation for the last layer and Binary Cross Entropy as the loss function.

5.2 Adding synthetic data to SGD

Table 6 shows the results from adding synthetic data to the training set. Here we use all of SME dataset for evaluation. `Synthetic-all` consists of all synthetic utterances whereas `Synthetic-Dis` is the subset of synthetic utterances taken from conversations which are disjoint from the ones used in SME dataset (test set). We see a significant increase in the performance upon adding just a few hundred synthetic utterances. The size of synthetic dataset is quite small when compared to the size of the SGD dataset which can inhibit the model from learning from the synthetic generations. Owing to this, we experiment with various sampling factors, where a sampling factor of k means we duplicate the synthetic dataset k times. As an example, an affirming utterance for the request “Do you want to setup okta mfa? I’d like to" gives no prediction when trained only on SGD, whereas SGD + `Synthetic-all` predicts AFFIRM. Similarly, a negating utterance to the same request “I’d like to skip" also gives no prediction for SGD, whereas SGD + `Synthetic-all` predicts NEGATE.

Table 7 shows how performance varies with the sampling factors. The performance increases with sampling factor up to a certain point after which it degrades, indicating that a balance between the SGD dataset and synthetic dataset is essential for good performance. More notably, we see that with adequate oversampling we can bridge the gap in performance between `Synthetic-dis` and `Synthetic-all` for NEGATE and bring F1 score for AFFIRM within 0.03 points.

5.3 Adding linguist data

Next, we check the performance upon adding a small amount of SME data to the training mix to get an idea of the gap between synthetic and human generated data. We use 20% of the SME data for training (`SME-train`) and use the remaining 80% for evaluation (`SME-test`). To have a fair comparison, we compare `SME-train` with `Synthetic-dis` since using `Synthetic-all` would have overlapping conversations with `SME-test`.

	SGD-test		SME	
	AFFIRM	NEGATE	AFFIRM	NEGATE
SGD-train	0.98	0.98	0.39	0.16
SGD-train, Synthetic-dis	0.98	0.98	0.71	0.46
SGD-train, Synthetic-all	0.98	0.98	0.76	0.62

Table 6: F1 scores for the Dialogue Act Recognition models with and without synthetic data

sampling factor	SGD-train, Synthetic-dis		SGD-train, Synthetic-all	
	AFFIRM	NEGATE	AFFIRM	NEGATE
1	0.71	0.46	0.76	0.62
2	0.73	0.51	0.78	0.65
4	0.75	0.53	0.79	0.69
8	0.76	0.58	**0.8**	**0.72**
16	0.76	0.62	0.78	0.71
32	0.77	0.64	0.77	0.69
64	0.77	0.66	0.76	0.71
128	**0.77**	**0.73**	0.77	0.71
256	0.76	0.71	0.75	0.69

Table 7: F1 scores for the Dialogue Act Recognition models with different oversampling factors applied to the synthetic training sets - bold text shows best performance

Table 8 shows that both with and without oversampling, using SME data performs better than synthetic data, especially for NEGATE. However, using both SME and synthetic data performs better than using just SME data, showing the value of augmenting human-generated data with synthetic data.

5.4 Filtered vs unfiltered data

So far we have used AFFIRM and NEGATE generations from various LLM prompts which have been vetted by humans. However, this approach is not scalable and we thus check the value of using synthetic generations without any human intervention.

We select 2 prompts for each AFFIRM and NEGATE which work the best according to human evaluation and take all generations from those prompts across all conversations. This covers more conversations but since we restrict the synthetic data to only 2 prompts per dialogue act, we end up with $1k$ (only 2 prompts each for AFFIRM and NEGATE) utterances as opposed to $1.9k$ (human annotated data from all prompts) earlier.

We report the results from the best oversam-

sampling factor	SGD-train, Synthetic-dis		SGD-train, SME-train		SGD-train, Synthetic-dis, SME-train	
	AFFIRM	NEGATE	AFFIRM	NEGATE	AFFIRM	NEGATE
1	0.73	0.54	0.69	0.78	0.86	0.84
2	0.75	0.58	0.77	0.83	0.88	0.88
4	0.77	0.58	0.81	0.86	0.88	0.87
8	0.78	0.62	**0.83**	**0.87**	**0.88**	**0.89**
16	0.79	0.67	0.80	0.86	0.86	0.86
32	0.79	0.66	0.80	0.83	0.88	0.87
64	0.79	0.68	0.80	0.84	0.85	0.84
128	**0.79**	**0.74**	0.77	0.80	0.83	0.84
256	0.79	0.72	0.75	0.79	0.83	0.83

Table 8: F1 scores for the Dialogue Act Recognition models with synthetic and SME data - bold text shows best performance

pling factor using filtered and noisy synthetic data in Table 9. With the disjoint synthetic dataset, noisy utterances give the same performance as filtered utterances. Using noisy utterances from just 2 good prompts we get significantly better performance than using filtered utterances across all prompts. This shows the importance of choosing good prompts for data generation. With careful prompt selection, LLMs can generate high quality data without the need for human intervention.

	sampling factor	AFFIRM	NEGATE
SGD-train, Synthetic-dis	256	0.77	0.73
SGD-train, Synthetic-dis-noisy	128	0.79	0.7
SGD-train, Synthetic-all	16	0.8	0.72
SGD-train, Synthetic-all-noisy	16	0.85	0.88

Table 9: F1 scores for the Dialogue Act Recognition models with filtered and unfiltered synthetic data. `Synthetic-dis`, `Synthetic-all` denote the filtered versions and `Synthetic-dis-noisy`, `Synthetic-all-noisy` denote the unfiltered versions

6 Conclusion

We present shortcomings of existing datasets utilized towards Dialogue Act Recognition, such as the Schema-Guided Dialogue (SGD) datasets and propose using LLMs to generate data for overcoming the issues. We find that data generated synthetically helps with generalization to new domains without the need for human labeling. Moreover, in presence of labeled domain data, the synthetically generated data complements the variety of forms and linguistic properties present in training data and improves performance for Dialogue Act Recognition.

We utilize OpenAI's GPT-3 Completions API to generate the synthetic data, and find some interesting general takeaways for †ext generation. We present our findings in detail in Section 4.3. Mainly, we find that 1) questions should end in a question mark; 2) instead of saying a question was posed by a bot, it is better to append “Bot:" to the beginning of an utterance; and 3) multiple, simple instructions work better than a single, long instruction.

We find that even a small number of synthetic generations which are more varied in forms lead to better generalizability and performance for dialogue act recognition. We detail the findings in Section 5. We find that adding synthetic data is helpful, especially once we are able to class balance with oversampling. Synthetic data also complements human generated well, and used together help with making a model more robust - we find that using a few good prompts for generation without filtering can perform as well as (or even better than) using multiple prompts with human filtering.

References

Ali Ahmadvand, Jason Ingyu Choi, and Eugene Agichtein. 2019. Contextual dialogue act classification for open-domain conversational agents. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 1273–1276.

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 239–251.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021.

Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

Kristy Boyer, Eun Young Ha, Robert Phillips, Michael Wallis, Mladen Vouk, and James Lester. 2010. Dialogue act modeling in a complex task-oriented domain. In *Proceedings of the SIGDIAL 2010 Conference*, pages 297–305.

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

Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. In *The 41st international acm sigir conference on research & development in information retrieval*, pages 225–234.

Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel. 2020. Guiding attention in sequence-to-sequence models for dialogue act prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7594–7601.

Sergio Grau, Emilio Sanchis, Maria Jose Castro, and David Vilar. 2004. Dialogue act classification using a bayesian approach. In *9th Conference Speech and Computer*.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric P Xing, and Zhiting Hu. 2021. Text generation with efficient (soft) q-learning. *arXiv preprint arXiv:2106.07704*.

Léo Jacqmin, Lina M Rojas-Barahona, and Benoit Favre. 2022. " do you follow me?": A survey of recent approaches in dialogue state tracking. *arXiv preprint arXiv:2207.14627*.

Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, and Sachindra Joshi. 2018. Dialogue act sequence labeling using hierarchical encoder with crf. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

Bill Noble and Vladislav Maraev. 2021. Large-scale text pre-training helps with dialogue act recognition, but not without fine-tuning. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 166–172.

Vipul Raheja and Joel Tetreault. 2019. Dialogue act classification with context-aware self-attention. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3727–3733.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Schema-guided dialogue state tracking task at dstc8. *arXiv preprint arXiv:2002.01359*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Tulika Saha, Saurabh Srivastava, Mauajama Firdaus, Sriparna Saha, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Exploring machine learning and deep learning frameworks for task-oriented dialogue act classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Maryam Tavafi, Yashar Mehdad, Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2013. Dialogue act recognition in synchronous and asynchronous conversations. In *Proceedings of the SIGDIAL 2013 Conference*, pages 117–121.

Dewayne Whitfield. 2021. Using gpt-2 to create synthetic data to improve the prediction performance of nlp machine learning classification models. *arXiv preprint arXiv:2104.10658*.