

Comparing Data Augmentation Methods for End-to-End Task-Oriented Dialog Systems

Christos Vlachos^{1,2}, Themos Stafylakis^{1,2,3}, Ion Androutsopoulos^{1,3}

¹Department of Informatics, Athens University of Economics and Business, Greece

²Omilias Natural Language Solutions Ltd. ³Archimedes/Athena RC, Greece

Abstract

Creating effective and reliable task-oriented dialog systems (ToDSs) is challenging, not only because of the complex structure of these systems, but also due to the scarcity of training data, especially when several modules need to be trained separately, each one with its own input/output training examples. Data augmentation (DA), whereby synthetic training examples are added to the training data, has been successful in other NLP systems, but has not been explored as extensively in ToDSs. We empirically evaluate the effectiveness of DA methods in an end-to-end ToDS setting, where a single system is trained to handle all processing stages, from user inputs to system outputs. We experiment with two ToDSs (UBAR, GALAXY) on two datasets (MultiWOZ, KVRET). We consider three types of DA methods (word-level, sentence-level, dialog-level), comparing eight DA methods that have shown promising results in ToDSs and other NLP systems. We show that all DA methods considered are beneficial, and we highlight the best ones, also providing advice to practitioners. We also introduce a more challenging few-shot cross-domain ToDS setting, reaching similar conclusions.

1 Introduction

Task-oriented dialog systems (ToDSs) aim to understand and fulfil a user’s goal in a particular application domain (e.g., booking tickets, restaurant reservations) through natural language conversation (Bang et al., 2023; Zhao et al., 2022). They traditionally include several modules to handle a variety of processing stages. A natural language understanding (NLU) module typically extracts structured information from each user’s utterance, usually in the form of slot-value pairs (e.g., for destination, travel date). A dialog state tracker (DST) keeps track of the information that has been revealed about the user’s goal (dialog state) throughout the conversation. A policy optimizer (PO) de-

terminates the system’s next action (e.g., request information for a particular unfilled slot of the dialog state, or recommend a restaurant). A natural language generation (NLG) module realises communication actions of the system in natural language. In spoken conversations, automatic speech recognition (ASR) and text-to-speech (TTS) modules are also included (Oesterreich et al., 2023), though here we consider only written conversations.

When the modules of a ToDS are trained separately, large volumes of annotated training data are needed, since each module requires its own input/output training examples. In end-to-end ToDSs (Wen et al., 2017; Lei et al., 2018), a single system is jointly trained to handle all (or most) of the processing stages, reducing (or at least harmonising) the kinds of annotated training data required, since each training example can be used by all (or most) of the modules (Deriu et al., 2021). Recent Transformer-based end-to-end ToDSs (Ham et al., 2020; Peng et al., 2021; Sun et al., 2022) aim to reduce the necessary training data further, by leveraging pre-training on unlabelled data. However, the best performance is still obtained by fine-tuning on domain-specific annotated training datasets (Hudeček and Dusek, 2023), which are costly to obtain, especially when moving to a new domain (e.g., from booking airline tickets to reserving restaurants). LLMs that promise adequate results in few- or even zero-shot settings (Radford et al., 2019; Brown et al., 2020) have been underexplored in ToDSs, and work along this direction has focused mostly on the DST module, with contradicting results (Hudeček and Dusek, 2023; Hu et al., 2022). Thus cost-effective methods to obtain or augment training sets for ToDSs are needed.

We experiment with data augmentation (DA), which adds synthetic training examples to an existing annotated (possibly small) training set. DA methods have shown promising results in various NLP tasks, e.g., text classification, text generation,

question answering (Feng et al., 2021; Shorten et al., 2021; Pappas et al., 2022), but have not been systematically compared in end-to-end ToDSs (Section 6). We simulate the case where the available annotated training set is limited, which is often the case with real-life ToDS applications, especially when moving to new application domains. We experiment with two end-to-end pre-trained Transformer-based TODSs, namely UBAR (Yang et al., 2021) and GALAXY (He et al., 2022b), and two English textual conversational datasets, MultiWOZ 2.0 (Budzianowski et al., 2018) and KVRET (Eric et al., 2017). We vary both the size of the initial annotated training set (e.g., 2%, 10%, 25% of MultiWOZ’s training set) and the expansion factor (generating one, two, or four synthetic examples from each original one, i.e., x2, x3, x5 expansion). We also introduce a more challenging *few-shot cross-domain setting*: in each iteration, we select a single domain as the target one, we remove from the training set all but 20 (few-shot) dialogs of the target domain, we generate synthetic training dialogs from the 20 dialogs, and evaluate only on test dialogs from the left-out target domain.¹

We compare eight DA methods that have shown promising results in ToDSs (mostly when training individual modules) and other NLP systems. The DA methods we consider are of three types. *Word-level* augmentation methods replace words of the original training examples by similar words, e.g., words that have similar word embeddings or words suggested by models that predict masked words. *Sentence-level* augmentation includes paraphrasing sentences of the original training data via back-translation (translating to another language and back), invoking systems trained specifically to perform paraphrasing, or by prompting Large Language Models (LLM); we also include in this type methods that rotate (swap) parts of the dependency tree of the original sentence. Finally, *dialog-level* augmentation methods exploit dialog-specific annotations of the original training examples (e.g., dialog states), which are available in ToDS datasets.

Our contributions are: (1) We conduct a systematic comparison of DA methods in an end-to-end ToDS setting, the largest (in terms of DA methods) comparison to date in this setting. (2) We show that substantial performance gains can be obtained with DA methods, even when using pre-trained models,

¹For example, the domains of MultiWOZ are: *train, taxi, hotel, restaurant, hospital, police* and *attraction*.

and we offer concrete advice to ToDS practitioners (Section 5). (3) We introduce and utilise a new few-shot cross-domain evaluation setting for DA approaches in ToDS, where we show that DA methods again boost performance.

2 Data Augmentation Methods

2.1 Word-level Augmentation

Word-level DA methods replace words of the original training examples by similar words, e.g., words with similar word embeddings or words suggested by models that predict masked words (Wei and Zou, 2019; Wang and Yang, 2015). We experiment with word substitutions using either Word2Vec (Mikolov et al., 2013) or RoBERTa (Liu et al., 2019). Apart from boosting performance, word-level DA can generate more synthetic instances per original one, compared to other DA types considered.

2.1.1 Word2Vec-based Word Substitution

We employ Google’s 300-dimensional embeddings, created using Word2Vec.² We exclude stop-words or delexicalised tokens (special tokens replacing slot values) as candidates for DA. For every word in each dialog utterance, we find the 10 most similar ones in terms of cosine similarity of word embeddings. Preliminary testing showed that this may actually deteriorate performance, as some word replacements create dialog states and system actions inconsistent with the original ones, in agreement with Mi et al. (2021), who note that swapping certain words may change the original semantics. Instead, for every experiment, we train the ToDS with the original training data (the percentage available per experiment). For every word position, we test every candidate substitution from the pool of 10 most similar ones selecting at random, by passing the augmented sentence to that model. If the model’s prediction (in terms of dialog state and system action) is different than the original one, the augmentation is discarded and the next word replacement from the pool is considered.

2.1.2 LM-based Word Substitution

Masking a word and using a language model (LM) to predict the masked token is a more advanced case of word substitution, previously used in numerous cases (Pantelidou et al., 2022; Kobayashi, 2018; Mi et al., 2021). We leverage the large version of an out-of-the-box RoBERTa (Liu et al., 2019).

²Available via Gensim (Rehurek and Sojka, 2010).

The list of candidate substitutions now comprises words that the LM deemed more likely, otherwise the method remains as in Word2Vec-based substitution, including the requirement that the dialog state and system action must remain the same.

2.2 Sentence-level Augmentation

Sentence-level augmentation includes paraphrasing (via back-translation, with systems trained specifically to perform paraphrasing, or by prompting LLMs) and fragment rotation (which moves parts of the dependency tree). These methods cannot generate as many synthetic training instances as word-level substitution, but their synthetic utterances are semantically closer to the original ones.

2.2.1 Back-translation

Back-translation has been used in many NLP tasks (Kumar et al., 2019; Huang et al., 2021; Federmann et al., 2019). It translates a piece of text (an utterance in our case) to a different (pivot) language and back to the original one. We use French as the pivot language and the Google translate API.³ As explained by Ding et al. (2022), it is important to retain the original names of entities, such as attraction names and addresses. Special tokens used for delexicalisation also face the same issue. To avoid wrongly back-translating these special tokens, we apply back-translation to each utterance only after delexicalising and replacing the special tokens with special numbers (preceded by '#'), which remain unchanged during translation. The special tokens are relexicalized after DA.

2.2.2 PEGASUS Paraphrasing

The utilisation of LMs for paraphrasing was beneficial in the ToDS work of Gao et al. (2020), Hou et al. (2018), and Quan and Xiong (2019). Following Sharma et al. (2022), we leverage an instance of the Transformer-based PEGASUS (Zhang et al., 2020a), fine-tuned for paraphrasing.⁴ We set PEGASUS to generate up to two paraphrased utterances, and randomly select one, an approach that allows us to generate multiple synthetic dialogs per training instance.⁵ For each synthetic utterance, we check whether the delexicalisable tokens are preserved. If not, the augmentation is not applied.

³More languages could be used to generate more data.

⁴huggingface.co/tuner007/PEGASUS_paraphrase

⁵More paraphrases can be generated, but most of them tend to be of lower quality and may lead to wrong dialog states.

2.2.3 Fragment Rotation

Cropping or rotating/flipping images is a popular label preserving DA approach in Computer Vision (Krizhevsky et al., 2017; Engstrom et al., 2019). An NLP analog of rotation was proposed by Şahin and Steedman (2018) and has also been tested in a ToDS setting (Louvan and Magnini, 2020, 2021). The main idea is to rotate (swap) in the dependency tree of a sentence, fragments (such as subjects/objects) around the root (usually a verb). We leverage Stanford’s dependency parser (Qi et al., 2020) to generate annotations for MultiWOZ 2.0, consistent with the Universal Dependencies 2.1 (Nivre et al., 2020) in Şahin and Steedman’s work. We then employ their method to generate synthetic utterances.⁶ This may generate multiple outputs per original training instance, from which we sample uniformly. This method is a better fit for more free word-order languages, but both in Louvan and Magnini (2020) and in our experiments, it leads to substantial improvements in English too.

2.2.4 LLM Paraphrasing

LLMs have shown impressive results in tasks they have not encountered during training (Brown et al., 2020; Ouyang et al., 2022). Here we leverage ChatGPT, GPT-3.5-turbo specifically, and prompt it to generate augmented data via paraphrasing. We limit ourselves to two synthetic utterances to minimise the LLM’s usage cost, randomly selecting one of the two, provided it includes all of the tokens that will be delexicalised. As a prompt we use a simple instruction and no in-context learning.⁷

2.3 Dialog-level Augmentation

Dialog-level DA methods exploit dialog-specific annotations that ToD datasets include (e.g., dialog states, system actions). These methods can generate more synthetic data than sentence-level DA methods, but fewer than word-level ones. Also, the dialog-level DA methods considered here assume that dialog turns (user utterance–system response pairs) that have the same dialog state convey the same information, which is not always the case; hence, they may produce erroneous data.

2.3.1 Dialog Tree

The first dialog-level DA method we test was originally proposed by Aksu et al. (2022). Based on

⁶<https://github.com/gozdesahin/crop-rotate-augment>

⁷More information can be found in Appendix A.

their approach, we create a tree structure containing dialogs that can be formed by combining the turns of the available training data in three steps. The first step is the creation of turn templates. Each template comprises a delexicalised turn, along with the current delexicalised dialog state (*Cds*) and the delexicalised dialog states of the previous (*Pds*) and next (*Nds*) turns of the dialog they originate from. If a template refers to the first dialog turn, its *Pds* is “Root”. If a template refers to the last turn, its *Nds* is “Leaf”. Let the set of templates created in the first step be $T = \{r, t_1, \dots, t_n\}$, where r is a special root template used in the second step.

In the second step, templates are linked in a tree structure under a certain condition. A template t_j can be linked with a template t_i as its child node, only if t_i ’s *Nds* matches t_j ’s *Cds* and t_j ’s *Pds* matches t_i ’s *Cds* (see Fig. 7 of Appendix D for an example). All templates t that appear at the beginning of a dialog are automatically assigned as children of the special root template r . The third step is surface realisation, where delexicalised slots are filled with slot values.

As the time complexity of this method increases exponentially in the number of dialogs, we sample only 50 dialogs from the original training set in the first step. We then create the tree and generate a new synthetic dialog by starting from the root and randomly selecting a child recursively until we reach a leaf. We then repeat, sampling again 50 dialogs (with replacement) from the original training set and generating another synthetic dialog.

2.3.2 Act-Response Substitution

Multi-Action Data Augmentation (MADA) is another DA approach based on dialog state matching, proposed by Zhang et al. (2020b). For every turn, they identify all other turns (of the original training dialogs) that have the same delexicalised dialog state and extract the system actions associated with the identified turns. They then train their ToDS using the original and extracted system actions per turn. Normally, a ToDS would encounter a single system action in each instance and Zhang et al. (2020b) observe that some system actions are more likely to be the ground truth actions, compared to others that may be correct as well. Instead, through MADA, the model encounters a more balanced set of system actions, helping it generate diverse responses during inference. In our case, along with valid system actions, we also store the corresponding system responses per turn, following the same

principle. Then, for each dialog turn we replace the original system action and response by sampling from the set created by the process discussed.

3 Experimental Setup

In all experiments, we use three different randomly sampled training subsets (e.g., 3 different 10% subsets) and report averages.

3.1 Main Experiments

MultiWOZ: For our main experiments, we use MultiWOZ, an English human-to-human conversational dataset produced by a Wizard-of-Oz experiment, which has been used extensively in previous work (Wu et al., 2019; Heck et al., 2020; Zhao et al., 2021; Kulhánek et al., 2021; Wu et al., 2023). MultiWOZ 2.0, the most commonly used version that we also use, includes dialogs spanning 7 domains. It includes 10,438 dialogs in total, of which 1,000 are reserved for validation and another 1,000 for testing. Most dialogs fall into two or more domains making it a challenging benchmark. Of the tasks supported by MultiWOZ 2.0 (e.g., DST, intent classification, action-to-response generation) we select the response generation (RG) task, which UBAR and GALAXY can handle end-to-end.

Evaluation measures: Following Budzianowski et al. (2018), we mainly report *Score* values for MultiWOZ. We also present *Inform*, *Success*, and *BLEU* results in the appendix. *Inform* is the percentage of dialogs where the system provided a correct recommendation based on the user’s constraints (e.g., cheap hotel). *Success* is the percentage of dialogs where the system satisfied the user’s constraints (*Inform* = 100%) and provided all the requested information (Budzianowski et al., 2018; Nekvinda and Dušek, 2021). *BLEU* (Papineni et al., 2002) measures the word n -gram similarity between the system and the gold response. Finally, $Score = \frac{1}{2}(Inform + Success) + BLEU$.

UBAR: For the MultiWOZ experiments, we use UBAR (Yang et al., 2021), a GPT2-based ToDS (Radford et al., 2019) that remains competitive as shown in recent studies (Nekvinda and Dušek, 2021; Sun et al., 2022), and is also easier (and much less expensive) to use compared to larger LLMs. UBAR is also based on the framework used by MADA (Zhang et al., 2020b), thus facilitating experimenting with dialog-level DA methods.

MultiWOZ with all DA methods: In a first set of experiments, we reduce the size of the MultiWOZ

training set to 850 dialogs ($\sim 10\%$ of the training set) and produce an extra synthetic dialog from each original one, doubling ($\times 2$) the training set.

MultiWOZ with best three DA methods: In a second set of experiments, we consider only the best three DA methods of the previous experiments, but we now consider generating one, two, or four synthetic dialogs from each original dialog ($\times 2$, $\times 3$, $\times 5$), using only $\sim 2\%$ (170), $\sim 10\%$ (850), or $\sim 25\%$ (2,125) of the MultiWOZ training dialogs.

Few-shot cross-domain experiments: Finally, we test the best three DA methods of the first set of experiments, now in a few-shot, cross-domain setting for the *restaurant*, *hotel* and *attraction* domains of MultiWOZ. As in the first set of experiments, we use 10% of the MultiWOZ training dialogs. Additionally, in three iterations, we leave one domain out of the training set apart from 20 (few-shot) dialogs, and we use only test (and validation) dialogs that include the left-out domain. We generate four synthetic dialogs for each few-shot dialog ($\times 5$) in these experiments, which was the augmentation size that worked best in the second set of experiments when there were very few training instances.

3.2 Additional Experiments

To ensure that our previous findings are not dataset- and model-specific, we repeat them using KVRET (Eric et al., 2017) and GALAXY (He et al., 2022b).

KVRET is also an English multi-domain dataset, the domains being *weather*, *navigation*, *schedule*. It includes 2,424 dialogs, with 302 being reserved for validation and 304 for testing. Contrary to MultiWOZ, each dialog belongs in a single domain.

Evaluation measures: Following He et al. (2022b), for KVRET we report *Score* (calculated using the same formula as in MultiWOZ), with slight modifications. *Match* is reported, which is equivalent to the *Inform* of MultiWOZ. Moreover, Lei et al. (2018) argues that the original *Success* rate favours recall, as a dialog would be considered successful if the model responded with every possible recommendation. Hence *Success F1* is used to balance precision and recall instead of *Success*.

GALAXY is a more recent ToDS, pretrained using UniLM (Dong et al., 2019), currently among the best ToDSs on KVRET (He et al., 2022a).

Experiments: We conduct similar experiments on KVRET as on MultiWOZ. The initial training sets (before DA) include $\sim 5\%$ (120), $\sim 10\%$ (250),

$\sim 25\%$ (600) dialogs from KVRET’s training set. We select the three best performing DA approaches on the 5% subset and conduct the same experiments using the 10% and 25% subsets, generating one, two, or four synthetic instances from each original one ($\times 2$, $\times 3$, $\times 5$). KVRET has an average dialog length of 2.5 turns, contrary to MultiWOZ’s 13.68. We exclude the two dialog-level DA methods from the KVRET experiments, since they had the worst performance in the first set of experiments on MultiWOZ. As each dialog belongs in a single domain, our cross-domain evaluation setting is also not applicable to KVRET.

4 Experimental Results and Analysis

4.1 Results of the Main Experiments

MultiWOZ with all DA methods: The results of the first set of experiments (all 8 DA methods, 10% of MultiWOZ’s training set, $\times 2$ expansion) are presented in Table 1. All DA methods lead to a substantial performance boost. The three highest scores are obtained with Word2Vec substitution, PEGASUS paraphrasing and fragment rotation, improving UBAR without DA (“10% training set”) by about 9 points. Surprisingly, fragment rotation is among the best DA method, even though English is not particularly free word-order (Section 2.2.3). The worst results come from the dialog-level DA methods, presumably because matching delexicalised dialog states introduces noise (Aksu et al., 2022).

MultiWOZ with best three DA methods: The results of the second set of experiments (best 3 DA methods, 2%, 10%, 25% of original training set, expansion by $\times 2$, $\times 3$, $\times 5$) are shown in Fig. 1. For each original training set size (horizontal axis), we show the *Score* (vertical axis) of each DA method (curves). To save space, we show scores using the best (per DA method and training set size) expansion size ($\times 2$, $\times 3$, $\times 5$). As expected, DA methods are more beneficial when the original annotated training set is limited (e.g. 2%, 10%), with much smaller improvements when more annotated data are available (25%). Also, when the original training set is very limited (2%), $\times 5$ expansion leads to the best results, but when the original training set is larger (10%, 25%), more conservative ($\times 3$ and $\times 2$, respectively) expansion is better, presumably because it becomes more important to avoid adding multiple variants of the same original training instance. Overall, the three best DA methods lead to

very similar performance boosts, in agreement with the results of the previous experiments (Table 1).

Few-shot cross-domain experiments: The results of these experiments are shown in Fig. 2. All DA methods led to substantial improvements in all three few-shot domains, increasing the *Score* by up to 8, 9 and 10 points for the *attraction*, *hotel* and *restaurant* domains, respectively. Contrary to previous experiments, here PEGASUS paraphrasing was more effective overall compared to the other two DA methods.⁸

DA Type	DA Method	Score
No DA	Full training set	105.1
	10% training set	75.88
Word-level	Word2Vec replace	84.69
	RoBERTa replace	80.11
Sentence-level	Back-translation (FR)	81.31
	PEGASUS	83.16
	Fragment rotation	84.59
	LLM paraphrase	81.48
Dialog-level	Dialog Tree	79.94
	Act-Response	80.78

Table 1: **MultiWOZ Score**, for all **8 DA methods**, using **10%** of the original training dialogs, producing a single synthetic dialog from each original one (**x2**).

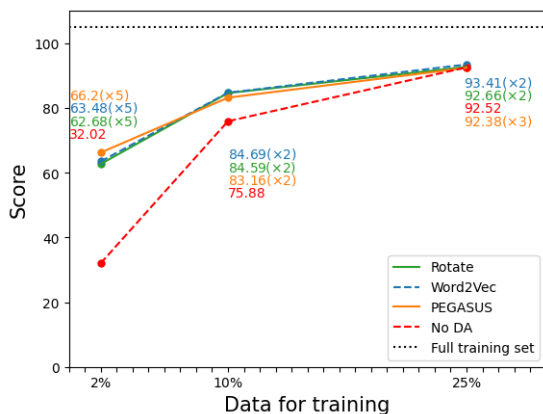


Figure 1: **MultiWOZ Score**, for the **3 best DA methods** of Table 1, using **2%**, **10%**, **25%** of the original training dialogs. We report in brackets the expansion size (**x2**, **x3**, **x5**) that led to the best *Score*.

4.2 Results of the Additional Experiments

KVRET with all DA methods: GALAXY’s results with all 6 applicable (to KVRET) DA methods, using 5% of KVRET’s training dialogs, and x2 expansion are reported in Table 2. Word2Vec substitution and fragment rotation remain among the

⁸More detailed results of the main and additional experiments can be found in Appendix B.

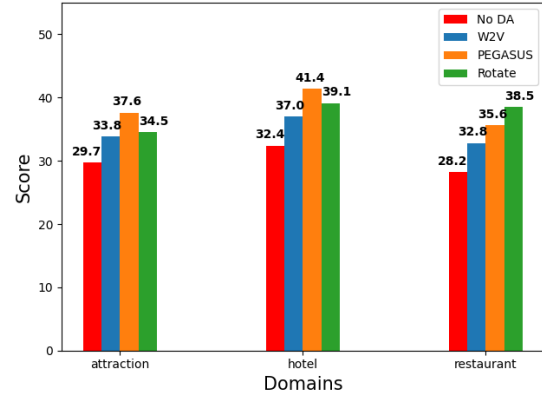


Figure 2: **MultiWOZ Score**, for the **3 best DA methods** of Table 1, in the **few-shot cross-domain** setting (domains: *attraction*, *hotel*, *restaurant*).

top performers, with the LLM-based paraphrasing marginally overtaking PEGASUS paraphrasing.

KVRET with best three DA methods: Figure 3 shows results obtained with the best three DA methods of the previous experiment, now using 5%, 10%, 25% of KVRET’s training set, and x2, x3, x5 expansion. Interestingly, DA (any method) manages to improve performance only when 5% of KVRET’s training set is used.⁹ Although KVRET is smaller than MultiWOZ (2,424 dialogs in total vs. 10,438) and one would expect DA methods to be more useful in KVRET, it is also much easier than MultiWOZ, because each dialog concerns a single domain. This is evident by the fact that GALAXY with only 25% of the KVRET training set and no DA (Fig. 3, rightmost result of the red dashed line) is almost as good as GALAXY trained on the entire KVRET training set, again with no DA (horizontal dotted line). Even with only 10% of KVRET’s training set, GALAXY with no DA (red dashed line) obtains a *Score* of 92.16, leaving very little scope for improvement. When using 5% of the original training set, where there is much larger scope for improvement (Fig. 3, left), all DA methods substantially improve performance, and the best DA method (fragment rotation) improves performance by approx. 9 points (Table 2). The results of Figures 1 and 3 also indicate that there is no universal threshold of initial annotated training instances, below which DA is beneficial; the threshold depends on the difficulty of the particular task (e.g., whether dialogs are single- or multi-domain). Instead, one should examine performance; when it is very low (e.g., with 2% of MultiWOZ or 5%

⁹This is also why we select the best three DA methods using 5% of KVRET’s training set in the previous experiment.

of KVRET), DA methods are definitely worth considering, otherwise when performance is already very high (e.g., exceeding 90 *Score* when using 10% or 25% of KVRET), applying DA is probably pointless. In all cases, generating more than one or two synthetic examples per original one (more than x2 or x3 expansion) is not beneficial.

DA Type	DA Method	Score
No DA	Full training data	102.5
	5% training data	73.27
Word-level	Word2Vec replace	81.44
	RoBERTa replace	80.67
Sentence-level	Back-translation (FR)	77.83
	PEGASUS	75.78
	Fragment rotation	82.76
	LLM paraphrase	82.04

Table 2: **KVRET Score**, for **6 applicable DA methods**, using **5%** of original training dialogs, producing a single synthetic dialog from each original one (**x2**).

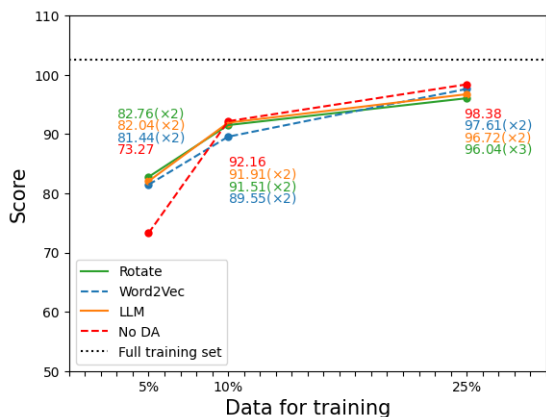


Figure 3: **KVRET Score**, for the **3 best DA methods** of Table 2, using **5%, 10%, 25%** of the original training dialogs. We report in brackets the expansion size (**x2**, **x3**, **x5**) that led to the best *Score*.

4.3 Response and Error Analysis

In MultiWOZ dialogs the domain may change from utterance to utterance making the benchmark much more challenging, compared to KVRET. Using DA approaches have proven beneficial in terms of the overall *Score*, but this is not always the case with the underlying metrics *Success* and *BLEU* score that concern the generated response as seen in Figures 3 - 23. To gain a better view of the effectiveness of the DA methods tested, we present an error rate per domain category based on *Success*. To do so, we calculate the percentage of unsuccessful dialogs (*Success* = 0) per dialog category in the test set. Consider for example the case of

using 25% data of the training set for DA leveraging the Word2Vec approach. A comparison of this approach when using x2 and x5 the amount of data is depicted in Figure 4. Additional comparisons for the case of PEGASUS and fragment rotations are presented in Appendix C. In the figure, “tx”, “at”, “re”, “ho” and “tr” refer to the *taxi*, *attraction*, *restaurant*, *hotel*, and *train* domains respectively. Multiple domains in each category denote domain switches. It comes as no surprise that the more domain switches a dialog contains, the more errors the model makes. Using x5 the amount of training data seems to amplify this behaviour; an observation that could be attributed to the additional noise introduced mostly in terms of the synthetically generated responses. Thus, generating more than a single synthetic dialog (especially in the x5 case) may not prove as beneficial for all cases (e.g. when 25% of the MultiWOZ dialogs are available). Overall, of the three best performing methods, Word2Vec and PEGASUS seem to perform similarly, making more errors in the same dialog categories (mostly the ones including the restaurant and hotel domains). On the other hand fragment rotation struggles more mostly with dialogs involving the attraction domain. One should keep in mind that, regardless of applying DA or not, many errors made by the model are also to be blamed to the automatic evaluation as during inference the user’s utterance remains static and is always in accordance with the ground truth system response, thus potentially confusing the model.

After careful, manual examination of system responses on a small sample of the test set across all experimental cases, the responses themselves are fluent and informative. Generally, it is very difficult to deduce which DA method was used to train the model that generated the response. The small differences in scores could be justified by this conclusion, especially when comparing two methods that perform DA at the same level (e.g. the sentence-level PEGASUS and fragment rotation methods). Even when generating multiple synthetic dialogs per training instance (e.g. x5 DA level), the responses remain fluent even though some DA methods may alter the structure of the sentence (e.g. Figure 13). Using fragment rotation to generate four synthetic dialogs is an exception to the above observation, as in few cases the resulting responses may also feature rotated fragments of the respective dependency trees. An example of a dia-

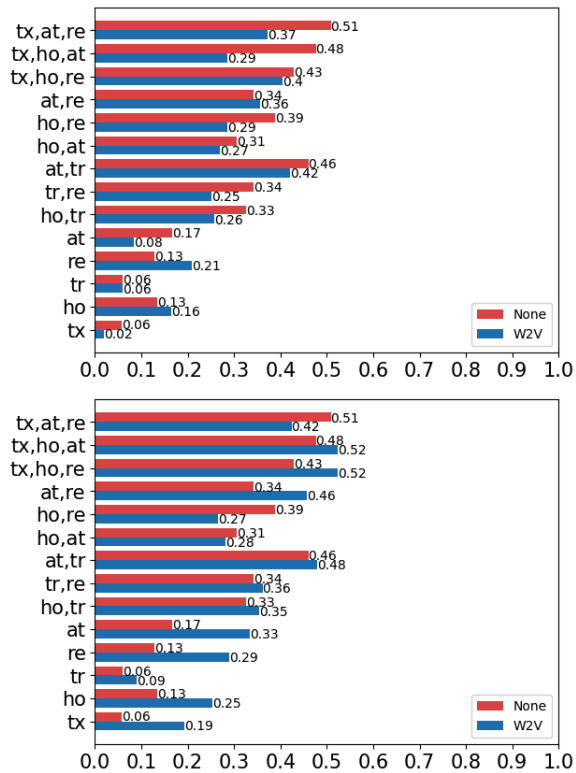


Figure 4: Error rates per MultiWOZ domain category using the Word2Vec DA method and having **25%** of the training data available. The upper and lower figures depict error rates when generating a single (**x2**) and four (**x5**) synthetic dialogs per instance respectively.

log and its corresponding response per DA method is depicted in Figure 17 of Appendix D. Moreover, a comparison of the models’ outputs when generating a single and four synthetic dialogs per training instance is depicted in Figure 18.

5 Advice to ToDS Practitioners

Based on our experiments, we provide the following advice to ToDS practitioners. (1) Although DA methods are currently not widely used in ToDS (Section 6), they can lead to very substantial gains when annotated training examples are scarce, even when using pre-trained end-to-end models. However, data scarcity depends on the difficulty of the task (e.g., whether the dialogues are single- or multi-domain), it cannot be judged using a universal threshold of initial annotated training instances, and is best assessed using performance scores. (2) Although all DA methods considered led to substantial improvements, dialog-state DA methods (at least the ones we considered) should be the last choices, as they led to a smaller performance boost and they also require dialog-specific annotations (e.g., dialog states) that are difficult to obtain in practice. (3) Among the other DA methods consid-

ered, the best overall performers were Word2Vec-based word substitution (Section 2.1.1), rotating parts of the dependency tree (Section 2.2.3), and paraphrasing the utterances by invoking paraphrasing models or prompting LLMs (Sections 2.2.2, 2.2.4). We recommend using any of these methods, provided that the resources they require are available; e.g., rotation requires a dependency parser for the target language, prompting requires an LLM supporting the target language and adds the cost of invoking an LLM. (4) Generating more than two synthetic instances per original one is worth considering only when the original annotated training instances are very scarce (e.g., when using only 2% of MultiWOZ’s multi-domain training set).

6 Related Work

DA has been applied to a wide range of tasks from Computer Vision (Krizhevsky et al., 2017; Engstrom et al., 2019) to Speech Processing (Ko et al., 2015) and NLP (Kumar et al., 2019; Huang et al., 2021; Wei and Zou, 2019; Kobayashi, 2018).

For NLP specifically, Pappas et al. (2022) carried out one of the most extensive comparisons of DA methods, in the context of biomedical QA, including Word2Vec-based word replacement (but without our semantics preserving test), LM-based word replacement, and back-translation. Word2Vec-based replacement was the most effective of the three methods, with back-translation and LM-based word replacement having similar performance.

Whitehouse et al. (2023) and Sahu et al. (2022) prompt off-the-self LLMs to produce synthetic data in cross-lingual commonsense reasoning and intent classification, respectively. Mehri et al. (2022) also leverage an LLM to produce synthetic data for intent classification, slot filling, and next action prediction in a zero-shot ToDS setting, showing improvements in performance. By contrast, Hudeček and Dusek (2023) reported that relying on LLMs to directly generate belief states and system actions was not beneficial, compared to a fine-tuned model.

Zhang et al. (2020b) propose the MADA (Multi-Action Data Augmentation) framework that we modified in order to include both the system actions and responses as explained in Section 2.3.2. Aksu et al. (2022) also capitalise on dialog states to produce a tree of possible dialogs (Section 2.3.1). Both of these approaches proved beneficial in our work. Similar to the approach of Aksu et al. (2022), Gritta et al. (2021) also capitalise on a graph struc-

ture, linking dialog states across all dialogs. In a similar spirit, Qiu et al. (2022) propose clustering dialog turns based on their dialog states. The extracted cluster structure forms the base for an augmentation approach similar to MADA.

Louvan and Magnini (2020, 2021), inspired by Şahin and Steedman (2018), introduce an NLP version of image cropping and rotation, based on permutations of sentence fragments (Section 2.2.3). To improve performance on out-of-scope (OOS) dialog utterances (not represented in the training set, e.g., pertaining to new domains), Chen and Yu (2021) augment their training set by mixing OOS utterances with the ones found in their training set. Yin et al. (2020) use Reinforcement Learning to find the optimal (in terms of context) text span replacements (instead of word replacements).

Paraphrasing as a form of DA was also used by Hou et al. (2018) and Gao et al. (2020) for slot-tagging (subpart of DST) and RG; they implemented their own paraphrasing model based on RNNs. Similarly, Axman et al. (2023) train a BART model (Lewis et al., 2020) for paraphrasing in an end-to-end ToD setting. Mi et al. (2021), also employ DA in a ToDS setting, by substituting masked tokens of the input using an LM (much as in Section 2.1.1). Song et al. (2021) augment their dataset by replacing slot-values with randomly generated strings to simulate references to out-of-vocabulary (OOV) words, a simple yet effective approach for models that learn to copy words from input to output by incorporating the copy-mechanism (Gu et al., 2016). Huang et al. (2022) randomly change the letter order of slot-values to improve the robustness of their ToDS. Improving robustness is also the main goal of Li et al. (2021), who generate turns with replaced slot-values not encountered in the training set.

The closest previous work is that of Quan and Xiong (2019). They experiment with four DA methods in an end-to-end ToDS setting, including stop-word removal, paraphrasing, back-translation, synonym substitution (and their combination). We compare more (eight) DA methods, of three different types (word-level, sentence-level, dialog-level), while Quan and Xiong consider fewer (four) methods of the word-level and sentence-level types only. Furthermore, we also experiment with a more challenging dataset containing dialogs that may belong (the same dialog) in multiple domains (MultiWOZ), whereas Quan and Xiong use the single-

domain Camrest676 (Wen et al., 2017), and (like us) KVRET, which covers three domains, but each dialog pertains to a single domain. Contrary to the work of Quan and Xiong, we also experiment with several sizes of initial annotated training set (e.g., 2%, 10%, 25% in MultiWOZ) and different expansions (x2, x3, x5), and we also introduce a new few-shot cross domain setting (Section 3.1).

7 Conclusions and Future Work

We performed the largest to date (in terms of methods and types) empirical comparison of DA methods in the context of end-to-end ToDSs, using two pre-trained Transformer ToDSs (UBAR, GALAXY) and two datasets (MultiWOZ, KVRET). We varied both the size of the initial annotated training set (e.g., 2%, 10%, 25% of MultiWOZ’s training set) and the expansion factor (generating one, two, or four synthetic instances from each original one). We also introduced a new, more challenging few-shot cross-domain evaluation setting. We showed that substantial performance gains can be obtained with DA methods, even when using pre-trained end-to-end models, and we offered concrete advice to ToDS practitioners regarding when to use DA, which DA methods to prefer, and with which expansion factors (Section 5).

In future work, we plan to explore combinations of DA methods, to check if the synthetic training instances they produce are complementary. We also plan to investigate how well DA methods perform in low-resource languages (e.g., languages for which LLMs or reliable dependency parsers may not be available). Finally, we aim to measure the costs (computational, monetary) of DA methods.

8 Limitations

A major limitation of our work is based on the nature of the datasets. Most of our DA methods rely on dialog annotations to ensure that they preserve the original semantics of the dialogs (Subsection 2.1.1) or to directly generate synthetic dialogs (Subsection 2.3.1). Many ToD datasets lack such detailed annotation, as is the case with the second track of DSTC 11 (Soltau et al., 2023).¹⁰ In real world scenarios such annotations are even harder to come by, even though there is a plethora of structured dialogs covering a wide range of topics. In

¹⁰The dataset can be found at <https://github.com/amazon-science/dstc11-track2-intent-induction>.

such cases, DA approaches that rely on the annotations of dialogs are not guaranteed to perform as well or at all on their own, compared to the experiments presented.

Another limitation of our work is the lack of detailed tuning/finetuning, in favour of a wider range of DA methods. Taking LLM-based paraphrasing as an example (Subsection 2.2.4), although we do experiment (up to a point) with different prompts during preliminary testing, a more detailed prompt tuning may have led to better results. In the same manner, for the case of PEGASUS paraphrasing (Subsection 2.2.2) or RoBERTa-based word substitution (Subsection 2.1.2), further specialised pre-training on other similar ToD datasets could have led to improved performance as shown by Wu et al. (2020). Similarly we avoid tuning the hyperparameters (e.g., learning rate, context window etc.) of the models used (UBAR and GALAXY). After all, the purpose of our work was not to achieve SOTA performance, but to compare DA methods and show their benefits under a common task (end-to-end).

Finally, we only take into account DA methods that can be applied regardless of the ToDS used. ToDSs such as the ones proposed by Gao et al. (2020) and Zhang et al. (2020b) consider online DA methods, that were not covered in our work. We recognise the advantages of such methods but opt not to include them as they are ToDS specific (they are build around a specific ToDS). Instead we experiment with offline DA methods that are architecture-free and have proven beneficial in ToD and NLP in general.

9 Ethical considerations

Implementing the DA methods and training the ToDS proposed in the settings mentioned in Section 3, requires a lot of GPU processing (excluding dialog-level DA and fragment rotation). Leaving aside the cost that comes as a result of the methods' execution, such GPU usage also has environmental impact in the form of CO2 emissions. We hope to alleviate this effect by offering advice to ToDS practitioners (Section 5) and guiding them in choosing the best fit (in terms of DA methods) for their task, in order to avoid such costly experimentation in the future.

10 Acknowledgements

This work was partially supported by Omilia Natural Language Solutions Ltd; the Research Center

of the Athens University of Economics and Business (AUEB-RC); and project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0, funded by the European Union under the NextGenerationEU Program. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission-EU. Neither the European Union nor the granting authority can be held responsible for them.

References

- Ibrahim Aksu, Zhengyuan Liu, Min-Yen Kan, and Nancy Chen. 2022. [N-shot learning for augmenting task-oriented dialogue state tracking](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1659–1671, Dublin, Ireland. Association for Computational Linguistics.
- Dustin Axman, Avik Ray, Shubham Garg, and Jing Huang. 2023. [Contextual data augmentation for task-oriented dialog systems](#). In *ECML-PKDD 2023 Workshop on Challenges and Opportunities of Large Language Models in Real-World Machine Learning Applications (COLLM)*.
- Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. [Task-optimized adapters for an end-to-end task-oriented dialogue system](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7355–7369, Toronto, Canada. Association for Computational Linguistics.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*, Red Hook, NY, USA. Curran Associates Inc.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. [MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.
- Derek Chen and Zhou Yu. 2021. [GOLD: Improving out-of-scope detection in dialogues using data augmentation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*,

- pages 429–442, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echevoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. [Survey on evaluation methods for dialogue systems](#). *Artif. Intell. Rev.*, 54(1):755–810.
- Bosheng Ding, Junjie Hu, Lidong Bing, Mahani Aljunied, Shafiq Joty, Luo Si, and Chunyan Miao. 2022. [GlobalWoZ: Globalizing MultiWoZ to develop multilingual task-oriented dialogue systems](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1639–1657, Dublin, Ireland. Association for Computational Linguistics.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. [Unified language model pre-training for natural language understanding and generation](#). Curran Associates Inc., Red Hook, NY, USA.
- Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. 2019. [A rotation and a translation suffice: Fooling CNNs with simple transformations](#).
- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. [Key-value retrieval networks for task-oriented dialogue](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.
- Christian Federmann, Oussama Elachqar, and Chris Quirk. 2019. [Multilingual whispers: Generating paraphrases with translation](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 17–26, Hong Kong, China. Association for Computational Linguistics.
- Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Edward Hovy. 2021. [A survey of data augmentation approaches for NLP](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.
- Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. [Paraphrase augmented task-oriented dialog generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649, Online. Association for Computational Linguistics.
- Milan Gritta, Gerasimos Lampouras, and Ignacio Iacobacci. 2021. [Conversation graph: Data augmentation, training, and evaluation for non-deterministic dialogue management](#). *Transactions of the Association for Computational Linguistics*, 9:36–52.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. [Incorporating copying mechanism in sequence-to-sequence learning](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.
- Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. [End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.
- Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022a. [Unified dialog model pre-training for task-oriented dialog understanding and generation](#). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22*, page 187–200, New York, NY, USA. Association for Computing Machinery.
- Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022b. [Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10749–10757.
- Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geischauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. [TripPy: A triple copy strategy for value independent neural dialog state tracking](#). In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.
- Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. [Sequence-to-sequence data augmentation for dialogue language understanding](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. [In-context learning for few-shot dialogue state tracking](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Dandan Huang, Kun Wang, and Yue Zhang. 2021. [A comparison between pre-training and large-scale back-translation for neural machine translation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1718–1732, Online. Association for Computational Linguistics.

- Tianjian Huang, Shaunak Ashish Halbe, Chinnadhurai Sankar, Pooyan Amini, Satwik Kottur, Alborz Geramifard, Meisam Razaviyayn, and Ahmad Beirami. 2022. [Robustness through data augmentation loss consistency](#). *Transactions on Machine Learning Research*.
- Vojtěch Hudeček and Ondřej Dusek. 2023. [Are large language models all you need for task-oriented dialogue?](#) In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.
- Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. [Audio augmentation for speech recognition](#). In *Proc. Interspeech 2015*, pages 3586–3589.
- Sosuke Kobayashi. 2018. [Contextual augmentation: Data augmentation by words with paradigmatic relations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. [Imagenet classification with deep convolutional neural networks](#). *Commun. ACM*, 60(6):84–90.
- Jonáš Kulháněk, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. 2021. [AuGPT: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models](#). In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 198–210, Online. Association for Computational Linguistics.
- Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. 2019. [Cross-lingual training for automatic question generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4863–4872, Florence, Italy. Association for Computational Linguistics.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. [Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. [Coco: Controllable counterfactuals for evaluating dialogue state trackers](#). In *International Conference on Learning Representations*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Samuel Louvan and Bernardo Magnini. 2020. [Simple is better! lightweight data augmentation for low resource slot filling and intent classification](#). In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 167–177, Hanoi, Vietnam. Association for Computational Linguistics.
- Samuel Louvan and Bernardo Magnini. 2021. [Simple data augmentation for multilingual nlu in task oriented dialogue systems](#). In *Proceedings of the Seventh Italian Conference on Computational Linguistics CLiC-it 2020*, pages 250–256, online. Torino: Accademia University Press.
- Shikib Mehri, Yasemin Altun, and Maxine Eskenazi. 2022. [LAD: Language models as data for zero-shot dialog](#). In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 595–604, Edinburgh, UK. Association for Computational Linguistics.
- Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. 2021. [Self-training improves pre-training for few-shot learning in task-oriented dialog systems](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1887–1898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). *arXiv preprint arXiv:1301.3781*.
- Tomáš Nekvinda and Ondřej Dušek. 2021. [Shades of BLEU, flavours of success: The case of MultiWOZ](#). In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46, Online. Association for Computational Linguistics.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. [Universal Dependencies v2: An evergrowing multilingual treebank collection](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

- Thuy Duong Oesterreich, Eduard Anton, Julian Schuir, Alexander Brehm, and Frank Teuteberg. 2023. [How can i help you? design principles for task-oriented speech dialog systems in customer service](#). In *Information Systems and e-Business Management*, pages 37–79.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Kyriaki Pantelidou, Despoina Chatzakou, Theodora Tsikrika, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2022. [Selective word substitution for contextualized data augmentation](#). In *Natural Language Processing and Information Systems*, pages 508–516, Cham. Springer International Publishing.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Dimitris Pappas, Prodromos Malakasiotis, and Ion Androutsopoulos. 2022. [Data augmentation for biomedical factoid question answering](#). In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 63–81, Dublin, Ireland. Association for Computational Linguistics.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2021. [Soloist: Building task bots at scale with transfer learning and machine teaching](#). *Transactions of the Association for Computational Linguistics*, 9:807–824.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Liang Qiu, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. [Structure extraction in task-oriented dialogues with slot clustering](#). *ArXiv*, abs/2203.00073.
- Jun Quan and Deyi Xiong. 2019. [Effective data augmentation approaches to end-to-end task-oriented dialogue](#). In *2019 International Conference on Asian Language Processing (IALP)*, pages 47–52.
- Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#).
- Radim Rehurek and Petr Sojka. 2010. [Software framework for topic modelling with large corpora](#).
- Gözde Gül Şahin and Mark Steedman. 2018. [Data augmentation via dependency tree morphing for low-resource languages](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.
- Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. [Data augmentation for intent classification with off-the-shelf large language models](#). In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.
- Saket Sharma, Aviral Joshi, Namrata Mukhija, Yiyun Zhao, Hanoz Bhathena, Prateek Singh, Sashank Santhanam, and Pritam Biswas. 2022. [Systematic review of effect of data augmentation using paraphrasing on named entity recognition](#). In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*.
- Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. [Text data augmentation for deep learning](#). *Journal of Big Data*, 8(1):101.
- Hagen Soltau, Izhak Shafran, Mingqiu Wang, Abhinav Rastogi, Wei Han, and Yuan Cao. 2023. [DSTC-11: Speech aware task-oriented dialog modeling track](#). In *Proceedings of The Eleventh Dialog System Technology Challenge*, pages 226–234, Prague, Czech Republic. Association for Computational Linguistics.
- Xiaohui Song, Liangjun Zang, and Songlin Hu. 2021. [Data augmentation for copy-mechanism in dialogue state tracking](#). In *Computational Science – ICCS 2021*, pages 736–749, Cham. Springer International Publishing.
- Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. [BORT: Back and denoising reconstruction for end-to-end task-oriented dialog](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2156–2170, Seattle, United States. Association for Computational Linguistics.
- William Yang Wang and Diyi Yang. 2015. [That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.
- Jason Wei and Kai Zou. 2019. [EDA: Easy data augmentation techniques for boosting performance on text classification tasks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. [A network-based end-to-end trainable task-oriented dialogue system](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.
- Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. [LLM-powered data augmentation for enhanced cross-lingual performance](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 671–686, Singapore. Association for Computational Linguistics.
- Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. [TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. [Transferable multi-domain state generator for task-oriented dialogue systems](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.
- Qingyang Wu, James Gung, Raphael Shu, and Yi Zhang. 2023. [DiactTOD: Learning generalizable latent dialogue acts for controllable task-oriented dialogue systems](#). In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 255–267, Prague, Czechia. Association for Computational Linguistics.
- Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. [Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14230–14238.
- Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. [Dialog state tracking with reinforced data augmentation](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9474–9481.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. [Pegasus: Pre-training with extracted gap-sentences for abstractive summarization](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML’20*. JMLR.org.
- Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020b. [Task-oriented dialog systems that consider multiple appropriate responses under the same context](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9604–9611.
- Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. [Description-driven task-oriented dialog modeling](#). *ArXiv*, abs/2201.08904.
- Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. [Effective sequence-to-sequence dialogue state tracking](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Appendix

A LLM prompting

We initially construct a simplistic template prompt, without giving any in-context example paraphrases, that has the following form : "*Paraphrase the following sentence twice. Maintain as much information as possible intact. The sentence to paraphrase is : { }*". The "{ }" symbol denotes the placeholder for the utterance (not delexicalised) to be paraphrased. More elaborate templates were also tested (e.g. providing the LLM with the dialog history), such as "*Your job is to augment the given utterance from a dialog between a user and an assistant. You will be given the dialog history as context. The user’s utterances start with USER :, while the assistant’s utterances with ASSISTANT :. Here is the dialog up to this point : {1} . Paraphrase the following sentence twice. Maintain as much information as possible intact. The sentence to paraphrase is : {2}*", where "{1}" and "{2}" denote the placeholders for the dialog history and current utterance respectively. During preliminary testing such templates did not provide a significant improvement, and as a result we opted for the more cost-efficient approach.

B More Detailed Results

In this section we present the detailed results of all the experiments conducted, including the additional metrics that accompany each dataset. For MultiWOZ, we include the *Inform* and *Success* score along with *BLEU* score, as explained in Subsections 3.1. Similarly for the case of KVRET the *Match*, *Success F1* and *BLEU* scores are reported. For each case, we first present the performance on the initial experiments where we select the 3 best performing DA methods followed by the results of these three methods in various other settings. From the results, we observe that the additional metrics reported tend to follow the trends of the main metric (*Score*), especially *BLEU* and *Success*.

DA type	DA Method	Inform	Success	BLEU	Score
No DA	Full training set	95.40	80.70	17.00	105.1
	10% training set	76.37	53.73	10.83	75.88
Word-level	Word2Vec replace	83.57	61.70	12.06	84.69
	RoBERTa replace	81.57	55.70	11.48	80.11
Sentence-level	back-translation (FR)	80.97	58.83	11.41	81.31
	PEGASUS paraphrase	83.33	60.87	11.06	83.16
	Fragment rotation	83.17	61.37	12.33	84.59
	LLM paraphrase	78.87	58.70	12.70	81.48
Dialog-level	Dialog Tree	78.83	57.23	11.90	79.94
	Act-Response substitution	79.33	58.90	11.66	80.78

Table 3: **MultiWOZ metrics**, for all **8 DA methods**, using **10%** of the original training dialogs, producing a single synthetic dialog from each original one (**x2**).

DA Method	Inform	Success	BLEU	Score
10% training set	76.37	53.73	10.83	75.88
Word2Vec replace	83.60	58.33	11.42	82.39
PEGASUS paraphrase	82.37	58.83	10.50	81.10
Fragment rotation	80.87	58.93	11.52	81.42

Table 4: **MultiWOZ metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing two synthetic dialogs from each original one (**x3**).

DA Method	Inform	Success	BLEU	Score
10% training set	76.37	53.73	10.83	75.88
Word2Vec replace	83.67	54.67	10.45	79.62
PEGASUS paraphrase	82.93	58.07	9.73	80.23
Fragment rotation	81.73	57.47	10.10	79.70

Table 5: **MultiWOZ metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**).

DA Method	Inform	Success	BLEU	Score
25% training set	88.00	69.73	13.65	92.52
Word2Vec replace	89.42	70.40	13.51	93.41
PEGASUS paraphrase	88.90	69.27	13.08	92.17
Fragment rotation	87.57	69.93	13.91	92.66

Table 6: **MultiWOZ metrics**, for the **top 3 DA methods**, using **25%** of the original training dialogs, producing a single synthetic dialog from each original one (**x2**).

DA Method	Inform	Success	BLEU	Score
25% training set	88.00	69.73	13.65	92.52
Word2Vec replace	89.77	70.33	13.13	93.18
PEGASUS paraphrase	89.00	71.03	12.37	92.38
Fragment rotation	88.17	69.60	13.18	92.06

Table 7: **MultiWOZ metrics**, for the **top 3 DA methods**, using **25%** of the original training dialogs, producing two synthetic dialogs from each original one (**x3**).

DA Method	Inform	Success	BLEU	Score
25% training set	88.00	69.73	13.65	92.52
Word2Vec replace	88.73	66.97	12.78	90.63
PEGASUS paraphrase	88.20	66.13	10.92	88.08
Fragment rotation	88.37	67.63	11.67	89.67

Table 8: **MultiWOZ metrics**, for the **top 3 DA methods**, using **25%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**).

DA Method	Inform	Success	BLEU	Score
2% training set	37.20	15.60	5.62	32.02
Word2Vec replace	57.87	7.13	1.95	34.45
PEGASUS paraphrase	73.33	14.20	2.82	46.59
Fragment rotation	62.40	12.10	3.23	40.53

Table 9: **MultiWOZ metrics**, for the **top 3 DA methods**, using **2%** of the original training dialogs, producing a single synthetic dialog from each original one (**x2**).

DA Method	Inform	Success	BLEU	Score
2% training set	37.20	15.60	5.62	32.02
Word2Vec replace	73.93	29.93	6.27	58.20
PEGASUS paraphrase	73.50	34.50	7.13	61.13
Fragment rotation	70.37	32.43	7.14	58.54

Table 10: **MultiWOZ metrics**, for the **top 3 DA methods**, using **2%** of the original training dialogs, producing two synthetic dialogs from each original one (**x3**).

DA Method	Inform	Success	BLEU	Score
2% training set	37.20	15.60	5.62	32.02
Word2Vec replace	75.17	37.03	7.38	63.48
PEGASUS paraphrase	75.50	41.70	7.60	66.20
Fragment rotation	71.83	39.03	7.25	62.68

Table 11: **MultiWOZ metrics**, for the **top 3 DA methods**, using **2%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**).

DA Method	Inform	Success	BLEU	Score
10% training set	28.37	13.72	8.61	29.65
Word2Vec replace	33.16	17.17	8.63	33.80
PEGASUS paraphrase	38.30	19.53	8.64	37.56
Fragment rotation	33.25	17.93	8.90	34.49

Table 12: **MultiWOZ metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**) and leaving out the **attraction** domain.

DA Method	Inform	Success	BLEU	Score
10% training set	32.15	17.43	7.62	32.41
Word2Vec replace	36.21	22.17	7.77	36.95
PEGASUS paraphrase	40.69	26.06	7.97	41.35
Fragment rotation	37.06	24.62	8.26	39.10

Table 13: **MultiWOZ metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**) and leaving out the **hotel** domain.

DA Method	Inform	Success	BLEU	Score
10% training set	28.37	13.65	7.17	28.19
Word2Vec replace	33.41	16.55	7.78	32.76
PEGASUS paraphrase	34.55	19.60	8.51	35.59
Fragment rotation	37.61	22.20	8.55	38.45

Table 14: **MultiWOZ metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**) and leaving out the **restaurant** domain.

DA Type	DA Method	Match	Success F1	BLEU	Score
No DA	Full training set	81.58	81.67	20.86	102.48
	5% training set	58.68	64.99	11.44	73.27
Word-level	Word2Vec replace	61.75	73.62	13.76	81.44
	RoBERTa replace	61.05	73.75	13.27	80.67
Sentence-level	back-translation (FR)	60.00	72.39	11.64	77.83
	PEGASUS paraphrase	56.49	69.81	12.63	75.78
	Fragment rotation	64.04	73.71	13.89	82.76
	LLM paraphrase	64.74	72.57	13.39	82.04

Table 15: **KVRET metrics**, for all **6 DA methods**, using **5%** of the original training dialogs, producing a single synthetic dialog from each original one (**x2**).

DA Method	Match	Success F1	BLEU	Score
5% training set	58.68	64.99	11.44	73.27
Word2Vec replace	54.74	75.41	12.83	77.91
Fragment rotation	56.05	74.98	11.93	77.44
LLM paraphrase	57.89	71.87	10.82	75.70

Table 16: **KVRET metrics**, for the **top 3 DA methods**, using **5%** of the original training dialogs, producing two synthetic dialogs from each original one (**x3**).

DA Method	Match	Success F1	BLEU	Score
5% training set	58.68	64.99	11.44	73.27
Word2Vec replace	56.05	74.25	9.58	74.73
Fragment rotation	57.11	67.00	10.37	72.42
LLM paraphrase	53.16	74.10	11.35	74.97

Table 17: **KVRET metrics**, for the **top 3 DA methods**, using **5%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**).

DA Method	Match	Success F1	BLEU	Score
10% training set	74.21	76.69	16.70	92.16
Word2Vec replace	68.42	79.00	15.84	89.55
Fragment rotation	73.94	78.28	15.40	91.51
LLM paraphrase	74.47	77.42	15.96	91.91

Table 18: **KVRET metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing a single synthetic dialog from each original one (**x2**).

DA Method	Match	Success F1	BLEU	Score
10% training set	74.21	76.69	16.70	92.16
Word2Vec replace	71.84	76.10	13.66	87.63
Fragment rotation	72.90	77.93	13.82	89.22
LLM paraphrase	73.94	79.48	13.81	90.53

Table 19: **KVRET metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing two synthetic dialogs from each original one (**x3**).

DA Method	Match	Success F1	BLEU	Score
10% training set	74.21	76.69	16.70	92.16
Word2Vec replace	69.73	77.16	13.37	86.81
Fragment rotation	71.58	77.17	13.79	88.16
LLM paraphrase	72.10	75.20	11.41	85.06

Table 20: **KVRET metrics**, for the **top 3 DA methods**, using **10%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**).

DA Method	Match	Success F1	BLEU	Score
25% training set	78.94	80.25	18.79	98.38
Word2Vec replace	78.95	79.61	18.32	97.61
Fragment rotation	77.11	78.16	18.09	95.72
LLM paraphrase	80.00	80.38	16.54	96.72

Table 21: **KVRET metrics**, for the **top 3 DA methods**, using **25%** of the original training dialogs, producing a single synthetic dialog from each original one (**x2**).

DA Method	Match	Success F1	BLEU	Score
25% training set	78.94	80.25	18.79	98.38
Word2Vec replace	76.31	80.63	17.75	96.22
Fragment rotation	78.42	79.77	16.95	96.04
LLM paraphrase	78.69	78.94	16.34	95.15

Table 22: **KVRET metrics**, for the **top 3 DA methods**, using **25%** of the original training dialogs, producing two synthetic dialogs from each original one (**x3**).

DA Method	Match	Success F1	BLEU	Score
25% training set	78.94	80.25	18.79	98.38
Word2Vec replace	75.53	78.14	14.91	91.75
Fragment rotation	76.58	78.68	15.19	92.81
LLM paraphrase	80.53	79.22	13.84	93.72

Table 23: **KVRET metrics**, for the **top 3 DA methods**, using **25%** of the original training dialogs, producing four synthetic dialogs from each original one (**x5**).

C Additional error comparisons

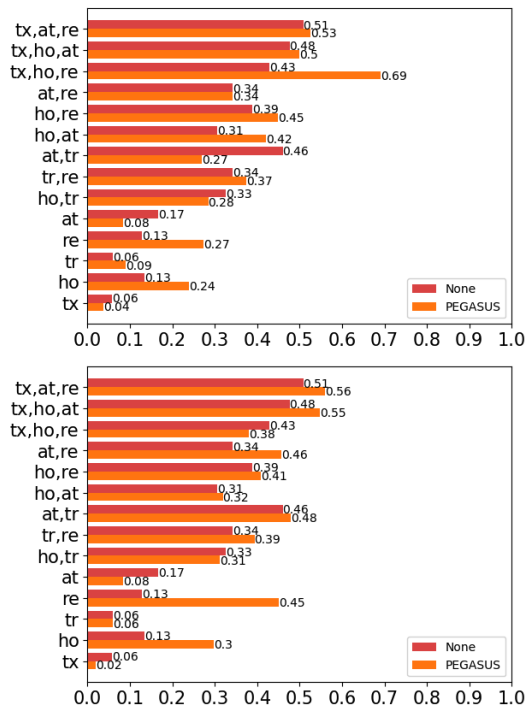


Figure 5: Error rates per MultiWOZ domain category using the PEGASUS DA method and having **25%** of the training data available. The upper and lower figure depict error rates when generating a single (**x2**) and four (**x5**) synthetic dialogs per instance respectively.

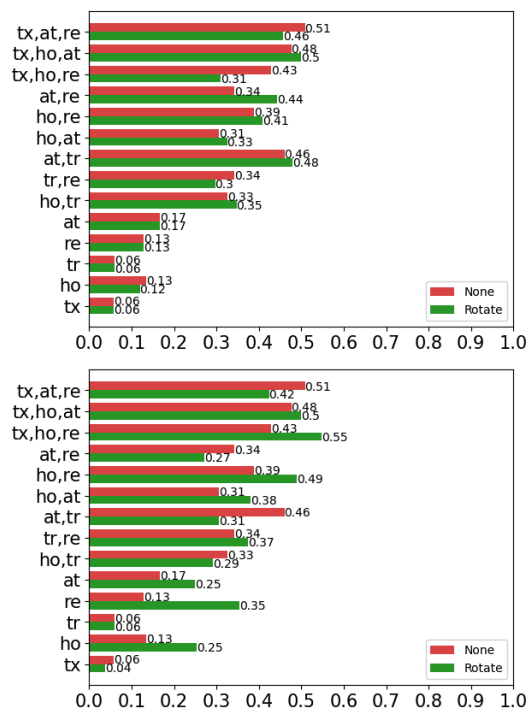


Figure 6: Error rates per MultiWOZ domain category using the fragment rotation DA method and having **25%** of the training data available. The upper and lower figure depict error rates when generating a single (**x2**) and four (**x5**) synthetic dialogs per instance respectively.

We follow the same procedure of Section 4.3 for the cases of PEGASUS and fragment rotation. Once again, we count the number of unsuccessful dialogs $Success = 0$ (i.e. the ToDS did not provide all the requested information or a valid recommendation), and divide by the number of dialogs belonging to each category, for the case where 25% of the training dialogs of MultiWOZ was used. We focus on $Success$ as it features greater variance compared to $BLEU$ and $Inform$, which generally remains unaffected. As was the case with Word2Vec, DA using PEGASUS or fragment rotation to generate more than one synthetic dialog reduces the overall $Success$. Interestingly there are a few instances where using multiple synthetic dialogs managed to increase the performance on specific, stand-alone domain categories such as “tx, ho, re” for PEGASUS and “at, tr” for fragment rotation, but such results may vary from case to case.

D Examples of synthetic and generated dialogs

In this section we present an example (Fig. 7) of the template matching performed by the dialog tree DA method (Subsection 2.3.1). In this case the Cds of the first turn of *pmul3825* matches the Pds of *mul0665*’s second turn. Similarly, *pmul3825*’s Nds matches *mul0665*’s Cds , thus setting *mul0665*’s turn as the continuation of *pmul3825*. Repeating the process until reaching a “Leaf” node (turn), creates a new synthetic dialog.

Figures 9-16 depict a single delexicalised dialog (*woz20572*) after applying each of the DA methods, for the case of MultiWOZ. An example of the same dialog, that UBAR is trained on, without any DA is depicted in Figure 8.

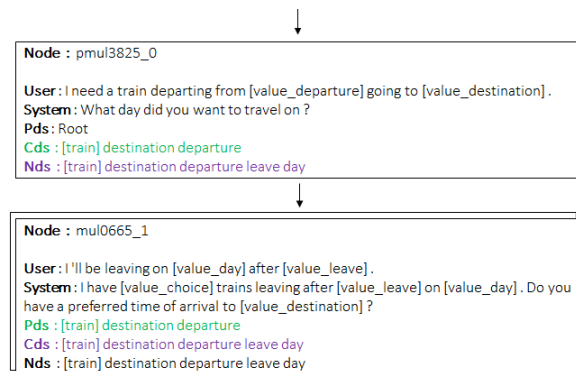


Figure 7: Two templates (from MultiWOZ dialogs *pmul3825* and *mul0665*) are linked as they satisfy the dialog state conditions (shown in green and purple).

USER: i am looking for an expensive restaurant in the south .
SYSTEM: hi ! we have [value_food], [value_food], [value_food], and [value_food]
restaurant-s that fit the bill . do you have any preferences ?

USER: i do n't have any preference .
SYSTEM: [value_name] is located in the [value_area] part of town and in the
[value_price] price range .

USER: do you have a phone number for them ?
SYSTEM: the number is [value_phone].

USER: have a good evening .
SYSTEM: have a good evening .

Figure 8: Dialog *woz20572* of **MultiWOZ**, as used by [Yang et al. \(2021\)](#) for training UBAR, without any DA applied.

USER: i am look for an expensive restaurant in the south .
SYSTEM: hello ! we have [value_food], [value_food], [value_food], and [value_food]
restaurant s that suited the legislation . do you have any preference ?

USER: i do ca have any preferred .
SYSTEM: [value_name] is situated in the [value_area] integral part of village and in the
[value_pricing] pricing ranging .

USER: do you have a telephone number for them ?
SYSTEM: the number is [value_phone].

USER: have a great night .
SYSTEM: have a great night .

Figure 9: A synthetic dialog derived from *woz20572* of **MultiWOZ** after applying our word substitution process via **Word2Vec**.

USER: i am searching for an expensive restaurant in the south .
SYSTEM: hello ! we have [value_food], [value_food], [value_food], and [value_food] restaurants that fill the bill . do you have any recommendations ?

USER: i do not have any preference .
SYSTEM: [value_name] is located in the [value_area] side of city and in the [value_price] price zone .

USER: do you have a contact number for them ?
SYSTEM: the phone is [value_phone] .

USER: have a great time .
SYSTEM: have a great day .

Figure 10: A synthetic dialog derived from *woz20572* of **MultiWOZ** after applying our word substitution process using **RoBERTa**.

USER: i am looking for a restaurant expensive in the south .
SYSTEM: hi ! we have [value_food], [value_food], [value_food] and [value_food] restaurant -s that match your expectations . do you have any preferences ?

USER: i have no preference .
SYSTEM: [value_name] is located in part [value_area] of the city and in the price range [value_price] .

USER: do you have a phone number for them ?
SYSTEM: the number is [value_phone] .

USER: have a good evening .
SYSTEM: have a good evening .

Figure 11: A synthetic dialog derived from *woz20572* of **MultiWOZ** after applying **back-translation**.

USER: i'm looking for an expensive restaurant in the south .
SYSTEM: do you prefer [value_food], [value_food], [value_food], and [value_food] restaurant -s ?

USER: i do n't have a preference .
SYSTEM: the [value_name] is located in the [value_area] part of town in the [value_price] price range .

USER: do you have a phone number for them ?
SYSTEM: the number is [value_phone] .

USER: have a great evening .
SYSTEM: have a great evening .

Figure 12: A synthetic dialog derived from *woz20572* of **MultiWOZ** after paraphrasing using **PEGASUS**.

USER: looking for an expensive restaurant in the south i .
SYSTEM: hi ! have [value_food], [value_food], [value_food], and [value_food] restaurant -s that fit the bill we . you have any preferences .

USER: have any preference i .
SYSTEM: [value_name] located in the [value_area] part of town and in the [value_price] price range .

USER: you have a phone number for them .
SYSTEM: the number is [value_phone].

USER: a good evening have .
SYSTEM: a good evening have .

Figure 13: A synthetic dialog derived from *woz20572* of **MultiWOZ** after applying **fragment rotation**.

USER: my quest is to find a costly restaurant located in the south .
SYSTEM: greetings ! we offer restaurant -s that serve [value_food], [value_food], [value_food], and [value_food] cuisine . do you have any specific preferences ?

USER: i have no particular liking or inclination .
SYSTEM: [value_name] can be found in the [value_area]ern area of the city and falls within the higher price bracket .

USER: is there a phone number available for them ?
SYSTEM: [value_phone] is the numerical sequence for the phone number .

USER: may your evening be enjoyable .
SYSTEM: may your evening be enjoyable .

Figure 14: A synthetic dialog derived from *woz20572* of **MultiWOZ** after paraphrasing using the GPT-3.5-turbo **LLM**.

USER: i am looking for an expensive restaurant in the south .
SYSTEM: we have some very nice restaurant -s , such as [value_name], offering [value_food] cuisine, [value_name], with [value_food], and the [value_name], with [value_food] fare . what sounds good to you today ?

USER: i do n't have any preference .
SYSTEM: alright , shall i make a reservation for the [value_name] ?

USER: do you have a phone number for them ?
SYSTEM: the number is [value_phone].

USER: have a good evening .
SYSTEM: have a good evening .

Figure 15: A synthetic dialog derived from *woz20572* of **MultiWOZ** after applying our modified **MADA** approach.

USER: i am looking for an expensive restaurant in the south .
SYSTEM: hi ! we have [value_food] , [value_food] , [value_food] , and [value_food] restaurant -s that fit the bill . do you have any preferences ?

USER: i want to eat thai food .
SYSTEM: there are n't any [value_food] restaurant -s in that price range in the [value_area] . would you like to try something else ?

USER: let me know about your italian restaurant -s .
SYSTEM: okay , i found [value_choice] options for you : [value_name] and [value_name] . do either of those appeal to you ?

USER: book the 1 in the centre a table for 2 at 13:45 on thursday
SYSTEM: i was able to book your table . your reference number is : [value_reference] .

USER: thank you . i will also need a hotel in that same area . please , no guest house !
SYSTEM: there are [value_choice] options for [value_type] in the [value_area] . there is [value_choice] [value_stars] star , 1 3 star and 1 [value_stars] star option . which would you prefer ?

USER: yes , please . for 2 nights .
SYSTEM: how many people will be staying with you and what day would you like to reserve your [value_type] room ?

USER: 2 for the same day . i need a taxi to commute between the 2 place -s to be sure it arrives by the booked time . the contact number and car type ?
SYSTEM: your room is booked , ref# [value_reference] , and a [value_car] will pick you up and get you to [value_destination] for that reservation . contact number [value_phone] .

USER: thanks , those are all i need for today .
SYSTEM: you are welcome . have a great day .

USER: you too have a great day
SYSTEM: thank you so much .

Figure 16: A synthetic dialog derived from the combination of *woz20572* and *pmul4652* of **MultiWOZ** using the **dialog tree** approach.

Turn 1

User : can you help me find a hotel called a and b guest house?

GT : it is in the [value_area] , and [value_price] . would you like to book a room ?

No DA : i have [value_choice] [value_type] in the [value_area] and [value_name] in the [value_area].

Word2Vec : [value_name] is a [value_price] accommodation in the [value_area]. would you like me to book you a room?

ROBERTa : i have [value_name] located in the [value_area] area. it is [value_price]. would you like me to book you a room?

BT : [value_name] is a [value_price] [value_type] in the [value_area] with [value_stars] stars. would you like me to book you a room?

PEGASUS : it is a [value_price] [value_type] in the [value_area]. it is free to enter. would you like me to book it?

RotatE : it is a [value_price] [value_stars] star [value_type] in the [value_area] with free parking and internet. would you like me to book it?

LLM : [value_name] is a [value_type] in the [value_area]. it is [value_price] and has [value_price] admission. would you like me to book you a room?

Dialog tree : [value_name] is a [value_type] in the [value_area], it is [value_price] and has [value_stars] stars. would you like me to book it?

MADA : [value_name] is a [value_type] in the [value_area] area of town. it is [value_price] and has internet. would you like me to book you a room?

Turn 2

User : can i get the address and phone number, please?

GT : the address is [value_address] , postcode [value_postcode] . the phone number is [value_phone] . anything else ?

No DA : the address is [value_address] and the phone number is [value_phone].

Word2Vec : their address is [value_address] and their phone number is [value_phone].

ROBERTa : the address is [value_address]. is there anything else i can help you with?

BT : the address is [value_address]. is there anything else i can help you with?

PEGASUS : the address is [value_address]. the phone number is [value_phone].

RotatE : the address is [value_address]. the phone number is [value_phone].

LLM : the address is [value_address] and the phone number is [value_phone]. is there anything else i can help you with?

Dialog tree : the address is [value_address] and the phone number is [value_phone].

MADA : the address is [value_address]. the phone number is [value_phone]. is there anything else i can help you with?

Turn 3

User : that is all. thank you.

GT : you are welcome have a great day .

No DA : you are welcome. goodbye.

Word2Vec : thank you goodbye

ROBERTa : you are welcome. have a great day!

BT : you are welcome. have a great day!

PEGASUS : you are welcome. goodbye.

RotatE : you are welcome. have a great day. goodbye.

LLM : thank you for using our system today!

Dialog tree : you are welcome, goodbye.

MADA : thank you goodbye

Figure 17: Dialog *sn01386* of **MultiWOZ** along with the generated responses using all the DA methods described. The models were trained using 10% of the training data and the augmentation level is **x2**. “GT” stands for ground truth while “BT” for back-translation.

Turn 0

User : can you help me find a hotel called a and b guest house?

GT : it is in the [value_area] , and [value_price] . would you like to book a room ?

No DA : i have [value_choice] [value_type] in the [value_area] and [value_name] in the [value_area].

Word2Vec x2 : [value_name] is a [value_price] accommodation in the [value_area]. would you like me to book you a room?

Word2Vec x5 : it is a [value_price] [value_type] in the [value_area]. would you like me to book it?

PEGASUS x2 : it is a [value_price] [value_type] in the [value_area]. it is free to enter. would you like me to book it?

PEGASUS x5 : they are a [value_price] [value_type] in the [value_area]. would you like to make a reservation?

Rotate x2 : it is a [value_price] [value_stars] star [value_type] in the [value_area] with free parking and internet. would you like me to book it?

Rotate x5 : happy to help! that is a [value_type] in the [value_area] and is considered [value_price]. would you like to make a reservation?

Turn 1

User : can i get the address and phone number, please?

GT : the address is [value_address] , postcode [value_postcode] . the phone number is [value_phone] . anything else ?

No DA : the address is [value_address] and the phone number is [value_phone].

Word2Vec x2 : their address is [value_address] and their phone number is [value_phone].

Word2Vec x5 : their address is [value_address] and their phone number is [value_phone].

PEGASUS x2 : the address is [value_address]. the phone number is [value_phone].

PEGASUS x5 : the address is [value_address] and the phone number is [value_phone].

Rotate x2 : the address is [value_address]. the phone number is [value_phone].

Rotate x5 : their phone number is [value_phone] and their address is [value_address].

Turn 2

User : that is all. thank you.

GT : you are welcome have a great day .

No DA : you are welcome. goodbye.

Word2Vec x2 : thank you goodbye

Word2Vec x5 : you are welcome, goodbye.

PEGASUS x2 : you are welcome. goodbye.

PEGASUS x5 : you are welcome, goodbye.

Rotate x2 : you are welcome. have a great day. goodbye.

Rotate x5 : you thank.

Figure 18: Dialog *sng01386* of **MultiWOZ** along with the generated responses using the 3 best performing DA methods and 10% of the available training dialogs. The figure depicts 2 responses per model (for x2 and x5 the amount of training data). “GT” stands for ground truth.