Building Adaptive Acceptability Classifiers for Neural NLG

Soumya Batra,[1] **Shashank Jain,**[1] **Peyman Heidari,**[1] **Ankit Arun**[1]
Catharine Youngs,[1] **Xintong Li,**[2] **Pinar Donmez,**[1] **Shawn Mei**[1]
Shiun-Zu Kuo,[1] **Vikas Bhardwaj,**[1] **Anuj Kumar,**[1] **Michael White**[1*]
[1]Facebook
[2]Ohio State University
{sbatra, shajain, peymanheidari, ankitarun, cathyoung}@fb.com
zncculee@gmail.com
{pinared, smei, skuo, vikasb, anujk, mwhite14850}@fb.com

Abstract

We propose a novel framework to train models to classify acceptability of responses generated by natural language generation (NLG) models, improving upon existing sentence transformation and model-based approaches. An NLG response is considered acceptable if it is both semantically correct and grammatical. We don't make use of any human references making the classifiers suitable for runtime deployment. Training data for the classifiers is obtained using a 2-stage approach of first generating synthetic data using a combination of existing and new model-based approaches followed by a novel validation framework to filter and sort the synthetic data into acceptable and unacceptable classes. Our 2-stage approach adapts to a wide range of data representations and does not require additional data beyond what the NLG models are trained on. It is also independent of the underlying NLG model architecture, and is able to generate more realistic samples close to the distribution of the NLG model-generated responses. We present results on 5 datasets (WebNLG, Cleaned E2E, ViGGO, Alarm, and Weather) with varying data representations. We compare our framework with existing techniques that involve synthetic data generation using simple sentence transformations and/or model-based techniques, and show that building acceptability classifiers using data that resembles the generation model outputs followed by a validation framework outperforms the existing techniques, achieving state-of-the-art results. We also show that our techniques can be used in few-shot settings using self-training.

1 Introduction

A key component of these models is a synthetic error generation step that applies various sentence transformations to some seed data. However, these simple transformations may not always be able to generate realistic error samples with respect to the NLG models. In this paper, we take an adaptive approach to synthetic data generation that employs a variety of model-based sentence transformations, some of which are additionally adaptive to the NLG models or dataset, in order to generate samples that better resemble the output of these models. We then pass these synthetic samples through a novel validation framework that filters and sorts them into acceptable and unacceptable classes, further improving the quality of the overall synthetic dataset. We show that an acceptability classifier built on top of the data generated by our approach improves upon existing techniques, and that we achieve state-of-the-art results by combining our adaptive data generation approaches with Harkous et al.'s non-adaptive ones.

2 Related Work

Work on automated evaluation metrics in the tradition of BLEU (Papineni et al., 2002) shares similar goals as our work, except that such metrics make use of reference sentences and thus are not designed for use at inference time. Moreover, such methods have not been found to correlate well with human evaluation of individual texts outside of the machine translation paradigm (Reiter, 2018). Çelikyilmaz et al. (2020) presents a comprehensive literature survey of the three broad categories of evaluation of the text generation models—human, automated and machine-learned—along with providing strong motivation for doing NLG evaluation.[1] Our approach is inspired by work in the third category of machine-learned evaluation.

[*]Work done while on leave from Ohio State University.

[1]See also Kryściński et al. (2019), who explore NLG evaluation in the specific NLG sub-fields of summarization and paraphrasing.

As noted, Harkous et al. (2020) improve upon earlier heuristic-based filtering by generating synthetic error data for training a semantic fidelity classifier. To do so, they use simple sentence transformations to create artificial omission, repetition, hallucination and value errors. However, since such transformations are not adaptive to NLG generation models the classifier is used with, they may not always produce the kind of unacceptable samples the corresponding NLG model would. Also related is Sellam et al.'s (2020) work on building a machine-learned scorer, BLEURT, to replace automated metrics such as BLEU. They use mask filling with a pretrained language model for creating synthetic unacceptable examples. In this paper, we introduce several new techniques for synthetic data generation, and comprehensively evaluate them in comparison to Harkous et al. (2020)'s methods, as well as to BLEU and BLEURT. In addition, we introduce a validation framework to sort the samples into the 2 classes. Our validation framework uses a pretrained entailment model, similarly to how Dušek and Kasner (2020) use one for semantic evaluation; here, we go beyond their approach by using it to develop an adaptive acceptability classifier that is better suited to runtime use.

As an alternative to using acceptability classifiers, one can make use of reconstruction models (Shen et al., 2019; Yee et al., 2019) to determine how well the NLG model's output predicts its input. These models are capable of detecting content errors but are not designed to capture grammatical mistakes. Additionally, since such approaches employ a second autoregressive decoding step, they are less well-suited to runtime inference in systems with tight latency budgets.

Regarding our self-training experiments, we note that self-training has been previously investigated for NLG by Kedzie and McKeown (2019), Qader et al. (2019) and Stevens-Guille et al. (2020), though they do not explore using pre-trained models with self-training. Also related are earlier approaches that use cycle consistency between parsing and generation models for automatic data cleaning (Nie et al., 2019; Chisholm et al., 2017). More recently, Chang et al. (2021) have developed a method for randomly generating new text samples with GPT-2 then automatically pairing them with data samples. By comparison, we take a much more direct and traditional approach to generating new text samples from unpaired inputs in self-training (He et al., 2020), using pre-trained models fine-tuned on the few-shot data for both generation and reconstruction filtering.

Dataset	#of samples	#of unacc.
WebNLG	2453	922
Cleaned E2E	500	250
ViGGO	500	394
Weather	493	83
Alarm	1470	102
Delexed WebNLG	500	176

Table 1: Test set statistics

3 Datasets

We conducted experiments on 5 datasets: WebNLG[2] (Gardent et al., 2017), Cleaned E2E (Dušek et al., 2019, 2020), ViGGO (Juraska et al., 2019), a Conversational Weather dataset created following the method described in Arun et al. (2020), and an Alarm dataset released by Arun et al. (2020). Further, in order to examine the effectiveness of self-training for building the acceptability classifier in the few-shot setting, we delexicalized (Arun et al., 2020) the meaning representations in the WebNLG dataset. For NLG models fed as inputs to our framework, we built an LSTM model for Alarm and finetuned BART (Lewis et al., 2020) models for others as per Arun et al.'s recommendations.

For test sets, we used (1) system outputs from the public human evaluation set for WebNLG 2017 and converted the labels to Acceptable class if both grammar and semantics ratings were greater than 2 (out of 3); (2) the Data Sabotaging strategy described in Section 5.1 to create model responses for Cleaned E2E, ViGGO and delexicalized WebNLG; and (3) responses generated by the Weather and Alarm NLG models for those datasets. We used these methods as a practical way to create a variety of errors in sufficient quantities to be able to effectively test the acceptability classifiers. Additionally, the Weather and Alarm test sets are representative of current SOTA models built for these domains. Human evaluations were done for all test sets except WebNLG to determine acceptability, using two annotators and a tie-breaker round in case of disagreement. The number of samples in all human annotated test sets can be found in Table 17.

[2]Obtained under CC BY-NC-SA 4.0 license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

4 Framework Design

In Figure 1, we show the overall design of our proposed framework. The framework takes in as input the training data of the text generation model as well as the trained generation model. The next step is the synthetic data generation that makes use of these inputs and is able to generate as many samples as needed. The synthetic samples are then passed through a validation framework that either sorts them into acceptable and unacceptable classes or rejects them altogether.

5 Synthetic Data Generation

Our synthetic data generation methods use the training data of the generation models (seed data) and the trained generation model. In Sections 9 and 10, we observe that a classifier built on data using our model-based and adaptive approaches improves upon the average F1 scores of standalone non-adaptive approaches by 1.1% to 18%. Following are the 4 strategies we introduce. Table 2 shows sample responses generated by each of these methods.

5.1 Data Sabotaging (SBTG)

We intentionally sabotage low-capacity LSTM models by only training them using 25% of the seed data to generate synthetic responses. These responses are more likely to be unacceptable with respect to the generation model responses as the full training data may contain considerably different inputs than the sabotaged one. We carry out this process 4 times with a different 25% sample of the training data and make predictions on the remaining 75% of the training data.

5.2 Noisy Beam Search (NBM)

We add random noise to beam scores at each inference step of the generation model. With this technique, the generated unacceptable responses tend to have grammatical errors, while the acceptable responses tend to be paraphrases having a different sentence structure compared to the seed responses.

5.3 Mask-Filling with vanilla BART (BART)

We insert 3 to 7 random masks in the seed data and use the vanilla BART (Lewis et al., 2020) model for filling in the masks. A small number of masks tends to produce acceptable data whereas a large number of masks tends to produce unacceptable semantically incorrect but grammatical data. This approach generates out-of-domain data (OOD).

5.4 Mask-Filling with fine-tuned BART (FTB)

We improve upon the OOD distribution limitation of vanilla BART by fine-tuning BART on noised sequences from seed data to reconstruct the original sequences. Denoising responses helps capture similar patterns in the seed data and masked words in the response are replaced by tokens most similar to that in seed data. We obtained best results by noising seed data using an insert mask ratio of 0.3 and random mask ratio of 0.5 where we mask the whole word. We use the same masking parameters to generate synthetic responses by mask-filling.

6 Validation Framework

The validation framework takes in the synthetic samples generated by above described methods and filters and sorts them into acceptable and unacceptable classes. Our experiments in Section 9 show that using a validation framework improves Macro F1 scores across all models by 1.4% to 5%. Following are the techniques we introduce.

6.1 Reconstruction Model Validator (REC-VAL)

We use this technique solely for the data sabotaging synthetic data generation method. In this approach, we use all the seed data to fine-tune BART (Lewis et al., 2020) as a reverse model with model responses as input and model input as the output. We then feed the synthetic responses to this reconstruction model and obtain the model inputs. Finally, we partitioned samples into acceptable and unacceptable classes based on whether they had the exact reconstruction match or not, respectively.

6.2 Entailment Model Validator (ENT-VAL)

For each seed response, we create a pair of the seed response and the generated synthetic sample. Next, we pass this pair twice to a RoBERTa-based entailment model (Liu et al., 2019) to obtain {entailment, neutral, contradiction} labels in both directions. The synthetic sample is sorted as acceptable if there is 2-way entailment within specified confidence thresholds (set heuristically for all domains through initial experimentation). Otherwise, the sample is sorted as unacceptable if the confidence score is within specified thresholds for the neutral or contradiction class in either direction. If none of the conditions are met, the sample is rejected.

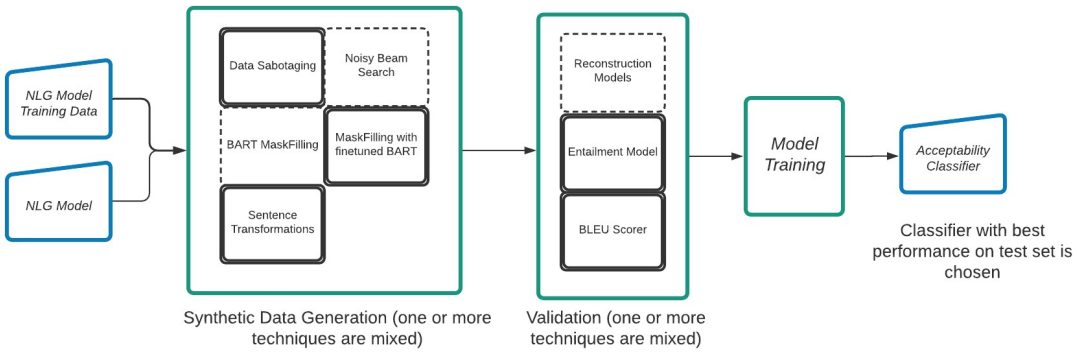

Figure 1: Overall Framework Design. Run multiple times with different data generation & validation strategy combinations. Best combination is indicated based on the 10-fold CV test set performance.

Ref	The 11th Mississippi Infantry Monument, established in 2000, can be found in Adams County Pennsylvania. The county is bordered to the north by Cumberland county and to the southeast by Carroll County.
SBTG (+)	the 11th Mississippi Infantry Monument **was** established in 2000 and **is located in** Adams County, Pennsylvania. **To the north of Adams County** is Cumberland County and Carroll County is to the southeast.
SBTG (-)	the 11th Mississippi Infantry Monument was established in 2000 in Adams County, Pennsylvania. It is located in Adams County, Pennsylvania, which has Carroll County to its southeast and Cumberland County **(PA) to its southeast**
NBM (+)	the 11th Mississippi Infantry Monument was **erected in the year** 2000. **Adams County** is bordered to the north by Cumberland County and to the southeast by Carroll County.
NBM (-)	**a 2000 monument** was erected in Adams County, Pennsylvania. Adams County is located to the **west** of Cumberland County, and **southwest** of Carroll County. **Adams County has Carroll County as its nearest county.**
FTB (+)	The 11th Mississippi Infantry Monument, established in 2000, can be found in Adams County, Pennsylvania. **Adams County** is bordered to the north by Cumberland County and to the southeast by Carroll County.
FTB (-)	**in** the Mississippi Infantry Monument, the monument was established in Adams County, Pennsylvania. this county is **west** of Cumberland County and **east** of Carroll County, **Maryland**.
BART (+)	11th Mississippi Infantry Monument **was** established in 2000, and **is located in** Adams County Pennsylvania, **the north of which** is Cumberland County and to **its** southeast is Carroll County.
BART (-)	11th Mississippi Infantry Monument was established in 2000 and is **located to the south** in Adams County Pennsylvania, the north of which is which is Cumberland County and to its southeast is Carroll County.

Table 2: Example Synthetic Acceptable(+) and Unacceptable(-) samples generated from a seed WebNLG response (Ref). The abbreviations follow the naming discussed in Section 9.

6.3 BLEU Score Validator (BLEU-VAL)

We calculate BLEU for a synthetic sample with respect to the original seed text. The sample is then sorted as acceptable or unacceptable when it lies within a specified range of BLEU scores.

7 Classifier Model Architecture

We formulate the task as binary classification with labels {Acceptable, Unacceptable} and learn a discriminator model using both the training data of the generator models and synthetic samples generated and refined using our adaptive approach. The training data consists of generation model input concatenated to the response (synthetic or original text) with a separator token. For the underlying model architecture, we use RoBERTa-Base (Liu et al., 2019).

8 Few-Shot Setting

Recently, there have been several efforts to train NLG models in a few-shot setting (Chen et al., 2020; Peng et al., 2020; Arun et al., 2020; Heidari et al., 2021). We adopt the self-training strategy introduced by Heidari et al. (2021) to generate training data for generative models, which is also used as the seed data needed for acceptability modeling. Self-training consists of several cycles of generation and reconstruction. For generation, we fine-tune BART (Lewis et al., 2020) using only 500 annotated examples to generate NLG responses given the input meaning representations (MRs). The same generation data is used to fine-tune a reconstruction BART model to obtain the input MR given the response. We use the reconstruction model to select samples with the exact reconstruction match after the generation step. At the end of the self-training cycles, we use all the

selected samples as seed data for acceptability modeling. Following Heidari et al. (2021), we delexicalize the meaning representations of the WebNLG dataset and pair them with existing delexicalized responses.

9 Results

In the following subsections, we compare the overall performance of classifiers. We report precision (P) and recall (R) of both acceptable (A) and unacceptable (U) classes. We also report Macro-F1 (F1) as the main metric we use to compare performance of the generation techniques. We use 10-fold cross validation (CV) to adjust the classification thresholds used for making predictions on the test sets. To calculate standard deviation of the F1 values, we use bootstrapping with 1000 rounds. Finally, we report the means over 3 runs for each technique. Further, we perform McNemar's statistical significance test comparing models trained with the best combination of methods with those trained with sentence transformations.

We use the following abbreviations to refer to the different synthetic data generation techniques: sentence transformation (SNT), data sabotaging (SBTG), noisy beam search (NBM), mask filling with vanilla BART (BART), mask filling with fine-tuned BART (FTB). Table 3 shows examples from 3 of the datasets where the acceptability classifiers capture unacceptable responses that pass the sentence transformation (SNT) baseline models. Our experiments show that fine-tuned BART (FTB) is often the best single method, so we include it in all the results along with its combination with sentence transformation (SNT+FTB). The results indicate that in the absence of a representative validation set, SNT+FTB should be used as it performs competitively across all datasets; otherwise, the validation set can be used to pick the best combination of techniques. We present comprehensive ablation experiments across all datasets in the appendix.

We performed nearest neighbor analysis (using BLEU) between test and synthetic unacceptable responses generated during training by our adaptive methods and sentence transformation. We found that across datasets, 52.27% to 98.48% unacceptable responses have a closest match to a sample generated by an adaptive method, suggesting that adaptive techniques produce more realistic samples compared to sentence transformation. In Table 4, we show sample unacceptable responses from 3 datasets with their closest match to a sample generated by each technique.[3]

9.1 Comparing Synthetic Data Generation Techniques

Tables 5–9 compare the performance of models trained with data generated from the mentioned techniques using a RoBERTa-based architecture. We also compare against the techniques described in Harkous et al. (2020), which we call sentence transformation (SNT). When data from different techniques including SNT are combined, we mix them in equal proportions. Additionally, we use all of the seed data as acceptable samples as well as ensure a 50:50 split between overall acceptable and unacceptable samples for training. We use different validation methods across these techniques: for BART and FTB we use a combination of BLEU and entailment models, for NBM we use entailment models, and for SBTG we use reconstruction models.

We observed that mask filling with fine-tuned BART performed consistently well across all datasets. We think this is because the pre-trained language model property of BART tends to generate grammatical responses, and by finetuning BART on the generation model training data, mask filling tends to generate words from the training data distribution, resulting in consistent generation of realistic samples. Noisy Beam Search tends to generate unacceptable samples that are ungrammatical (thus complementing fine-tuned BART technique) and hence, we see it included in the best combination for datasets where the test sets contain ungrammatical samples, such as WebNLG. Having said this, these are our initial observations and digging into more insights of the exact conditions under which different data generation techniques perform better is left for future research.

As can be seen in Figure 2, the mean macro-F1 score improves over the base sentence transformation in all 5 datasets. Note that macro-F1 for a majority class baseline is at best 50% since F1 for the minority class is always zero, and ranges from 33.3 to 48.2 for our test sets. Likewise, we built and tested against baseline supervised classifiers using available test data and 5-fold cross validation, and observed that our acceptability classifiers improve on the macro-F1 scores of these baseline classifiers by 5% to 27.3%.[3] We also performed

[3]More detailed comparisons can be found in appendix.

Dataset	Reference	Unacceptable Response
WebNLG	Rolando Maran , who was born in Italy , is the manager of a A. C . Chievo Verona	A.C. Chievo Verona is **voiced** by Rolando Maran, who was born in Italy.
Cleaned E2E	The Blue Spice pub located near Burger King has been rated average by customers.	Blue Spice is a **three star** pub with average **pricing**. it is located near the **river** to the burger king.
ViGGO	I know you're into role-playing games, so I wonder if you've tried the action-adventure RPG The Witcher 3: Wild Hunt.	I know you like **third person** action - adventure rpgs, have you played the witcher 3:wild hunt?

Table 3: Unacceptable responses caught by our acceptability classifier and missed by baseline SNT models.

Dataset	Unacceptable Response	Closest Match (ADP)	Closest Match (SNT)
ViGGO	what is it about tarsier studios' games that makes you find them fun?	what is it about tarsier studios' **rated** games that makes you find them fun?	what is it about **the pc games developed by good** that makes you find them fun?
Alarm	august 7th. for what time?	**july 5**th. for what time?	august **august**. for what time?
WebNLG (delexed, few)	agent-1 has patient-1 members and agent-1 ground is in patient-2 . agent-1 play in patient-3 and were in patient-4 .	agent-1 has patient-1 members and agent-1 ground is in patient-2 . agent-1 play in patient-3 and were in **patient-2** .	**agent-1 ground is in patient-2 and** agent-1 **have** patient-1 **members** . agent-1 play in patient-3 and were in **agent-1** .

Table 4: Closest matched (based on BLEU) synthetic unacceptable response in training data generated by adaptive techniques (ADP) and sentence transformations (SNT) with respect to a sample unacceptable response in test set.

Method	P(A)	R(A)	P(U)	R(U)	F1
SNT	79.8	81.3	72.8	70.8	76.2 ± 0.9
FTB	81.4	82.2	74.4	73.3	77.8 ± 0.9
BADP	81.6	84.2	76.5	73.0	**78.8 ± 0.9**
SNT+FTB	81.8	82.0	74.4	74.0	78.0 ± 0.9
SNT+ADP	81.3	83.0	75.2	72.9	78.1 ± 0.9

Table 5: Performance on **WebNLG** dataset. The best adaptive combination (BADP) is NBM+FTB and the best adaptive methods combined with sentence transformations (SNT+ADP) is NBM by itself.

Method	P(A)	R(A)	P(U)	R(U)	F1
SNT	62.0	69.1	65.2	57.5	63.1 ± 2.2
FTB	62.9	61.6	62.4	63.7	62.6 ± 2.2
BADP	64.6	60.8	62.9	66.6	63.6 ± 2.1
SNT+FTB	64.2	64.2	64.2	64.2	64.2 ± 2.2
SNT+ADP	68.0	62.1	65.1	70.7	**66.3 ± 2.1**

Table 6: Performance on **Cleaned E2E** dataset. The best adaptive combination (BADP) is NBM+BART+FTB and the best adaptive methods combined with sentence transformations (SNT+ADP) are NBM+FTB.

Method	P(A)	R(A)	P(U)	R(U)	F1
SNT	60.5	84.6	95.4	85.1	80.1 ± 2.0
FTB	64.6	70.8	91.9	89.6	79.1 ± 2.2
BADP	64.7	74.2	92.8	89.0	79.9 ± 2.1
SNT+FTB	66.2	78.8	94.0	89.2	**81.7 ± 2.1**
SNT+ADP	67.2	77.1	93.6	89.8	**81.7 ± 2.1**

Table 7: Performance on **ViGGO** dataset. The best adaptive combination (BADP) is BART+FTB and the best adaptive methods combined with sentence transformations (SNT+ADP) are SBTG+NBM+FTB.

Method	P(A)	R(A)	P(U)	R(U)	F1
SNT	90.4	94.3	64.6	50.6	74.4 ± 2.8
FTB	91.0	94.9	68.4	53.5	76.3 ± 2.7
BADP	91.0	96.0	73.4	53.4	77.5 ± 2.7
SNT+FTB	92.1	95.3	71.9	59.4	**80.2 ± 2.5**
SNT+ADP	91.9	95.4	72.5	58.3	78.8 ± 2.8

Table 8: Performance on **Weather** dataset. The best adaptive combination (BADP) is BART+FTB and the best adaptive methods combined with sentence transformations (SNT+ADP) are also BART+FTB.

McNemar's significance test comparing the best combination of techniques to the SNT version. The performance increase is statistically significant for the Weather, Alarm, and WebNLG datasets with p-values of 0.023, 0.004, and 0.006, respectively. Note that the Weather and Alarm datasets are representative of current SOTA models. The significance test of the Cleaned E2E was inconclusive with a p-value of 0.055, suggesting more samples are needed to make a determination. The improvement in ViGGO dataset is not significant.

We conjecture that this may be in part due to the proportion of acceptable vs. unacceptable samples in the Cleaned E2E and ViGGO test sets, along with differences in the datasets themselves. Since the other test sets have a relatively higher proportion of acceptable samples, they are likely to be more challenging for the classifiers, since the classifier needs to ensure complete parity between the meaning representation and generated output with

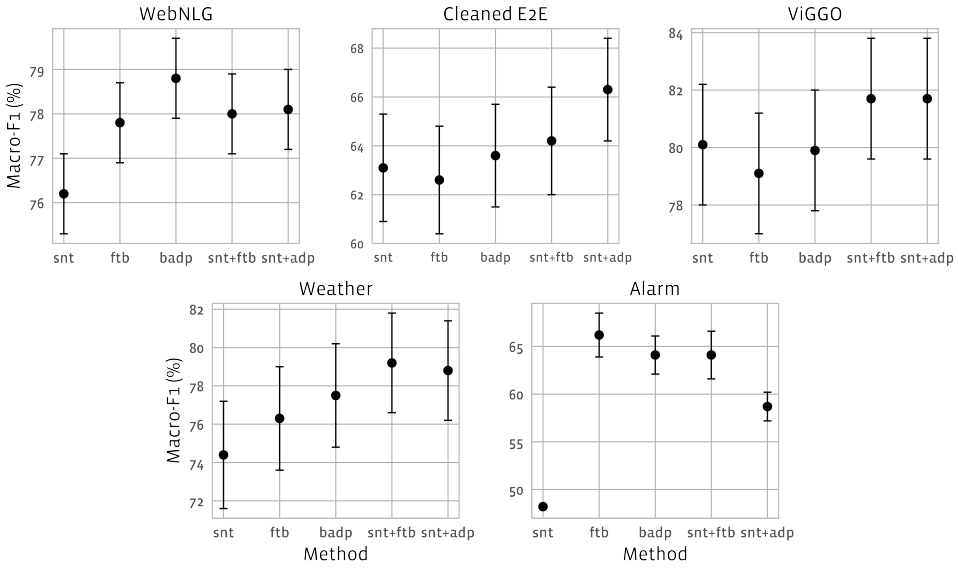

Figure 2: Comparing Macro-F1 scores of some combination techniques in the proposed framework with sentence transformations across 5 datasets. Notation: SNT = sentence transformations, FTB = fine-tuned BART, BADP = best adaptive combination, SNT+FTB = combination of sentence transformation and fine-tuned BART, SNT+ADP = combination of sentence transformation and best adaptive techniques. More metrics can be found in Tables 5–9.

Method	P(A)	R(A)	P(U)	R(U)	F1
SNT	93.1	99.9	0	0	48.2 ± 0.2
FTB	95.1	96.3	48.1	33.9	**66.2 ± 2.3**
BADP	95.6	91.6	29.5	43.6	64.1 ± 2.0
SNT+FTB	94.5	98.4	53.9	23.0	64.1 ± 2.5
SNT+ADP	99.4	71.8	21.5	93.8	58.7 ± 1.5

Table 9: Performance on **Alarm** dataset. The best adaptive combination (BADP) is SBTG+FTB and the best adaptive methods combined with sentence transformations (SNT+ADP) are NBM+BART+FTB.

the acceptable samples (which may be more difficult than spotting an error). Conversely, if the Cleaned E2E and ViGGO test sets are easier, then that could explain why it would be more difficult for a model to significantly improve on the sentence transformation baseline performance. Further analysis of this possibility is left for future work.

9.2 Effect of Validation Framework

Table 10 compares the performance of 4 validation strategies across the 5 datasets. We choose the best performing synthetic data generation technique combination for each dataset from Tables 5–9 and apply validation strategies on them for this comparison. The strategies ENT-VAL, BLEU-VAL and ENT+BLEU-VAL are applied only to the NBM, BART and FTB synthetic data generation methods. When SBTG is used, REC-VAL is used. Table 10 compares all validation strategies except REC-VAL (since it is always used when using SBTG data generation method) across the 5 datasets. As can be seen, not using any validation framework performs the worst across all 5 datasets, with 1.4% to 5% decrease in average F1 scores. Moreover, the best performing validation strategy includes the entailment model validator for 4 of the datasets.

Next, we compare the effect of adding synthetic acceptable data to the acceptability classifier's training data for the best performing synthetic data generation technique combination for each dataset. For all experiments, REC-VAL is applied for SBTG, ENT-VAL is applied for NBM, and ENT+BLEU-VAL is applied for both BART and FTB. As can be seen in Table 11, the average macro-f1 scores are improved by 0.4%-4.3% across all 5 datasets when adding synthetically generated acceptable data.

9.3 Few-Shot Setting

We delexicalized meaning representations of the WebNLG dataset and used the delexicalized version to build NLG models. We used 500 samples in the few-shot setting and auto-annotated 8,000 more through 2 cycles of self-training. We compared the performance of our few-shot acceptability classifiers with the full data ones, which were trained using more than 20,000 samples. As can be seen in Figure 3, there is no significant drop in the per-

Dataset	NO-VAL	BLEU-VAL	ENT-VAL	ENT+BLEU-VAL
WebNLG	75.6 ± 0.9	75.7 ± 0.9	**78.2 ± 0.9**	77.6 ± 0.9
Cleaned E2E	76 ± 2.4	78.1 ± 2.3	80.8 ± 2.1	**81.1 ± 2.2**
ViGGO	62.5 ± 2.2	**63.9 ± 2.2**	62.9 ± 2.2	63.4 ± 2.1
Weather	76.3 ± 2.8	70.1 ± 3	79.7 ± 2.5	**80.2 ± 2.5**
Alarm	68.4 ± 2.7	67.6 ± 2.8	**72.7 ± 2.4**	66.2 ± 2.3

Table 10: Comparing validation strategies across the 5 datasets

Dataset	no addition	with addition
WebNLG	77.4 ± 0.9	**78.8 ± 0.9**
Cleaned E2E	65.1 ± 2.1	**66.3 ± 2.1**
ViGGO	77.4 ± 2.3	**81.7 ± 2.1**
Weather	79.8 ± 2.6	**80.2 ± 2.5**
Alarm	64.5 ± 2.5	**66.2 ± 2.3**

Table 11: Effect of adding synthetic acceptable data (up to 30% seed data) on classifier Macro-F1%.

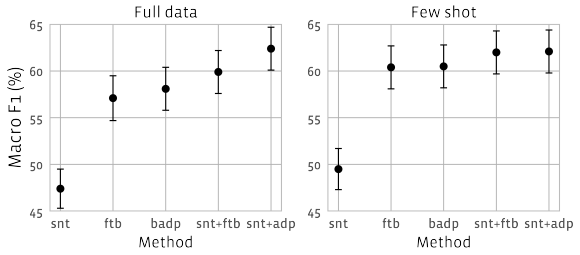

Figure 3: Comparing Macro-F1 scores of full data and few-shot acceptability classifiers using delexicalized WebNLG dataset. For the full data, the best adaptive combination (BADP) is NBM+FTB and the best adaptive methods combined with SNT (SNT+ADP) are SBTG + NBM + BART + FTB. For the few shot case, BADP is SBTG + NBM + BART + FTB and SNT+ADP is SBTG + FTB.

formance of the classifiers in the few-shot setting. Moreover, our adaptive methods work much better than sentence transformations on the delexicalized WebNLG dataset, including in the few-shot setting.

10 Comparison with Automated Metrics

Sellam et al. (2020) show that BLEURT-base achieves state-of-the-art consistency with human judgements on the WMT Metrics Shared Task (Bojar et al., 2017), and can be further fine-tuned on the WebNLG 2017 human ratings[4] to improve agreement. Since fine-tuned BLEURT checkpoints are not publicly available, we fine-tuned our own BLEURT models on human judgments of semantic adequacy (SEM) and grammatical correctness (GRAM) for WebNLG 2017, and of overall quality (E2E) for E2E human evaluations.[5] Specifically, we used all 5,363 items, sampling 1,000 of them as validation data, following BLEURT paper conclusions. Following Sellam et al. (2020), we stopped fine-tuning at 40,000 steps.

Metrics	WebNLG	Cleaned E2E	ViGGO
BLEU	45.4	48.7	54.7
BASE	47.4	49.3	46.9
SEM	51.9	49.5	49.5
GRAM	50.1	51.2	53.2
E2E	48.1	49.8	51.3
ACC	**78.8**	**66.3**	**81.7**

Table 12: Acceptability classifier (ACC) performance (macro F1%) vs. automated evaluation metrics. Notation: BLEU = BLEU score, BASE = BLEURT-base model, SEM and GRAM = our versions of BLEURT models fine-tuned on human judgments of semantic adequacy and grammatical correctness respectively for WebNLG, while E2E = our version of BLEURT model fine-tuned on overall quality metric for E2E human evaluations.

We obtained confidence thresholds for BLEU and all BLEURT based models by optimizing 10-fold cross validation Macro F1 scores as described in Section 9 at every 2 unit interval (step size of 0.02 for BLEURT and 2 for BLEU) between minimum and maximum BLEU/BLEURT scores. The threshold was then used to determine the predicted class. We show results in Table 12. As expected, BLEURT variants generally outperform BLEU. BLEURT fine-tuned on WebNLG outperforms BLEURT fine-tuned on E2E on WebNLG, and vice-versa for E2E, with an outlier outperformance of BLEU on ViGGO. Remarkably, our best acceptability classifier outperforms all BLEURT variants across all 3 datasets, despite BLEURT using reference sentences. This could be because BLEURT doesn't take the input into consideration, or because BLEURT is fine-tuned with a regression loss instead of a classification loss.

[4] https://gitlab.com/webnlg/webnlg-human-evaluation

[5] https://github.com/tuetschek/e2e-eval

11 Conclusion

In this paper, we introduced and analyzed several model-based and model-adaptive techniques, along with a validation framework, to create synthetic acceptable and unacceptable responses for training acceptability classifiers to filter outputs of neural NLG models. In addition, we compared and contrasted combinations of these techniques with using only the simple sentence transformation methods recently introduced by Harkous et al. (2020). We carried out a comprehensive study using 5 NLG datasets with varying levels of complexity and demonstrated that a combination of our methods and sentence transformations deliver state-of-the-art performance on all of them. Additionally, we demonstrated that using self-training, our models can be trained in few-shot settings without any significant drop in performance. This is especially important in light of recent efforts to develop few-shot NLG models, where avoiding semantic errors remains a central challenge. Finally, we recommend the strategy of using fine-tuned BART with the entailment model validator for building an acceptability classifier in the absence of a representative validation set. When such a set is available, we recommend performing ablation experiments across all combinations of different techniques using our framework in Section 4. Further analyzing the various conditions under which different synthetic data generation and validation strategies work with respect to the nature of underlying data is left for future work.

12 Ethical Considerations

The human annotators involved in the data evaluation for this paper are full time contracted employees. Before the data and evaluation guidelines are sent out to the annotators, the project goes through an approval process. The process starts by submitting a request containing human review workflow and guidelines according to project scope in layman's terms. Upon receiving the request, the the trained team automatically identifies risks based on the information contained in the request and assigns relevant reviewers. Subsequently, all potential risks are identified, documented and addressed before the start of the annotation process. This process ensures that the data and guidelines are designed to mitigate potential bias and risk. All of the guidelines and data used by this paper and sent to human annotators underwent this review process.

The classifiers laid out in this paper should only reduce the harms associated with models outputting semantically incorrect information, therefore reducing the risk of deploying such models. However, we would like to call out potential biases that may arise from training correctness models on a specific grammar. The grammatical evaluation done on the data uses prescriptive grammar of informal Standard American English. These prescriptive notions of grammaticality potentially serve to perpetuate systemic power imbalances as they're conveyed by language. The use of this grammar to train a correctness model may not be appropriate depending on the potential use case.

References

Ankit Arun, Soumya Batra, Vikas Bhardwaj, Ashwini Challa, Pinar Donmez, Peyman Heidari, Hakan Inan, Shashank Jain, Anuj Kumar, Shawn Mei, Karthik Mohan, and Michael White. 2020. Best practices for data-efficient modeling in NLG:how to train production-ready neural models with less data. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 64–77, Online. International Committee on Computational Linguistics.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021. Neural data-to-text generation with LM-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 758–768, Online. Association for Computational Linguistics.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online. Association for Computational Linguistics.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 633–642, Valencia, Spain. Association for Computational Linguistics.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Genera-*

tion, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137, Dublin, Ireland. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Peyman Heidari, Arash Einolghozati, Shashank Jain, Soumya Batra, Lee Callender, Ankit Arun, Shawn Mei, Sonal Gupta, Pinar Donmez, Vikas Bhardwaj, Anuj Kumar, and Michael White. 2021. Getting to production with few-shot natural language generation models. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 66–76, Singapore and Online. Association for Computational Linguistics.

Juraj Juraska, Kevin K. Bowden, and Marilyn Walker. 2019. Viggo: A video game corpus for data-to-text generation in open-domain conversation.

Chris Kedzie and Kathleen McKeown. 2019. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 584–593, Tokyo, Japan. Association for Computational Linguistics.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization.

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.

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.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679, Florence, Italy. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *In Proc. ACL-02*.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, Online. Association for Computational Linguistics.

Raheel Qader, François Portet, and Cyril Labbé. 2019. Semi-supervised neural text generation by joint learning of natural language generation and natural language understanding models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 552–562, Tokyo, Japan. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Comput. Linguist.*, 44(3):393–401.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text generation. 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 4060–4067, Minneapolis, Minnesota. Association for Computational Linguistics.

Symon Stevens-Guille, Aleksandre Maskharashvili, Amy Isard, Xintong Li, and Michael White. 2020. Neural NLG for methodius: From RST meaning representations to texts. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 306–315, Dublin, Ireland. Association for Computational Linguistics.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. 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 5696–5701, Hong Kong, China. Association for Computational Linguistics.

A. Çelikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *ArXiv*, abs/2006.14799.

A Appendix

A.1 Reproducibility

All the data and annotation guidelines for the experiments conducted in this paper will be shared post acceptance.

All experiments were conducted on 32GB Quadro GV100 GPUs. The Acceptability Classifiers were trained by optimizing roc-auc metric on the validation set. The average latency of the classifiers is 150ms.

Parameter	Value
tokenizer	BPE
tokenizer max length	1024
encoder output dropout	0.1
encoder embedding dim	768
#encoder layers	12
#encoder attention heads	12
decoder dropout	0
decoder activation	relu
Number of model params	124055810

Table 13: Parameters of RoBerta-Base

Method	Parameter	Value
NBM	beam size	5
	topk	5
	beta	1.1
BART	masking prop	0.7
	min num masks	3
	max num masks	7
	beam	5
FTB	beam size	5
	topk	3
	mask normal	0.5
	mask insert	0.3

Table 14: Parameters of Data Generation Models

Dataset	roc-auc
WebNLG	97.1
Cleaned E2E	99.3
ViGGO	99.0
Weather	99.8
Alarm	99.9
WebNLG(delex, full)	98.9
WebNLG(delex, few)	99.0

Table 15: ROC-AUC values of Winning Strategies on Validation Set

Synth gen	Class	Validator	Min Score	Max Score
FTB or BART	Acc	BLEU	0.95	0.99
FTB or BART	UnAcc	BLEU	0.55	0.95
FTB or BART	Acc	Entailment	0.95	1.0
FTB or BART	UnAcc	Entailment	0.5	1.0
NBM	Acc	Entailment	0.9	1.0
NBM	UnAcc	Entailment	0.7	1.0

Table 16: Validation Framework Parameters for Acceptability (Acc) and Unacceptability (UnAcc) classes.

A.2 Comparison of Acceptability Classifier with Baseline Classifiers

Dataset	Majority Class	Baseline Supervised	Acceptability
WebNLG	38.4	68.5 ± 1	$\mathbf{78.8 \pm 0.9}$
Cleaned E2E	33.3	50.4 ± 2.3	$\mathbf{64.2 \pm 2.2}$
ViGGO	44.1	56.3 ± 2.4	$\mathbf{81.7 \pm 2.1}$
Weather	45.4	52.9 ± 2.3	$\mathbf{80.2 \pm 2.5}$
Alarm	48.2	61.2 ± 1.5	$\mathbf{66.2 \pm 2.3}$
WebNLG(delex, full)	39.3	49.4 ± 2.2	$\mathbf{62.4 \pm 2.3}$
WebNLG(delex, few)	39.3	56.9 ± 2.4	$\mathbf{62.1 \pm 2.3}$

Table 17: Comparison of Macro F-1 scores between acceptability classifiers, Majority Class Baseline Classifier and Baseline Supervised Classifier trained and tested on available labelled data using 5-fold CV. Acceptability classifier improves upon both approaches.

A.3 Nearest Neighbor Analysis between Test and Synthetic Unacceptable Responses

Dataset	Match %(ADP)	Match %(SNT)
ViGGO	98.48	1.52
Weather	75.9	24.1
Alarm	59.8	40.2
WebNLG (delexed, full)	52.27	47.73
WebNLG (delexed, few)	57.96	42.04

Table 18: Percentage(%) of unacceptable samples having closest match (based on BLEU) to an unacceptable synthetic sample generated by adaptive techniques (ADP) vs. sentence transformations (SNT)

A.4 Ablation Results Across Techniques And Datasets

Below ablation results are on a single run. Top 10 winning strategies were selected and run 3 times to obtain the final results, as shown in main paper.

Dataset	Method	P(C)	R(C)	P(I)	R(I)	F1
WebNLG 2017	FTB	81.2	83.2	75.3	72.7	78.1 ± 0.9
	SNT	78.1	84.6	75.2	66.5	75.9 ± 0.9
	NBM	76.1	89.4	80.0	60.2	75.5 ± 1.0
	SBTG	74.9	87.3	76.5	58.6	73.5 ± 1.0
	BART	65.0	85.8	63.2	34.5	59.3 ± 1.1
	NBM+FTB	80.5	85.7	77.7	70.6	78.5 ± 0.9
	NBM+SNT	82.5	81.1	73.9	75.6	78.3 ± 0.9
	FTB+SNT	80.2	85.2	77.1	70.2	78.1 ± 0.9
	BART+FTB	80.7	83.0	74.9	71.8	77.6 ± 0.9
	SBTG+NBM	79.4	84.4	75.8	69.1	77.0 ± 0.9
	BART+SNT	79.2	84.4	75.6	68.5	76.8 ± 0.9
	SBTG+FTB	79.2	82.5	73.6	69.3	76.1 ± 1.0
	NBM+BART	76.7	88.5	79.1	61.9	75.8 ± 0.9
	SBTG+BART	77.4	84.7	74.9	64.9	75.2 ± 1.0
	SBTG+SNT	77.0	83.5	73.4	64.7	74.4 ± 0.9
	SBTG+NBM+SNT	80.3	85.2	77.0	70.5	78.1 ± 0.9
	NBM+BART+FTB	81.1	83.1	75.2	72.5	77.9 ± 0.9
	NBM+FTB+SNT	81.0	82.3	74.3	72.6	77.6 ± 0.9
	SBTG+NBM+FTB	79.6	85.7	77.3	68.7	77.6 ± 0.9
	BART+FTB+SNT	81.2	81.8	74.0	73.2	77.5 ± 0.9
	NBM+BART+SNT	78.1	87.5	78.5	65.1	76.8 ± 0.9
	SBTG+BART+FTB	80.4	81.5	73.2	71.9	76.8 ± 1.0
	SBTG+NBM+BART	79.0	84.7	75.9	68.1	76.8 ± 0.9
	SBTG+FTB+SNT	79.4	82.9	74.1	69.4	76.4 ± 0.9
	SBTG+BART+SNT	77.7	84.4	74.8	65.6	75.4 ± 1.0
	SBTG+NBM+BART+FTB	79.9	86.4	78.2	69.1	78.2 ± 0.9
	SBTG+NBM+FTB+SNT	81.5	81.3	73.6	73.9	77.6 ± 0.9
	NBM+BART+FTB+SNT	79.9	84.6	76.1	69.8	77.5 ± 0.9
	SBTG+BART+FTB+SNT	80.0	80.5	72.1	71.5	76.0 ± 0.9
	SBTG+NBM+BART+SNT	78.1	84.9	75.5	66.2	75.9 ± 0.9
	SBTG+NBM+BART+FTB+SNT	81.3	81.8	73.9	73.3	77.5 ± 0.9
Cleaned E2E	FTB	62.5	62.3	62.6	62.7	62.5 ± 2.1
	SNT	59.2	65.6	61.6	55.0	60.1 ± 2.2
	BART	56.2	83.6	67.9	34.7	56.5 ± 2.3
	SBTG	58.0	26.4	52.3	80.8	49.9 ± 2.2
	NBM	50.1	100.0	0.0	0.0	33.4 ± 1.0
	FTB+SNT	68.9	60.5	65.0	72.9	66.5 ± 2.1
	NBM+SNT	60.9	73.2	66.5	53.1	62.7 ± 2.2
	BART+FTB	62.0	61.3	61.7	62.3	61.8 ± 2.1
	SBTG+BART	61.9	61.6	61.7	62.0	61.8 ± 2.1
	BART+SNT	61.4	61.9	61.8	61.2	61.5 ± 2.1
	SBTG+FTB	60.5	67.2	63.1	56.1	61.5 ± 2.2
	NBM+FTB	62.7	54.9	59.9	67.3	60.9 ± 2.3
	NBM+BART	56.6	84.4	69.5	35.6	57.4 ± 2.3
	SBTG+NBM	56.8	33.7	52.8	74.4	52.0 ± 2.2
	SBTG+SNT	65.5	24.0	53.6	87.4	50.7 ± 2.3
	NBM+FTB+SNT	68.7	56.8	63.1	74.0	65.1 ± 2.1
	NBM+BART+SNT	63.2	67.1	64.9	60.9	63.9 ± 2.1
	NBM+BART+FTB	64.1	62.0	63.2	65.2	63.6 ± 2.1
	SBTG+NBM+FTB	61.6	70.0	65.2	56.3	62.9 ± 2.2

Dataset	Method	P(C)	R(C)	P(I)	R(I)	F1
ViGGO	SBTG+BART+FTB	61.8	67.8	64.3	58.1	62.8 ± 2.2
	BART+FTB+SNT	63.4	60.0	62.0	65.3	62.6 ± 2.2
	SBTG+BART+SNT	60.5	67.6	63.3	55.9	61.6 ± 2.1
	SBTG+NBM+SNT	60.6	59.0	60.1	61.7	60.3 ± 2.2
	SBTG+FTB+SNT	61.0	56.0	59.2	64.1	59.9 ± 2.2
	SBTG+NBM+BART	59.6	54.8	58.2	62.9	58.7 ± 2.2
	NBM+BART+FTB+SNT	67.3	61.8	64.7	70.0	65.8 ± 2.2
	SBTG+BART+FTB+SNT	62.3	71.3	66.5	56.8	63.8 ± 2.1
	SBTG+NBM+BART+SNT	61.2	71.3	65.7	54.9	62.8 ± 2.1
	SBTG+NBM+FTB+SNT	64.5	54.0	60.2	70.1	61.7 ± 2.1
	SBTG+NBM+BART+FTB	61.6	60.9	61.3	62.0	61.4 ± 2.2
	SBTG+NBM+BART+FTB+SNT	59.9	68.7	63.4	54.1	61.1 ± 2.1
	FTB	66.3	75.5	93.1	89.6	80.9 ± 2.1
	SNT	56.9	86.8	95.9	82.4	78.6 ± 2.0
	SBTG	58.1	67.8	90.9	86.8	75.6 ± 2.3
	NBM	50.6	79.1	93.4	79.2	73.7 ± 2.3
	BART	44.5	67.8	89.9	77.2	68.3 ± 2.4
	SBTG+SNT	71.0	81.0	94.6	91.1	84.2 ± 1.9
	FTB+SNT	71.6	73.4	92.8	92.2	82.5 ± 2.1
	BART+FTB	64.2	78.8	93.9	88.1	80.8 ± 2.1
	NBM+SNT	62.2	82.1	94.7	86.6	80.6 ± 2.1
	BART+SNT	58.4	83.0	94.9	84.1	78.8 ± 2.1
	SBTG+NBM	66.4	67.1	91.2	90.9	78.8 ± 2.3
	NBM+FTB	62.3	73.0	92.4	88.1	78.6 ± 2.2
	SBTG+BART	61.0	70.8	91.8	87.8	77.6 ± 2.2
	SBTG+FTB	57.5	67.2	90.7	86.6	75.2 ± 2.3
	NBM+BART	52.3	50.9	87.0	87.6	69.4 ± 2.4
	SBTG+FTB+SNT	67.5	74.6	93.0	90.3	81.2 ± 2.1
	NBM+FTB+SNT	68.1	72.7	92.6	90.9	81.0 ± 2.2
	SBTG+NBM+SNT	66.9	72.7	92.5	90.3	80.5 ± 2.1
	BART+FTB+SNT	63.0	80.0	94.2	87.3	80.5 ± 2.1
	NBM+BART+FTB	61.9	82.0	94.7	86.5	80.4 ± 2.1
	NBM+BART+SNT	64.7	75.7	93.1	88.9	80.3 ± 2.2
	SBTG+BART+SNT	61.3	72.7	92.3	87.7	78.2 ± 2.2
	SBTG+NBM+FTB	60.6	69.7	91.5	87.8	77.2 ± 2.3
	SBTG+BART+FTB	64.2	63.2	90.1	90.4	76.9 ± 2.3
	SBTG+NBM+BART	52.5	77.7	93.1	81.1	74.6 ± 2.2
	SBTG+NBM+FTB+SNT	76.2	66.9	91.5	94.4	82.1 ± 2.3
	NBM+BART+FTB+SNT	67.2	78.3	93.9	89.7	82.0 ± 2.1
	SBTG+NBM+BART+SNT	66.4	72.5	92.4	90.1	80.2 ± 2.2
	SBTG+BART+FTB+SNT	63.8	73.3	92.6	88.9	79.4 ± 2.2
	SBTG+NBM+BART+FTB	58.3	76.5	93.1	85.3	77.5 ± 2.2
	SBTG+NBM+BART+FTB+SNT	64.9	71.7	92.1	89.5	79.4 ± 2.2
Weather	SNT	91.4	94.1	66.2	56.6	76.8 ± 2.7
	FTB	90.4	93.9	62.8	50.8	74.1 ± 2.7
	NBM	87.9	95.8	62.7	34.8	68.1 ± 3.1
	SBTG	89.0	89.0	45.8	45.7	67.3 ± 2.7
	BART	86.4	98.0	71.5	24.1	63.8 ± 3.2
	SBTG+SNT	91.6	96.1	74.4	56.4	78.9 ± 2.6
	BART+SNT	91.9	94.7	69.2	59.1	78.4 ± 2.5

Dataset	Method	P(C)	R(C)	P(I)	R(I)	F1
Weather	BART+FTB	90.7	96.8	76.2	50.5	77.1 ± 2.8
	SBTG+NBM	93.4	89.8	57.5	68.6	77.0 ± 2.5
	NBM+SNT	90.5	96.6	74.7	49.6	76.4 ± 2.9
	SBTG+FTB	93.1	89.5	56.7	67.3	76.3 ± 2.5
	FTB+SNT	90.7	95.3	69.1	51.5	75.9 ± 2.8
	SBTG+BART	90.8	91.5	56.5	54.2	73.2 ± 2.6
	NBM+BART	89.3	95.4	65.4	43.5	72.1 ± 2.9
	NBM+FTB	88.5	97.1	72.1	37.3	70.8 ± 3.0
	BART+FTB+SNT	93.6	92.7	65.6	68.7	80.0 ± 2.4
	NBM+BART+SNT	91.6	95.3	71.1	56.7	78.2 ± 2.6
	NBM+FTB+SNT	91.3	95.4	70.7	55.1	77.5 ± 2.7
	SBTG+FTB+SNT	91.6	93.2	63.4	58.0	76.4 ± 2.6
	SBTG+NBM+SNT	92.0	92.2	60.8	60.2	76.2 ± 2.5
	NBM+BART+FTB	90.4	95.9	70.7	49.5	75.5 ± 2.7
	SBTG+BART+FTB	92.0	90.8	57.5	61.5	75.3 ± 2.6
	SBTG+NBM+BART	92.7	89.0	54.5	65.2	75.0 ± 2.4
	SBTG+NBM+FTB	92.2	89.0	53.7	62.7	74.1 ± 2.6
	SBTG+BART+SNT	90.8	91.2	55.6	54.2	72.9 ± 2.7
	SBTG+NBM+BART+FTB	92.2	92.6	62.9	61.3	77.2 ± 2.5
	NBM+BART+FTB+SNT	91.1	94.6	67.0	54.2	76.3 ± 2.8
	SBTG+NBM+BART+SNT	91.9	91.4	58.8	60.3	75.5 ± 2.6
	SBTG+BART+FTB+SNT	91.7	90.5	55.9	59.5	74.3 ± 2.5
	SBTG+NBM+FTB+SNT	90.1	95.2	66.9	48.2	74.2 ± 2.8
	SBTG+NBM+BART+FTB+SNT	92.7	89.0	54.4	65.0	75.0 ± 2.5
Alarm	FTB	94.6	99.6	83.0	24.5	67.4 ± 2.8
	SBTG	93.0	100.0	0.0	0.0	48.2 ± 0.2
	NBM	93.0	100.0	0.0	0.0	48.2 ± 0.2
	SNT	93.0	99.9	0.0	0.0	48.2 ± 0.2
	BART	93.1	100.0	0.0	0.0	48.2 ± 0.2
	FTB+SNT	94.5	98.8	58.9	23.6	65.1 ± 2.7
	SBTG+FTB	95.4	87.2	20.5	44.0	59.5 ± 1.8
	BART+FTB	93.9	95.0	20.0	16.7	56.3 ± 1.9
	NBM+FTB	93.2	91.6	8.0	9.9	50.6 ± 1.4
	NBM+BART	93.1	100.0	0.0	0.0	48.2 ± 0.2
	SBTG+BART	93.1	100.0	0.0	0.0	48.2 ± 0.2
	SBTG+NBM	93.1	100.0	0.0	0.0	48.2 ± 0.2
	NBM+SNT	93.0	99.1	0.0	0.0	48.0 ± 0.2
	SBTG+SNT	92.9	97.0	0.0	0.0	47.4 ± 0.2
	BART+SNT	92.8	96.6	0.0	0.0	47.3 ± 0.2
	NBM+BART+SNT	98.4	91.3	40.8	80.3	74.4 ± 1.9
	SBTG+FTB+SNT	93.0	95.2	5.7	3.9	49.4 ± 1.2
	BART+FTB+SNT	93.0	98.5	4.6	1.0	48.6 ± 0.8
	SBTG+NBM+SNT	93.0	98.0	3.5	1.0	48.5 ± 0.8
	NBM+FTB+SNT	93.1	99.9	0.0	0.0	48.2 ± 0.2
	SBTG+NBM+BART	93.0	100.0	0.0	0.0	48.2 ± 0.2
	SBTG+NBM+FTB	92.9	96.5	2.1	1.0	48.0 ± 0.7
	NBM+BART+FTB	92.7	87.1	4.8	8.7	48.0 ± 1.1
	SBTG+BART+FTB	93.0	98.8	0.0	0.0	47.9 ± 0.2
	SBTG+BART+SNT	92.7	94.9	0.0	0.0	46.9 ± 0.2
	SBTG+NBM+FTB+SNT	94.6	97.6	43.2	24.4	63.6 ± 2.4

Dataset	Method	P(C)	R(C)	P(I)	R(I)	F1
	NBM+BART+FTB+SNT	96.9	85.4	24.5	63.8	63.1 $\pm$ 1.7
	SBTG+NBM+BART+SNT	93.0	97.2	5.0	2.0	49.0 $\pm$ 1.1
	SBTG+NBM+BART+FTB	93.3	81.8	7.4	19.8	49.0 $\pm$ 1.2
	SBTG+BART+FTB+SNT	92.9	97.1	0.0	0.0	47.5 $\pm$ 0.2
	SBTG+NBM+BART+FTB+SNT	95.9	65.3	11.8	62.5	48.8 $\pm$ 1.2

Table 19: Ablation Experiments for Alarm, Weather, WebNLG, ViGGO and E2E datasets. Comparing precision (P) and recall (R) of the correct (C) and incorrect (I) classes. Lists down all performance numbers for individual and all possible combinations of synthetic data generation.