# Task-driven augmented data evaluation

**Olga Golovneva**[*1,2], **Pan Wei**[1], **Khadige Abboud**[1],
**Charith Peris**[1], **Lizhen Tan**[1], **Haiyang Yu**[*1,3]

[1]Alexa AI, Amazon, Cambridge, MA
[2]FAIR Labs, Meta, Washington, DC
[3]Google, New York, NY

olggol@meta.com, {panwei, abboudk, perisc, ltn}@amazon.com, yuhaiyang@google.com

## Abstract

The main focus of data augmentation research has been on the enhancement of generation models, leaving the examination and improvements of synthetic data evaluation methods less explored. In our work, we explore a number of sentence similarity measures in the context of data generation filtering, and evaluate their impact on the performance of the targeted Natural Language Understanding problem for the example of intent classification and named entity recognition tasks. Our experiments on ATIS dataset show that the right choice of filtering technique can bring up to 33% in sentence accuracy improvement for targeted underrepresented intents.

## 1 Introduction

Recent advances in transfer learning methods have been a driving force in the progress of many Natural Language Understanding (NLU) tasks. These methods typically involve pre-training of a large-scale language model, followed by the task-specific fine-tuning (Peters et al., 2018; Devlin et al., 2019). Although these approaches have helped achieve state-of-the-art results on a variety of supervised learning tasks, they do not directly address the problem of task-specific annotated data sparsity. This is where data augmentation techniques come in to play, boosting model performance for a given supervised task by generating novel data points that are similar in characteristics to the available data.

The main thrust of data augmentation research has been focused on improving generation models (Yu et al., 2017; Golovneva and Peris, 2020; Kim et al., 2020; Liu et al., 2020; Sun et al., 2020; Anaby-Tavor et al., 2020), while comparatively little work has been done on comprehensively evaluating and filtering high-quality synthetic utterances. Current approaches suggest the use of a combination of automated metrics that evaluate utterances at the word or embedding level (Sharma et al., 2017; Liu et al., 2020).

Popular word-based evaluation approaches are based on comparing n-grams in the original and generated text, and were originally developed for machine translation evaluations. Among them commonly used scores are Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002), Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004), and Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Lavie and Agarwal, 2007). Both BLEU and ROUGE-N are based on comparing the n-gram overlap of the reference and generated texts and counts the number of token matches, with the difference being that ROUGE is recall-focused whereas BLEU is precision-focused. METEOR uses a set 1-gram mappings between the reference and generated text to get a weighted F-score, and adds a penalty function for incorrect word order.

Word-based sentence evaluation can give low scores for predictions with high lexical variation, but these predictions are not necessarily poor quality. To address that, one could use embedding similarity (Sharma et al., 2017), that will measure the semantic similarity between the reference and prediction based on the cosine similarity between word and sentence embeddings.

While high-quality generated data should be similar to the source data at hand, it is also important for this data to be novel. To measure the diversity of the generated text, one can use self-BLEU (Zhu et al., 2018) score, which is computed by averaging the BLEU scores of each generated utterance using the rest of the generated text as the reference set. Furthermore, in an effort to address the diversity-quality trade-off of synthetically generated data, (Montahaei et al., 2019) propose joint diversity-quality metrics: MS-Jaccard similarity calculated as the average n-gram Jaccard index of the generated text, and Frechet BERT Distance

[*]Work done during the authors' tenure at Amazon.



*Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 18 - 25

(FBD) calculated as the Frechet distance between the generated text distribution and the reference set distribution while using BERT's feature space for the text.

Some approaches have been made to create a statistical model that would serve as an independent data evaluator. In their work, (Lowe et al., 2017) propose an RNN-based Automatic Dialogue Evaluation Model (ADEM) that predicts human evaluation score for the generated utterance. The drawback of this approach is that it requires additional data collection to train the model.

Despite a variety of approaches to annotated data evaluation, there is no golden standard, moreover, they often disagree on evaluation (Bhandari et al., 2020). However, the real value of the generated data can be only evaluated through the downstream task, for example by estimating how much performance improvement synthetic data can bring to the targeted supervised NLU task.

In our work, we are connecting automated data evaluation with the downstream tasks, and apply it as a filtering mechanism to generated utterances. Instead of attempting to build correlations with expensive human evaluations, we indirectly evaluate the quality of the generated data through the performance of the intent classification (IC) and named entity recognition (NER) tasks on the widely used Air Travel Information System (ATIS) dataset (Hemphill et al., 1990).

In summary, our contributions are as follows:

1. We propose a novel synthetic text data evaluation framework by adapting different word-based and embeddings-based similarity measures for post-generation quality evaluation linked to the performance improvement of the targeted NLU problem on the example of IC and NER tasks.

2. We propose a way to adapt a classification model, which can serve as an independent data evaluator and does not require additional data collection.

3. We adapt generation models to produce labeled data with and without delexicalization.

4. We conduct experiments on ATIS dataset, a standard benchmark dataset for intent classification and slot labelling. Our experimental results show that proposed methods help to improve generated data quality which reflects in model performance improvements.

## 2 Synthetic data evaluation

Data sparsity is a common issue in multiple areas of NLU research. It is often a difficult and costly exercise for researchers to secure the required amounts of high-quality annotated data to train their models. In our work, we aim to generate synthetic data that can be used to improve NLU model training. We use the original training data along with intent and slot labels as a source to the data generation model. Each utterance $u = w_0, w_1, ..., w_n$ of length $n$ is represented as a sequence of tokens, where $w_0$ is utterance's intent, and $w_i$ ($i = 1..n$) denotes the $i$th slot of the utterance. In this paper, we will use sentence and utterance interchangeably. Once data is generated, we use several text evaluation metrics to filter out high-quality utterances that will potentially bring greater benefit for the model.

### 2.1 Word-based sentence similarity

To evaluate the quality of the synthetic utterances through word similarity, we calculate the n-gram BLEU scores of the generated sentence pairing with each sentence in the original training data with the same intent. We also calculate the maximum of the n-gram BLEU scores with all other intents, then assign the difference between these two scores as our $maxBLEU$ score for the generated sentence. This score considers both the similarity of the generated utterance to within-intent source data and the dis-similarity to out-of-intent utterances. For each generated utterance $u$ for intent $j$, we calculate the in-intent BLEU score $BLEU_j = BLEU_N(u, U_j)$ where $N = \min(4, length(u))$, $U_j$ is the reference set of source utterances in intent $j \in I$, and calculate the out-of-intent BLEU score as the maximum BLEU scores across all other intents. Finally our quality score for the generated utterance is given by the difference:

$$maxBLEU = BLEU_j - maxBLEU_N(u, U_i)_{i \in I, i \neq j} \quad (1)$$

Instead of maximization, we can also use average operation to estimate the out-of-intent score to get the $avgBLEU$ score:

$$avgBLEU = BLEU_j - meanBLEU_N(u, U_i)_{i \in I, i \neq j} \quad (2)$$

maxBLEU score will show how the generated utterance is similar to the closest intent, other than

the generated one $j$, while avgBLEU score reflects how much close the utterance is to all other intents on average. Inherently, the maxBLEU and avgBLEU scores are similarity measures, while we aim to preserve both the similarity and dissimilarity within/out-of-intent that is in the original source data. The Jaccard distance on the other hand measures both the similarity and the diversity of the the data (Montahaei et al., 2019):

$$JD(S_{u_i}, S_{u_j}) = 1 - \frac{|S_{u_i} \cap S_{u_j}|}{|S_{u_i} \cup S_{u_j}|} \quad (3)$$

where $u_i$, $u_j$ denote two different sentences/corpora, $S_{u_i} = w \in u_i$ denotes the set of words in the sentence $u_i$. We apply intra-group similarity check by using Jaccard distance check, the steps are as follows:

1. for each intent $j$ in the reference, calculate the pairwise distance between each utterance and take the mean as the intent threshold, $t_j$;

2. for each generated utterance $u$, based on its first generated token (i.e. the intent $k$ it is predicted to be), calculate the Jaccard distance between uand all sentences in the reference intent group it falls into, if $mean(JD(u, u_m)_{u_m \in U_k}) < t_k$, then the generated sentence will be retained.

### 2.2 Embedding-based sentence similarity

In additional to token/ngram-based utterance similarity check, we also utilize word embedding which takes semantic context information into account for word similarity. For a given utterance, we construct a sentence embedding by averaging the embeddings of words composing in the sentence as in embedding similarity (Sharma et al., 2017). To compare the utterances, we use the popular cosine distance

$$CD(u_i, u_j) = 1 - \frac{\bar{e}_{u_i} \cdot \bar{e}_{u_j}}{\|\bar{e}_{u_i}\| \|\bar{e}_{u_j}\|} \quad (4)$$

where $\bar{e}_{u_i}, \bar{e}_{u_j}$ denote the average sentence embedding for sentences $u_i$, $u_j$ respectively.

With this definition, we apply the same intra-group similarity check algorithm as mentioned in the previous session to filter the generated sentences. In our experient, we use the pre-trained fastText English embeddings (Grave et al., 2018).

### 2.3 Independent evaluator

Finally, we use a machine learning model to evaluate synthetic data quality. Unlike ADEM model (Lowe et al., 2017), our evaluator, similar to the filtering method used in (Anaby-Tavor et al., 2020), does not aim to predict human evaluation scores. Instead, it acts as an independently trained discriminator that assigns the probabilities for the utterance to be real. For our evaluator, we first train a BERT-based intent classification model on the train partition of ATIS. To account for the data imbalance, we add class weights calculated based on class frequencies to the cross-entropy loss. This model is then used to evaluate each synthetic utterance $u$. An utterance is considered as legitimate and added to the augmented set only if the model confidence on predicting intent $w_0 = i, i \in I$, is greater than pre-defined threshold $t_i$. Each threshold is calculated as follows:

$$t_i = \begin{cases} \max(p_j), & \forall j \neq i, j \in J, |J| > 1 \\ \min(p_i), & \text{if } J = \{i\} \end{cases} \quad (5)$$

where $J \subset I$ is a set of all hypotheses for utterances that in the reference belong to the intent $i \in I$.

## 3 Experiments

In this section, we describe experimental setup, evaluation metrics and provide the summary of the experimental results.

### 3.1 Data

In our experiments, we use the Airline Travel Information System (ATIS) dataset imported from the Microsoft Cognitive Toolkit (CNTK). ATIS is a standard benchmark dataset widely used for intent classification and slot filling tasks. It consists of a set of spoken utterances in the context of airline information, classified into one of 26 intents with 127 slot labels. Table 1 shows train, dev and test sizes per intent. We note that the intent distribution of ATIS is highly imbalanced with over 70% of the data belonging to the one intent (*atis_flight*) while others intents have very low number of utterances sometimes within only one subset, train, dev or test.

### 3.2 Delexicalization

Similar to (Yu et al., 2020), in order to reduce noise and add more variety to the generated data, we experiment with using delexicalized utterances.

Table 1: Utterance count in training, development and test partition of ATIS dataset per intent.

| intent | train | dev | test |
|---|---|---|---|
| atis_flight | 3309 | 357 | 632 |
| atis_airfare | 385 | 38 | 48 |
| atis_ground_service | 230 | 25 | 36 |
| atis_airline | 139 | 18 | 38 |
| atis_abbreviation | 130 | 17 | 33 |
| atis_aircraft | 70 | 11 | 9 |
| atis_flight_time | 45 | 9 | 1 |
| atis_quantity | 41 | 10 | 3 |
| atis_flight#atis_airfare | 19 | 2 | 12 |
| atis_city | 18 | 1 | 6 |
| atis_airport | 17 | 3 | 18 |
| atis_distance | 17 | 3 | 10 |
| atis_capacity | 15 | 1 | 21 |
| atis_ground_fare | 15 | 3 | 7 |
| atis_flight_no | 12 | 0 | 8 |
| atis_meal | 6 | 0 | 6 |
| atis_restriction | 5 | 1 | 0 |
| atis_airline#atis_flight_no | 2 | 0 | 0 |
| atis_aircraft#atis_flight#atis_flight_no | 1 | 0 | 0 |
| atis_cheapest | 1 | 0 | 0 |
| atis_ground_service#atis_ground_fare | 1 | 0 | 0 |
| atis_airfare#atis_flight_time | 0 | 1 | 0 |
| atis_airfare#atis_flight | 0 | 0 | 1 |
| atis_day_name | 0 | 0 | 2 |
| atis_flight#atis_airline | 0 | 0 | 1 |
| atis_flight_no#atis_airline | 0 | 0 | 1 |

We preprocess the data by replacing slot values (some consisting of multiple tokens) with their corresponding slot labels, before feeding it into our data generation model. Once we obtain the generated data, we re-fill the slot labels present in these utterances with randomly sampled slot values which correspond to the label. The utterances thus created are used for the downstream tasks.

The following are the detailed steps together with examples:

- Use original training data to create catalogs, that for each label will contain a list of corresponding slot values extracted from catalogs, e.g.: {city_name: [london, denver, new york, boston]};

- For input data in generation model, anonymize slot values with slot labels, e.g. from “buy a ticket to denver” to “buy a ticket to city_name". This will help the generation model focus on carrier phrase and learn syntactic variations, rather than semantic similarities between slot values, as well as help to reduce the amount of noise in generated data, such as when model incorporates meaningless or confusing parts of the slot values (for example, "find the ticket price from new to san");

- Generate data using data generation model, e.g. generate sentence like “*atis_airfare* find the ticket price from city_name to city_name" with intent appended to the beginning of the utterance;

- Fill the slot value with catalogs by random sampling, e.g. from generated utterance “find the ticket price from city_name to city_name" we backpropagate to "find the ticket price from new york to boston".

### 3.3 Data generation

For data generation, we use the Sequence Generative Adversarial Networks (SeqGAN) model developed by (Yu et al., 2017). Generative Adversarial Nets (GANs) consist of two competing networks, generator and discriminator. Generator network implicitly learns data distribution through the feedback it receives from discriminator network, that is trained to distinguish fake and real data points. Unlike classic GANs, SeqGAN specifically addresses the issues of discrete tokens generation through a stochastic policy in reinforcement learning that will guide token-by-token sequence generation. The discriminator judges at sequence-level with the intermediate state-action value calculated using Monte Carlo (MC) search. While (Yu et al., 2017) applied the MC search at sequence-level, (Golovneva and Peris, 2020) expanded this work to apply the reinforcement learning reward as an average of the token-level rewards. Their results showed significant improvement in accuracy metrics for domain classification, intent classification, slot F1 and Frame accuracy in their task mimicking the bootstrapping of a new language. We use their methods as a basis for our data generation tasks here. It is worth noting, that synthetic data evaluation approach does not depend on the method chosen for data augmentation, but is driven by the downstream supervised NLU tasks.

Table 2: Performance results for stack-propagation model on ATIS dataset: baseline.

| method | slot F1 | intent acc | sentence acc |
|---|---|---|---|
| baseline, published | 95.900% | 96.900% | 86.500% |
| baseline, in-house | 96.031% | 96.678% | 89.212% |

Table 3: Performance results for stack-propagation model on ATIS dataset in terms of slot F1 score, intent accuracy (acc), and sentence acc. All metric values are calculated using the average of multiple trials, and t-test is used to determine whether the results are statistically significant or not. In the table, the results with * are statistically significant with $p < 0.1$, and best performing model for each metric is highlighted in bold. Utterance counts are provided for train and dev partitions combined. Data were generated with (wd) and without (nd) delexicalization.

| **method** | **# gen utt** | **# utt after filter** | **total # utt** | **slot F1** | **intent acc** | **sentence acc** |
|---|---|---|---|---|---|---|
| baseline | | | 4,978 | 96.031% | 96.678% | 89.212% |
| no filtering, nd | 7,519 | | 12,497 | 95.996% | 91.601% | 85.330% |
| maxBLEU, nd | 7,519 | 2,637 | 7,615 | 96.172% | 96.529% | **89.959%*** |
| weighted BERT, nd | 7,519 | 2,503 | 7,481 | 95.983% | **97.088%** | 89.586%* |
| jaccard, nd | 7,519 | 3,317 | 8,295 | **96.308%** | 96.267% | 89.436% |
| avgBLEU, nd | 7,519 | 3,099 | 8,077 | 95.948% | 96.417% | 89.205% |
| cosine, nd | 7,519 | 1,992 | 6,731 | 95.900% | 95.633% | 88.578% |
| BERT, nd | 7,519 | 4,739 | 9,717 | 94.882% | 96.081% | 83.763% |
| no filtering, wd | 9,591 | | 14,569 | 90.488% | 86.338% | 70.549% |
| maxBLEU, wd | 9,591 | 1,152 | 6,130 | 95.926% | 96.715% | 89.287% |
| weighted BERT, wd | 9,591 | 4,885 | 9,863 | 96.044% | 96.939% | 89.548%* |
| jaccard, wd | 9,591 | 3,348 | 8,326 | 95.998% | 96.753% | 89.474% |
| avgBLEU, wd | 9,591 | 3,499 | 8,477 | 95.056% | 95.529% | 88.026% |
| cosine, wd | 9,591 | 1,834 | 6,812 | 95.755% | 96.305% | 88.578% |
| BERT, wd | 9,591 | 5,332 | 10,310 | 94.134% | 96.001% | 83.521% |

### 3.4 Model Architecture

For IC and NER tasks, we select one of the recent state-of-the-art models, that achieved high performance in intent classification and slot labeling tasks on the ATIS dataset. A Stack-Propagation Framework with Token-Level Intent Detection proposed by (Qin et al., 2019) for joint intent detection and slot filling, that explicitly use intent information for slot labeling task. Unlike multitask framework, where two tasks share only encoder, stack-propagation explicitly provides features from one task (IC) to another (NER). Additionally to account for contextual information, BiLSTM encoder is enriched with self-attention.

### 3.5 Baseline

We evaluate model performance according to the three metrics: intent accuracy for IC task, micro-averaged slot F1 for NER prediction, and overall sentence accuracy which is the relative number of utterances for which the intent and all slots are correctly identified. First we train the model on non-augmented ATIS dataset, and average results over 3 runs. As shown in the Table 2, results published by (Qin et al., 2019) on application of a Stack-Propagation Framework to ATIS dataset are consistent with our results.

### 3.6 Results

In Table 3, we provide a summary of all experimental results which include data counts both pre- and post-filtering and final metric values. Using SeqGAN model described in Section 3.3) we generate two sets of 9600 utterances, one with delexicalization and one without. The generated data sets are approximately twice the size of the original training partition. Although we use training data for all intents as an input to the data generation model, we only augment utterances for underrepresented intents, which excludes the biggest *atis_flight* intent. We then remove any generated utterances which do not start with a valid intent. This led to 7519 and 9591 augmented sentences with and without delexicalization respectively. We apply our six filtering mechanisms as described in Section 2. based on the following approaches: maxBLEU, avgBLEU, Jaccard distance, cosine distance, BERT-based evaluator with and without class weight added to the loss function. We use the resulting filtered sets as augmentation for each experiment and present results of our quality evaluation tasks (IC and NER) in Table 3.

Our results show that three filtering methods consistently outperform the baseline, regardless of whether delexicalized or not, when considering

the overall sentence accuracy metric. These are maxBLEU and Jaccard scores, as well as BERT model with class weights. The overall highest improvement in sentence accuracy in observed when using maxBLEU with no delexicalization.

Contrary to expectations, we do not see much improvement offered by the use of delexicalization, with delexicalized augmentation outperforming non-delexicalized augmentation only in the case of Jaccard-based filtering. Filtering using the word-based sentence similarity methods - maxBLEU and Jaccard - show similar results, while the BERT-based classifier with weighted loss function outperforms on intent classification task. This is most likely due to the fact that the classifier was trained on the intent classification task at the slot value level, so it can be expected to be better at filtering those sentences where true sentence intent does not match the first token of the generated sequence, while it does not capture the correctness of the labels. maxBLEU shows better performance than aveBLEU, likely due to it being better at detecting and filtering those generated utterances that belong to a wrong intent. maxBLEU filtering without delexicalization produces the best overall sentence accuracy. In particular, in Table 4 we see up to 33% in overall sentence accuracy improvement for targeted underrepresented intents.

Table 4: Sentence accuracy comparison between in-house baseline and model with training data augmented by filtering generated data with maxBLEU score criteria. Per-intent performance shows significant improvements on intents with small amount of training data.

| intent | # test utt | baseline | maxBLEU |
|---|---|---|---|
| atis_flight | 632 | 94.357% | **94.568%** |
| atis_airfare | 48 | 97.222% | **97.917%** |
| atis_airline | 38 | 92.105% | **93.860%** |
| atis_ground_service | 36 | 66.667% | 66.667% |
| atis_abbreviation | 33 | 76.768% | **87.879%** |
| atis_capacity | 21 | **84.127%** | 82.540% |
| atis_airport | 18 | 59.259% | **62.963%** |
| atis_flight#atis_airfare | 12 | 41.667% | **47.222%** |
| atis_distance | 10 | 40.000% | **43.333%** |
| atis_aircraft | 9 | **81.481%** | 74.074% |
| atis_flight_no | 8 | 100.000% | 100.000% |
| atis_ground_fare | 7 | 66.667% | 66.667% |
| atis_city | 6 | **88.889%** | 61.111% |
| atis_meal | 6 | 55.556% | **77.778%** |
| atis_quantity | 3 | 77.778% | **88.889%** |
| atis_day_name | 2 | 0.000% | 0.000% |
| atis_flight_time | 1 | 33.333% | **66.667%** |
| atis_flight#atis_airline | 1 | **33.333%** | 0.000% |
| atis_flight_no#atis_airline | 1 | 33.333% | 33.333% |
| atis_airfare#atis_flight | 1 | 0.000% | 0.000% |

We run additional experiments to demonstrate that the maxBLEU filtering allows us to obtain higher quality utterances than simple random sampling. We randomly sample utterances from our GAN-generated dataset to match the utterance count yielded by maxBLEU filtering and add those to the original training set as shown in Table 5. We then train our NLU models on this dataset and obtain metric values for comparison. The results show that in all evaluations obtained using a train set augmented with maxBLEU filtered data significantly outperform those obtained using a train set augmented with random-sampled data (t-test $p < 0.1$). This shows that the maxBLEU filtering method helps in choosing utterances of significantly higher quality when compared to simple random sampling.

Table 5: Comparing maxBLEU filtering with random sampling.

| method | slot F1 | intent acc | sentence acc |
|---|---|---|---|
| rand sample, nd | 95.910% | 94.401% | 86.861% |
| maxBLEU, nd | **96.172%** | **96.529%** | **89.959%*** |

## 4 Conclusions

We have explored a number of different approaches for augmented data evaluation, that we used as a filter for generated data. We evaluated the effectiveness of the evaluations metrics based on the selected targeted supervised NLU tasks, intent classification and named entity recognition. Experiments show that the maxBLEU filtering method without delexicalization produces the best overall/sentence accuracy, while weighted BERT-based classifier and Jaccard distance provide the best performance in terms of intent accuracy and slot F1 scores, respectively. Filtering through word-based sentence similarity measures - maxBLEU and Jaccard - show consistent improvement across all metrics, while BERT-based classifier with weighted loss function filtering significantly outperforms on intent classification task. We relate it to the fact that being trained on the intent recognition task on full sentences, the classifier network can capture the correctness of the utterance intents, while it underperforms on evaluating the correctness of the labels.

## References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama

Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7383–7390.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, and Pengfei Liu. 2020. Metrics also disagree in the low scoring range: Revisiting summarization evaluation metrics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5702–5711, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Olga Golovneva and Charith Peris. 2020. Generative adversarial networks for annotated data augmentation in data sparse nlu. *Computing Research Repository*, arXiv:2012.05302.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Y. Kim, S. Won, S. Yoon, and K. Jung. 2020. Collaborative training of gans in continuous and discrete spaces for text generation. *IEEE Access*, 8:226515–226523.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Mingtong Liu, Erguang Yang, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2020. A learning-exploring method to generate diverse paraphrases with multi-objective deep reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2310–2321, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic Turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126, Vancouver, Canada. Association for Computational Linguistics.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, Minnesota. 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 *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. 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 2078–2087, Hong Kong, China. Association for Computational Linguistics.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *ArXiv*, abs/1706.09799.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Boya Yu, Konstantine Arkoudas, and Wael Hamza. 2020. Delexicalized paraphrase generation. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages

102–112, Online. International Committee on Computational Linguistics.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 2852–2858. AAAI Press.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 1097–1100, New York, NY, USA. Association for Computing Machinery.