ParaAMR: A Large-Scale Syntactically Diverse Paraphrase Dataset by AMR Back-Translation

Kuan-Hao Huang[†] **Varun Iyer**[⊕] **I-Hung Hsu**[◇]
Anoop Kumar[‡] **Kai-Wei Chang**[†‡] **Aram Galstyan**[‡◇]
[†]University of California, Los Angeles [⊕]University of Illinois Chicago
[◇]Information Science Institute, University of Southern California [‡]Amazon Alexa AI
{khhuang, kwchang}@cs.ucla.edu, viyer9@uic.edu
ihunghsu@isi.edu, {anooamzn, argalsty}@amazon.com

Abstract

Paraphrase generation is a long-standing task in natural language processing (NLP). Supervised paraphrase generation models, which rely on human-annotated paraphrase pairs, are cost-inefficient and hard to scale up. On the other hand, automatically annotated paraphrase pairs (e.g., by machine back-translation), usually suffer from the lack of syntactic diversity — the generated paraphrase sentences are very similar to the source sentences in terms of syntax. In this work, we present PARAAMR, a large-scale *syntactically diverse* paraphrase dataset created by abstract meaning representation back-translation. Our quantitative analysis, qualitative examples, and human evaluation demonstrate that the paraphrases of PARAAMR are syntactically more diverse compared to existing large-scale paraphrase datasets while preserving good semantic similarity. In addition, we show that PARAAMR can be used to improve on three NLP tasks: learning sentence embeddings, syntactically controlled paraphrase generation, and data augmentation for few-shot learning. Our results thus showcase the potential of PARAAMR for improving various NLP applications.

1 Introduction

Paraphrase generation is a long-standing task in natural language processing (NLP) (McKeown, 1983; Barzilay and Lee, 2003; Kauchak and Barzilay, 2006). It has been applied to various downstream applications, such as question answering (Yu et al., 2018), chatbot engines (Yan et al., 2016), creative generation (Tian et al., 2021), and improving model robustness (Huang and Chang, 2021). Most existing paraphrase generation models require a large amount of annotated paraphrase pairs (Li et al., 2019; Gupta et al., 2018; Kumar et al., 2020). Since human-labeled instances are expensive and hard to scale up (Dolan et al., 2004; Madnani et al., 2012; Iyer et al., 2017), recent research has explored the possibility of generating paraphrase pairs automatically. One popular approach is back-translation (Wieting and Gimpel, 2018; Hu et al., 2019a,b), which generates paraphrases of a source sentence by translating it to another language and translating back to the original language. Although back-translation creates large-scale automatically annotated paraphrase pairs, the generated paraphrases usually suffer from the lack of syntactic diversity — they are very similar to the source sentences, especially in syntactic features. Consequently, supervised paraphrase models trained with those datasets are also limited in their ability to generate syntactically diverse paraphrases. Furthermore, not all words can be perfectly translated into another language. As we will show in Section 4.3, this mismatch may produce subpar paraphrases.

In this work, we leverage abstract meaning representation (AMR) (Banarescu et al., 2013) to generate syntactically diverse paraphrase pairs. We present PARAAMR, a large-scale syntactically diverse paraphrase dataset based on AMR back-translation. As illustrated by Figure 1, our approach works by encoding a source sentence to an AMR graph, modifying the *focus* of the AMR graph that represents the main assertion, linearizing the modified AMR graph, and finally decoding the linearized graph back to a sentence. Since the new sentence shares the same AMR graph structure as the source sentence, it preserves similar semantics to the source sentence. At the same time, the change of *focus* makes the new main assertion different from that source sentence. When linearizing the AMR graph, a different concept will be emphasized at the beginning of the string. Therefore, the decoded sentence may have a much different syntax from the source sentence.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8047–8061

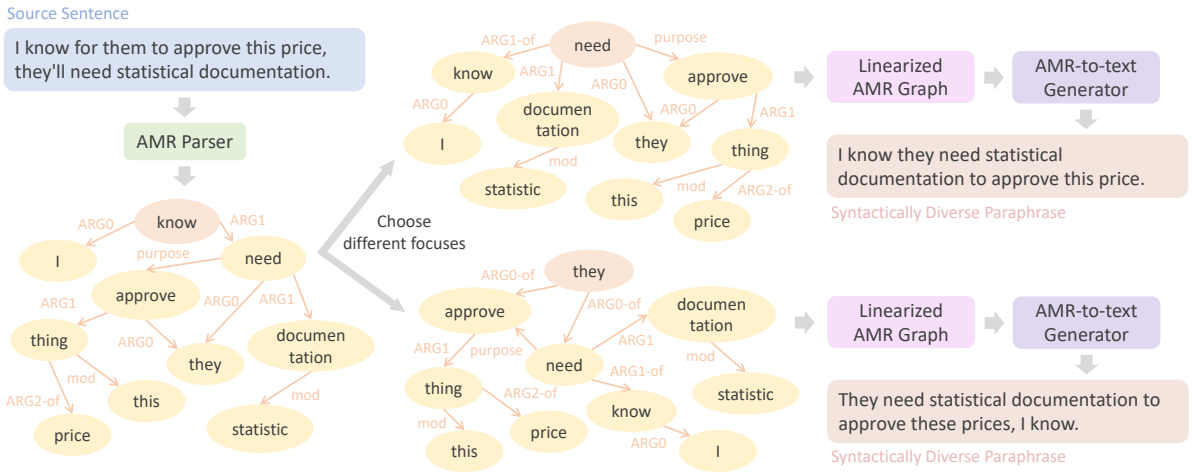

Figure 1: The overall framework to construct ParaAMR based on AMR back-translation. We encode a source sentence to an AMR graph, modify the *focus* of the AMR graph, linearize the modified AMR graph, and finally decode the linearized graph to a syntactically diverse paraphrase.

Our quantitative analysis (Section 4.2) and qualitative examples (Section 4.3) show that the paraphrases of ParaAMR are syntactically more diverse than existing datasets (Wieting and Gimpel, 2018; Hu et al., 2019a,b), while at the same time preserving good semantic similarity between paraphrased sentences. In addition, our human evaluation results (Section 4.4) confirm that ParaAMR is indeed more syntactically diverse than prior datasets. To showcase the benefits of syntactically diverse paraphrases, we conduct experiments on three downstream tasks: learning sentence embeddings (Section 5.1), syntactically controlled paraphrase generation (Section 5.2), and data augmentation for few-shot learning (Section 5.3). We observe that models trained on ParaAMR achieve better performance on all three downstream tasks compared to other datasets, thus indicating its potential value for various NLP applications.[1]

2 Related Work

Paraphrase generation and datasets. Traditional paraphrase generation models are usually based on hand-crafted rules, including rule-based methods (McKeown, 1983), thesaurus-based methods (Bolshakov and Gelbukh, 2004; Kauchak and Barzilay, 2006), and lattice matching methods (Barzilay and Lee, 2003). In recent years, different neural models have been proposed for paraphrase generation (Prakash et al., 2016; Mallinson et al., 2017; Cao et al., 2017; Egonmwan and Chali, 2019; Li et al., 2019; Gupta et al., 2018; Zhang et al., 2019c; Roy and Grangier, 2019; Iyyer et al., 2018; Huang and Chang, 2021). Some advanced techniques are proposed as well, such as multi-round generation (Lin and Wan, 2021), reinforcement-learning-based paraphrasing (Liu et al., 2020), and prompt-tuning (Chowdhury et al., 2022). To properly train those neural models, however, we needs a large corpus of annotated paraphrase pairs. Most existing paraphrase datasets and related resources, such as MRPC (Dolan et al., 2004), PAN (Madnani et al., 2012), PPDB (Ganitkevitch et al., 2013), and Quora (Iyer et al., 2017), have limited scale. Therefore, researchers have focused on automatically generating large-scale paraphrase corpora. One notable example is ParaNMT (Wieting and Gimpel, 2018), which is created by machine back-translation — translating texts to another language and translating them back to the original language.

Syntactically diverse paraphrase generation. Another line of research focuses on diversifying the generated paraphrases in terms of syntax. This includes sampling from latent spaces (Roy and Grangier, 2019; Zhang et al., 2019c; Cao and Wan, 2020), controlling word order (Goyal and Durrett, 2020), and controlling syntax (Iyyer et al., 2018; Cao and Clark, 2019; Kumar et al., 2020; Huang and Chang, 2021; Sun et al., 2021; Huang et al., 2022; Lee et al., 2022). Although they can diversify the generated paraphrases based on different model designs, those models are still limited due to the lack of diversity in existing large-scale paraphrase datasets. Some works propose large-scale

[1]Our proposed dataset is available at https://github.com/uclanlp/ParaAMR.

diverse paraphrases by considering different decoding methods during back-translation, including lexical constraints (Hu et al., 2019a) and cluster-based constrained sampling (Hu et al., 2019b). Although increasing the lexical diversity, the syntactic diversity of their datasets is still limited.

Text-to-AMR parsing. Abstract meaning representation (AMR) (Banarescu et al., 2013) is designed for capturing abstract semantics. Since it offers benefits to many NLP tasks, several works focus on parsing AMR from texts. Transition-based methods maintain a stack and a buffer for parsing AMR (Wang et al., 2015; Damonte et al., 2017; Ballesteros and Al-Onaizan, 2017; Vilares and Gómez-Rodríguez, 2018; Naseem et al., 2019). Graph-based approaches extract AMR based on graph information (Zhang et al., 2019a,b; Cai and Lam, 2020; Zhou et al., 2020). Sequence-to-sequence approaches directly linearize AMR and train end-to-end models to produce AMR (Konstas et al., 2017a; van Noord and Bos, 2017; Peng et al., 2017; Ge et al., 2019).

AMR-to-Text generation. Generating texts from AMR graphs is a popular research direction as well. Most existing approaches can be grouped into two categories. The first group is based on structure-to-text methods, where they build graphs to capture the structural information (Marcheggiani and Perez-Beltrachini, 2018; Song et al., 2018; Beck et al., 2018; Damonte and Cohen, 2019; Zhao et al., 2020; Wang et al., 2020). The second group is based on sequence-to-sequence methods (Konstas et al., 2017b; Ribeiro et al., 2021), where they treat AMR as a string and train end-to-end models.

3 PARAAMR

We propose PARAAMR, a large-scale syntactically diverse paraphrase dataset. Figure 1 illustrates the overall framework to construct PARAAMR by AMR back-translation. In summary, we encode a source sentence to an AMR graph, modify the *focus* of the AMR graph (see Section 3.3), linearize the modified AMR graph, and finally decode the linearized graph to a syntactically diverse paraphrase. We describe the details in the following.

3.1 Data Source

In order to fairly compare with prior works (Wieting and Gimpel, 2018; Hu et al., 2019a,b), we choose the same Czech–English dataset (Bojar et al., 2016) as our data source. Specifically, we directly use the English source sentences from the previous dataset (Hu et al., 2019b) as the source sentences for AMR back-translation. It is worth noting that our proposed method is not limited to this dataset but can be applied to any general texts for constructing syntactically diverse paraphrases.

3.2 Translating Texts to AMR Graphs

We use a pre-trained AMR parser to encode source sentences to AMR graphs. Specifically, we consider SPRING (Bevilacqua et al., 2021), a BART-based (Lewis et al., 2020) AMR parser trained on AMR 3.0 annotations[2] and implemented by amrlib.[3] As illustrated by Figure 1, each source sentence will be encoded to an AMR graph, which is a directed graph that has each node represents a semantic concept (e.g., *know*, *need*, and *they*) and each edge describe the semantic relations between two concepts (e.g., *ARG0*, *ARG1-of*, and *mod*) (Banarescu et al., 2013).

An AMR graph aims at capturing the meaning of a sentence while abstracting away syntactic, lexical, and other features. Each AMR graph has a *focus*, which is the root node of the graph, to represent the main assertion. For example, the focus of the AMR graph extracted from the source sentence in Figure 1 is *know*. Most of the time, the focus will be the main verb; however, it actually can be any concept node.

3.3 Translating AMR Graphs to Texts

Usually, syntactically different sentences with similar meanings have similar *undirected* AMR graph structures and differ only in their focuses and the directions of edges. We plan to use this property to construct syntactically diverse paraphrases of a source sentence.

Changing the focus of an AMR graph. After extracting the AMR graph from a source sentence, we construct several new graphs by changing the *focus*. More precisely, we randomly choose a node as the new focus and reverse all the incoming edges for that node. For instance, in Figure 1, when we choose *need* as the new focus, the incoming edge from *know* is reversed, and its edge label changes from *ARG1* to *ARG1-of*. Similarly, when we choose *they* as the new focus, the incoming edge from *need* and *approve* are reversed, and their

[2] https://catalog.ldc.upenn.edu/LDC2020T02
[3] https://github.com/bjascob/amrlib

edge labels change from *ARG0* to *ARG0-of*. Sometimes, to maintain a tree-like graph, some outgoing edges of the original focus node will be reversed as well (e.g., the edge between *know* and *need* is reversed when we choose *they* as the new focus). It is worth noting that when the focus changes, the *undirected* AMR graph structure remains the same, meaning that the new AMR graph preserves a similar abstract meaning to the old one. We implement the process of AMR re-focusing by the PENMAN package (Goodman, 2020).[4]

Linearizing AMR graph. After constructing several new graphs from the original AMR graph, we linearize the new graphs with the new focus (root node). This is done by traversing the AMR graph starting from the new focus node with a depth-first-search algorithm and converting it to the PENMAN notation. For example, the AMR graph with the focus being *need* can be linearized in the following format:

```
(z3 / need
    :ARG1-of (z1 / know
        :ARG0 (z2 / i))
    :ARG0 (z4 / they)
    :ARG1 (z5 / documentation
        :mod (z6 / statistic))
    :purpose (z7 / approve
        :ARG0 z4
        :ARG1 (z8 / thing
            :ARG2-of (z9 / price)
            :mod (z10 / this))))
```

Similarly, the AMR graph with the focus node *they* can be linearized in the following format:

```
(z4 / they
    :ARG0-of (z3 / need
        :ARG1 (z5 / documentation
            :mod (z6 / statistic))
        :purpose (z7 / approve
            :ARG0 z4
            :ARG1 (z8 / thing
                :ARG2-of (z9 / price)
                :mod (z10 / this)))
        :ARG1-of (z1 / know
            :ARG0 (z2 / i))))
```

Decoding AMR graph to texts. We use a T5-based pre-trained AMR-to-text generator (Ribeiro et al., 2021) to translate the linearized graphs back to sentences. Since the generated sentences share the same *undirected* AMR graph as the source sentence, they should have similar meanings and thus can be considered as paraphrases of the source sentence. In addition, we observe that the pre-trained AMR-to-text generator tends to emphasize the focus node of an AME graph at the beginning of the generated sentence. Therefore, the generated sentences from the linearized graphs with different focuses are very likely syntactically different from the source sentence.

3.4 Post-Processing

We notice that not all nodes are appropriate to be the focus. Choosing inappropriate nodes as the focus might generate paraphrases that are not grammatically fluent or natural. To avoid this situation, we use perplexity to filter out bad paraphrases. Specifically, we consider the GPT-2 model (Radford et al., 2019) implemented by HuggingFace's Transformers (Wolf et al., 2020) to compute the perplexity of a candidate paraphrase. We found that setting the filtering threshold to 120 is generally good enough, although some downstream applications may need different thresholds.

4 Comparison to Prior Datasets

We compare PARAAMR with the following three datasets. (1) PARANMT (Wieting and Gimpel, 2018) create paraphrase pairs by English-Czech-English back-translation. (2) PARABANK1 (Hu et al., 2019a) adds lexical constraints during the decoding of back-translation to increase the lexical diversity of generated paraphrases. (3) PARABANK2 (Hu et al., 2019b) proposes cluster-based constrained sampling to improve the syntactic diversity of generated paraphrases.

4.1 Basic Statistics

Table 1 lists the statistics of the PARANMT, PARABANK1, PARABANK2, and PARAAMR. PARAAMR contains syntactically diverse paraphrases to around 15 million source sentences. Notice that we consider the same source sentences as PARABANK2; however, some of the sentences fail to be parsed into ARM graphs. Therefore, the size of PARAAMR is slightly smaller than PARABANK2. The average length of paraphrases in PARAAMR is 15.20, which is similar to PARABANK2. Each source sentence in PARAAMR has

[4] https://github.com/goodmami/penman

Dataset	#Instances	Avg. #Para.	Avg. Len.
ParaNMT (Wieting and Gimpel, 2018)	51,409,584	1.00	11.90
ParaBank1 (Hu et al., 2019a)	57,065,358	4.31	12.16
ParaBank2 (Hu et al., 2019b)	19,723,003	4.75	15.51
ParaAMR (Ours)	15,543,606	6.91	15.20

Table 1: Basic statistics of ParaNMT, ParaBank1, ParaBank2, and ParaAMR.

Dataset	Semantic Similarity (↑)	Lexical Diversity 1 - BLEU (↑)	Lexical Diversity 1 - ∩/∪ (↑)	Syntactic Diversity TED-3 (↑)	Syntactic Diversity TED-F (↑)
ParaNMT (Wieting and Gimpel, 2018)	**84.28**	70.71	45.78	3.28	13.94
ParaBank1 (Hu et al., 2019a)	81.77	78.19	52.59	3.59	14.53
ParaBank2 (Hu et al., 2019b)	82.50	**88.82**	**59.61**	4.04	17.41
ParaAMR (Ours)	82.05	87.86	53.10	**5.86**	**22.07**

Table 2: Paraphrase diversity of different datasets. ParaAMR is syntactically more diverse than other datasets, while also showing comparable semantic similarity.

6.91 paraphrases on average, which is more than the other three datasets.

4.2 Quantitative Analysis

Following previous work (Hu et al., 2019b), we consider the same metrics to analyze semantic similarity, lexical diversity, and syntactic diversity of different paraphrase datasets. To fairly compare different datasets, we consider only those examples whose source sentences appear in all datasets. There are 193,869 such examples in total. All the following metrics are calculated based on those 193,869 examples.

We use the following metrics to evaluate the semantic similarity of paraphrases:

- **Semantic similarity measure by SimCSE**: Given two paraphrase sentences, we use the supervised SimCSE model (Gao et al., 2021) to get the sentence embeddings, and compute the cosine similarity between the two sentence embeddings as the semantic similarity.

Following the previous work (Hu et al., 2019b), we consider the following automatic metrics for lexical diversity:

- **1 - BLEU** (↑): We compute one minus BLEU score as the diversity score.
- **1 - ∩/∪** (↑): We first compute the ratio of the number of shared tokens between the two sentences and the union of all tokens in the two sentences, then use one minus the ratio as the diversity score.

We consider the following automatic metrics for syntactic diversity:

- **TED-3** (↑): We first get the constituency parse trees of the two sentences by using the Stanford CoreNLP parser (Manning et al., 2014). Then, we only consider the top-3 layers of trees and compute the tree editing distance as the score.
- **TED-F** (↑): We first get the constituency parse trees of the two sentences by using the Stanford CoreNLP parser (Manning et al., 2014). Then, we consider the whole tree and compute the tree editing distance as the score.

From Table 2, we conclude that the paraphrases generated by ParaAMR increase much more syntactic diversity while preserving comparable semantics compared to prior datasets.

4.3 Qualitative Examples

Table 3 shows some paraphrases generated by different datasets. We can observe that prior datasets based on machine back-translation tend to only replace synonyms as paraphrases. In contrast, ParaAMR is able to generate paraphrases that have much different word order and syntactic structures compared to the source sentence. This again showcases the syntactic diversity of ParaAMR.

In addition, we notice that other datasets may change the meaning of the source sentence (e.g., from *price* to *prize* and from *paddle* to *row*) due to the translation errors between different languages. ParaAMR, on the other hand, does not depend on other languages and thus is more reliable.

Source Sentence	I know for them to approve this price, they'll need statistical documentation.
PARANMT	I know that in order to accept this award, they'll need a statistical analysis.
PARABANK1	I know that to accept this prize, they're going to need statistical analysis. I know that in order to accept this prize, they're going to need a statistic analysis. I know that if they accept this prize, they're gonna need a statistical analysis.
PARABANK2	I know that to accept that prize, they're going to need a statistical analysis. I know that in order to accept this prize, they will require a statistical analysis. I know they'll require statistical analysis to accept that prize.
PARAAMR	I know they need statistical documentation to approve this price. There is statistic documentation I know they need to approve these prices. They need statistical documentation to approve these prices, I know.
Source Sentence	If I wanted to paddle down the river, where's the best place to launch out of?
PARANMT	If I wanted to row down a river, where's the best place to swim?
PARABANK1	If I wanted to row down the river, where's the best place to go? If I wanted to row down the riverside, where's the best place to go? If I wanted to row down the river, where's the best spot to float?
PARABANK2	If I want to paddle down the river, what'd be the most perfect spot to set sail?
PARAAMR	Where would be best for me to launch if I wanted to paddle down the river? It's a river I want to paddle down to, where's the best place to launch? Where's my best place to launch if I want to paddle down the river?

Table 3: Paraphrases generated by different datasets. The generated paraphrases by PARANMT, PARABANK1, and PARABANK2 usually have similar syntactic structures to the source sentences. In contrast, PARAAMR generates more syntactically diverse paraphrases.

Datasets	Semantic Similarity				Syntactic Diversity			
	3(%)	2(%)	1(%)	Average	3(%)	2(%)	1(%)	Average
PARANMT (Wieting and Gimpel, 2018)	28.7	46.7	24.6	**2.04**	16.7	45.0	38.3	1.78
PARABANK1 (Hu et al., 2019a)	26.8	49.0	24.2	2.03	15.1	47.8	37.1	1.78
PARABANK2 (Hu et al., 2019b)	26.8	50.3	22.9	**2.04**	14.2	51.8	34.0	1.80
PARAAMR (Ours)	26.5	47.2	26.3	2.00	18.2	53.8	28.0	**1.90**

Table 4: Human evaluation results. We evaluate semantic similarity and syntactic diversity in a score of three and report the distribution and the average score.

4.4 Human Evaluation

We additionally conduct human evaluations to measure the semantic similarity and the syntactic diversity of different datasets. We used the Amazon Mechanical Turk[5] to conduct the human evaluation. We randomly sample 300 paraphrases from each dataset, and design questions to measure the semantic similarity and syntactic diversity.

For semantic similarity, we design a 3-point scale question and ask the annotators to answer the question:

- **Score 3**: The two sentences are paraphrases of each other. Their meanings are near-equivalent.
- **Score 2**: The two sentences have similar meanings but some unimportant details differ.
- **Score 1**: Some important information differs or is missing, which alters the intent or meaning.

For syntactic diversity, we design a 3-point scale question and ask the annotators to answer the question:

- **Score 3**: The two sentences are written in very different ways or have much different sentence structures. (For example, "*We will go fishing if tomorrow is sunny.*" and "*If tomorrow is sunny, we will go fishing*")
- **Score 2**: Only some words in the two sentences differ. (For example, "*We will go fishing if tomorrow is sunny.*" and "*We are going to go fishing if tomorrow is sunny.*")
- **Score 1**: The two sentences are almost the same.

Appendix A lists more details of human evalua-

[5] https://www.mturk.com/

tion. The average scores of human evaluation are shown in Table 4. We observe that PARAAMR gets a much higher score for syntactic diversity although it has a slightly lower score for semantic similarity.

5 Applications

We focus on three downstream applications of PARAAMR corpus: learning sentence embeddings (Section 5.1), syntactically controlled paraphrase generation (Section 5.2), and data augmentation for few-shot learning (Section 5.3). We demonstrate the strength of PARAAMR and compare with prior datasets: PARANMT (Wieting and Gimpel, 2018), PARABANK1 (Hu et al., 2019a), and PARABANK2 (Hu et al., 2019b).

5.1 Learning Sentence Embeddings

We conduct experiments to show that PARAAMR is beneficial to learn sentence embeddings because of its syntactic diversity.

Settings. We consider the supervised SimCSE (Gao et al., 2021), a contrastive learning framework to learn sentence embeddings from (*reference sentence*, *positive sentence*, *negative sentence*) triplets. We train different SimCSE models with the paraphrase pairs in all four datasets. Specifically, for each (*source sentence*, *paraphrase sentence*) pair in the dataset, we consider the source sentence as the reference sentence, consider the paraphrase sentence as the positive sentence, and randomly sample one sentence from the dataset as the negative sentence.

Training details. We use the script provided by the SimCSE paper[6] (Gao et al., 2021) to train a SimCSE model with the weights initialized by `bert-base-uncased` (Devlin et al., 2019). The batch size is set to 128 and the number of epochs is 3. We set the learning rate to 10^{-5} and set other parameters as the default values from the script. It takes around 3 hours to train the SimCSE models for a single NVIDIA RTX A6000 GPU with 48GB memory. We set the perplexity threshold to 110 to filter PARAAMR. For each dataset, we train 5 different models with 5 different random seeds and report the average scores.

Evaluation. To evaluate the quality of sentence embeddings, we consider sentence textual similarity (STS) tasks from SentEval 2012 to 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016). We consider the script from SentEval[7] and use the learned sentence embeddings to calculate the cosine similarity between two sentences. We report the average Pearson correlation coefficient and the average Spearman correlation coefficient over all tasks.

Dataset	Pearson's r	Spearman's r
PARANMT	74.38 $\pm$ 0.70	73.80 $\pm$ 0.42
PARABANK1	74.80 $\pm$ 1.33	74.56 $\pm$ 1.02
PARABANK2	75.39 $\pm$ 0.29	75.17 $\pm$ 0.25
PARAAMR (ours)	**77.70** $\pm$ 0.40	**75.72** $\pm$ 0.43

Table 5: Results of learning sentence embeddings. We report 5-run average scores for STS 2012 to 2016. PARAAMR achieves the best performance.

Experimental results. Table 5 lists the average score for STS 2012 to 2016. We observe that the sentence embeddings learned with PARAAMR get better scores than other datasets, especially for the Pearson correlation coefficient. We hypothesize that the syntactic diversity of PARAAMR makes the sentence embeddings capture semantics better and reduce the influence of syntactic similarity.

5.2 Syntactically Controlled Paraphrase Generation

We demonstrate that PARAAMR is better for training a syntactically controlled paraphrase generator.

Settings. We consider the same setting as the previous works (Iyyer et al., 2018; Huang and Chang, 2021), which uses constituency parses as the control signal to train paraphrase generators. More precisely, the goal is to train a syntactically controlled paraphrase generator with the input being (*source sentence*, *target constituency parse*) pair and the output being a paraphrase sentence with syntax following the target constituency parse.

We consider the SCPN model (Iyyer et al., 2018), which is a simple sequence-to-sequence model, as our base model. We train different SCPN models with different datasets. for each (*source sentence*, *paraphrase sentence*) pair in the dataset, we treat the paraphrase sentence as the target sentence and use the Stanford CoreNLP toolkit (Manning et al., 2014) to extract constituency parse from the paraphrase sentence as the target parse.

Training details. Unlike the original SCPN paper (Iyyer et al., 2018), which uses LSTM as

[6] https://github.com/princeton-nlp/SimCSE

[7] https://github.com/facebookresearch/SentEval

Dataset	Quora	MRPC	PAN
ParaNMT	47.38 ± 0.39	45.24 ± 0.61	39.45 ± 0.50
ParaBank1	46.21 ± 0.26	44.52 ± 0.18	39.85 ± 0.11
ParaBank2	46.86 ± 0.45	45.17 ± 0.39	40.20 ± 0.56
ParaAMR (ours)	**48.50** ± 0.11	**47.38** ± 0.19	**40.30** ± 0.10

Table 6: Results of syntactically controlled paraphrase generation. We report 5-run average BLEU scores for Quora, MRPC, and PAN. ParaAMR performs the best.

the base model, we fine-tune the pre-trained `bart-base` (Lewis et al., 2020) to learn the syntactically controlled paraphrase generator. The batch size is set to 32 and the number of epochs is 40. The max lengths for source sentences, target sentences, and target syntax are set to 60, 60, and 200, respectively. We set the learning rate to 3×10^{-5} and consider the Adam optimizer without weight decay. For the beam search decoding, the number of beams is set to 4. It takes around 12 hours to train the SCPN model for a single NVIDIA RTX A6000 GPU with 48GB memory. We set the perplexity threshold to 85 to filter ParaAMR. For each dataset, we train 5 different models with 5 different random seeds and report the average scores.

Evaluation. We consider three human-annotated paraphrase datasets: Quora (Iyer et al., 2017), MRPC (Dolan et al., 2004), and PAN (Madnani et al., 2012), as the testing datasets. Specifically, we use the testing examples provided by previous work[8] (Huang and Chang, 2021) and calculate the BLEU score between the ground-truth and the generated output as the evaluation metric.

Experimental results. Table 6 shows the results of syntactically controlled paraphrase generation. The paraphrase generator trained with ParaAMR performs significantly better than others. We believe this is because ParaAMR provides several syntactically different paraphrases for one source sentence, therefore helping the paraphrase generator to better learn the association between parse and words.

5.3 Data Augmentation for Few-Shot Learning

Finally, we show that ParaAMR is helpful to generate augmented data for few-shot learning.

Settings. We choose the following three classification tasks from GLUE (Wang et al., 2019): MRPC, QQP, and RTE. We randomly sample 15 and 30 instances to train classifiers as the few-shot baseline. Since most tasks in GLUE do not provide the official test labels, we randomly sample 1/3 of instances from the dev set as the internal dev set and use the rest 2/3 instances as the testing set.

For each dataset, we use the learned syntactically controlled paraphrase generators from Section 5.2 to generate three augmented examples with different parses for each training instance. More specifically, we first use the pre-trained SCPN model (Iyyer et al., 2018) to generate the full parse trees from the following three parse templates: `(ROOT(S(NP)(VP)(.)))`, `(ROOT(S(VP)(.)))`, and `(ROOT(NP(NP)(.)))`. Then we use the generated full parse trees as the target parse for the syntactically controlled paraphrase generator. Finally, we train a classifier with the original 30 training instances and the augmented examples.

Dataset	MRPC	QQP	RTE
15-Shot Learning			
15-Shot Baseline	59.93	63.18	54.05
ParaNMT	49.26	63.54	**55.68**
ParaBank1	59.56	63.72	54.59
ParaBank2	58.46	63.54	54.05
ParaAMR (ours)	**62.87**	**64.08**	52.97
30-Shot Learning			
30-Shot Baseline	68.38	64.93	54.51
ParaNMT	67.65	66.20	52.71
ParaBank1	64.46	64.86	53.79
ParaBank2	68.38	64.91	54.15
ParaAMR (ours)	**69.36**	**67.03**	**55.60**

Table 7: ParaAMR has better performance of few-shot learning with data augmentation.

Training details. For the few-shot classifiers, we fine-tune `bert-base-uncased` (Devlin et al., 2019). We set the batch size to 8, set the learning rate to 10^{-4}, and set the number of epochs to 20. We consider Adam optimizer with weight decay being 10^{-5}. It takes around 5 minutes to train a few-shot classifier for a single NVIDIA RTX A6000 GPU with 48GB memory. We set the perplexity threshold to 110 to filter ParaAMR.

Experimental results. The results in Table 7 demonstrate that leveraging ParaAMR for data augmentation in few-shot learning scenarios leads to consistently better results compared to other

[8] https://github.com/uclanlp/synpg

paraphrasing corpora. This observation, combined with the two previous experiments, showcases the potential value of PARAAMR for various NLP applications.

6 Conclusion

In this work, we present PARAAMR, a large-scale syntactically diverse paraphrase dataset created by AMR back-translation. Our quantitative analysis, qualitative examples, and human evaluation demonstrate that the paraphrases of PARAAMR are more syntactically diverse than prior datasets while preserving semantic similarity. In addition, we conduct experiments on three downstream tasks, including learning sentence embeddings, syntactically controlled paraphrase generation, and data augmentation for few-shot learning, to demonstrate the advantage of syntactically diverse paraphrases.

Acknowledgments

We thank anonymous reviewers for their helpful feedback. We thank Amazon Alexa AI and the UCLA-NLP group for the valuable discussions and comments.

Limitations

Our goal is to demonstrate the potential of using AMR to generate syntactically diverse paraphrases. Although we have shown the strength of diverse paraphrases, there are still some limitations. First, our proposed techniques are strongly based on the quality of pre-trained text-to-AMR parsers and pre-trained AMR-to-text generators. If we cannot get a strong pre-trained text-to-AMR parser and a pre-trained AMR-to-text generator, the generated paraphrases might not have good quality. Second, one step in our proposed framework is modifying the root node of the AMR graph and therefore changing the focus of the AMR graph. However, not all nodes can be good root nodes to generate appropriate paraphrases. Some of them can be not fluent and much different from natural sentences. Although we use perplexity to filter out those paraphrases, there must be some imperfect paraphrases remaining. This partially affects the semantic scores of PARAAMR. Nevertheless, we still show that the current quality of PARAAMR is good enough to improve at least three NLP tasks.

Broader Impacts

Our dataset construction process relies on a pre-trained AMR-to-text generator. It is known that the models trained with a large text corpus may capture the bias reflecting the training data. Therefore, it is possible that PARAAMR contains offensive or biased content learned from the data. We suggest to carefully examining the potential bias before applying our dataset to any real-world applications.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING*.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013*.

Miguel Ballesteros and Yaser Al-Onaizan. 2017. AMR parsing using stack-lstms. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse (LAW-ID@ACL)*.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*.

Ondrej Bojar, Ondrej Dusek, Tom Kocmi, Jindrich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dusan Varis. 2016. Czeng 1.6: Enlarged czech-english parallel corpus with processing tools dockered. In *Text, Speech, and Dialogue - 19th International Conference (TSD)*.

Igor A. Bolshakov and Alexander F. Gelbukh. 2004. Synonymous paraphrasing using wordnet and internet. In *Proceedings of the 9th International Conference on Applications of Natural Languages to Information Systems*.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kris Cao and Stephen Clark. 2019. Factorising AMR generation through syntax. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Yue Cao and Xiaojun Wan. 2020. Divgan: Towards diverse paraphrase generation via diversified generative adversarial network. In *Findings of the Association for Computational Linguistics: (EMNLP-Findings)*.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.

Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. 2022. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*.

Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for amr-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

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 (NAACL-HLT)*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*.

Elozino Egonmwan and Yllias Chali. 2019. Transformer and seq2seq model for paraphrase generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: the paraphrase database. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

DongLai Ge, Junhui Li, Muhua Zhu, and Shoushan Li. 2019. Modeling source syntax and semantics for neural AMR parsing. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for AMR graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019a. PARABANK: monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*.

J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019b. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Kuan-Hao Huang, Varun Iyer, Anoop Kumar, Sriram Venkatapathy, Kai-Wei Chang, and Aram Galstyan. 2022. Unsupervised syntactically controlled paraphrase generation with abstract meaning representations. In *Findings of the Association for Computational Linguistics: (EMNLP-Findings)*.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. *data.quora.com*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics*.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017a. Neural AMR: sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017b. Neural AMR: sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha P. Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Trans. Assoc. Comput. Linguistics*, 8:330–345.

Fei-Tzin Lee, Miguel Ballesteros, Feng Nan, and Kathleen R. McKeown. 2022. Using structured content plans for fine-grained syntactic control in pretrained language model generation. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*.

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 (ACL)*.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*.

Zhe Lin and Xiaojun Wan. 2021. Pushing paraphrase away from original sentence: A multi-round paraphrase generation approach. In *Findings of the Association for Computational Linguistics (ACL/IJCNLP-Findings)*.

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 (COLING)*.

Nitin Madnani, Joel R. Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics*.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52th Conference of the Association for Computational Linguistics*.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*.

Kathleen R. McKeown. 1983. Paraphrasing questions using given and new information. *Am. J. Comput. Linguistics*, 9(1):1–10.

Tahira Naseem, Abhishek Shah, Hui Wan, Radu Florian, Salim Roukos, and Miguel Ballesteros. 2019. Rewarding smatch: Transition-based AMR parsing with reinforcement learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*.

Xiaochang Peng, Chuan Wang, Daniel Gildea, and Nianwen Xue. 2017. Addressing the data sparsity issue in neural AMR parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. In *Proceedings of the 26th International Conference on Computational Linguistics*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. AESOP: paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yufei Tian, Arvind Krishna Sridhar, and Nanyun Peng. 2021. Hypogen: Hyperbole generation with commonsense and counterfactual knowledge. In *Findings of the Association for Computational Linguistics: (EMNLP-Findings)*.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *arXiv preprint arXiv:1705.09980*.

David Vilares and Carlos Gómez-Rodríguez. 2018. A transition-based algorithm for unrestricted AMR parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations (ICLR)*.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020. Amr-to-text generation with graph transformer. *Trans. Assoc. Comput. Linguistics*, 8:19–33.

John Wieting and Kevin Gimpel. 2018. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Conference of the Association for Computational Linguistics (ACL)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP Demos)*.

Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. 2016. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *Proceedings of the 6th International Conference on Learning Representations*.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. 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)*.

Xinyuan Zhang, Yi Yang, Siyang Yuan, Dinghan Shen, and Lawrence Carin. 2019c. Syntax-infused variational autoencoder for text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Qiji Zhou, Yue Zhang, Donghong Ji, and Hao Tang. 2020. AMR parsing with latent structural information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

A Details of Human Evaluation

We use the template shown in Figure 2 to conduct the human evaluation. We sampled 300 paraphrases from PARANMT, PARABANK1, PARABANK2, and PARAAMR that share the same source sentences for human evaluation.

For each paraphrase pair, we ask three MTurkers to annotate the quality of semantics preservation and syntactic diversity in a 3-point scale question. We filter the MTurkers by approval rate greater than 97% and the number of approval greater than 50. The pay rate is $0.1 per paraphrase pair. We do not collect any personal information of MTurkers.

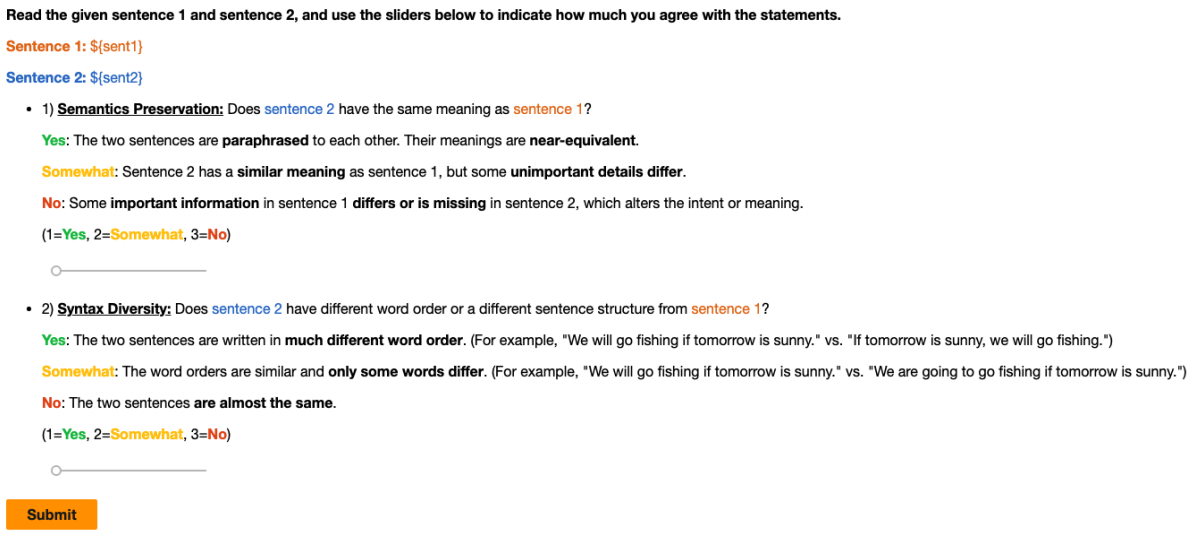

Figure 2: Screenshot of human evaluation instructions.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Please see the "Limitations" section.

☑ A2. Did you discuss any potential risks of your work?
Please see the "Broader Impacts" section.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3, 4, and 5

☑ B1. Did you cite the creators of artifacts you used?
Section 3, 4, and 5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3, 4, and 5

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3, 4, and 5

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Section 4

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Appendix A

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 4

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section 3

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix A