Revisiting Pivot-Based Paraphrase Generation: Language Is Not the Only Optional Pivot

Yitao Cai[*], **Yue Cao**[*] and **Xiaojun Wan**
Wangxuan Institute of Computer Technology, Peking University
Center for Data Science, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{caiyitao,yuecao,wanxiaojun}@pku.edu.cn

Abstract

Paraphrases refer to texts that convey the same meaning with different expression forms. *Pivot-based* methods, also known as the round-trip translation, have shown promising results in generating high-quality paraphrases. However, existing pivot-based methods all rely on language as the pivot, where large-scale, high-quality parallel bilingual texts are required. In this paper, we explore the feasibility of using *semantic and syntactic representations* as the pivot for paraphrase generation. Concretely, we transform a sentence into a variety of different semantic or syntactic representations (including AMR, UD, and latent semantic representation), and then decode the sentence back from the semantic representations. We further explore a pretraining-based approach to compress the pipeline process into an end-to-end framework. We conduct experiments comparing different approaches with different kinds of pivots. Experimental results show that taking AMR as pivot can obtain paraphrases with better quality than taking language as the pivot. The end-to-end framework can reduce semantic shift when language is used as the pivot. Besides, several unsupervised pivot-based methods can generate paraphrases with similar quality as the supervised encoder-decoder model, which indicates that parallel data of paraphrases may not be necessary for paraphrase generation.

1 Introduction

Paraphrase generation is an important and challenging task in the field of Natural Language Processing (NLP), which can be applied in a variety of applications such as information retrieval (Yan et al., 2016), question answering (Fader et al., 2014; Yin et al., 2015), machine translation (Cho et al., 2014), and so on.

Traditionally, paraphrase generation is usually implemented using ruled-based models (Fader et al., 2014; Zhao et al., 2009), lexicon-based methods (Bolshakov and Gelbukh, 2004; Kauchak and Barzilay, 2006), grammar-based methods (Narayan et al., 2016), statistical machine translation-based methods (Quirk et al., 2004; Zhao et al., 2008). With the rapid development of deep learning techniques, neural methods have shown great power in paraphrase generation and achieve state-of-the-art results (Gupta et al., 2018; Yang et al., 2019a). Neural paraphrase generation models usually follow the encoder-decoder paradigm. Given a sentence X, these models generate the paraphrase Y by directly modeling $P(Y|X)$ through a deep neural network. However, deep neural networks are sensitive to domains in general (Stahlberg, 2020), while existing mainstream paraphrase corpora only cover a few specific domains, such as image caption (Lin et al., 2014) and questions (Fader et al., 2013). High-quality paraphrases for general domains are difficult to obtain in practice, which greatly restricts the application of these seq2seq models.

Benefiting from the rapid development of machine translation technologies, pivot-based methods (Guo et al., 2019; Mallinson et al., 2017; Wieting et al., 2017) have been proposed for paraphrase generation. Formally speaking, pivot-based methods generate the paraphrase by following $P(Y|X) = P(Z|X)P(Y|Z)$, where Z denotes the pivot of X. Existing pivot-based methods all choose Z as representations in a different language, therefore the quality of the generated paraphrases largely depends on the pre-existing machine translation system.

Choosing language as pivot has some disadvantages, for example: (1) the pipeline translations may incur semantic shift (Guo et al., 2019), and (2) machine translation systems are sensitive to domain, and the quality of translating out-of-domain sentences can not be guaranteed.

[1]The first two authors contributed equally to this paper. Codes are available at https://github.com/caoy1996/Pivot-paraphrase.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4255–4268
November 7–11, 2021.

In this paper, we explore the feasibility of using different pivots for pivot-based paraphrasing models, including syntactic representation (Universal Dependencies (McDonald et al., 2013), UD), semantic representation (Abstract Meaning Representation (Banarescu et al., 2013), AMR), and latent semantic representation (LSR). Compared with choosing other languages as pivot, choosing syntactic or semantic as pivot is a more direct way, and is less likely to incur semantic shift. Apart from pipeline pivot-based generation, we also investigate how much an end-to-end pivot-based model, which can produce paraphrases in a single step with the help of pivot, affects the quality of paraphrases. In the end-to-end framework, the model directly learns the paraphrasing probability $P(Y|X)$ from text distribution $P(X)$ and $P(Y)$, pivot distribution $P(Z)$, as well as parallel text-pivot distribution $P(Z|X)$ and $P(Y|Z)$.

We conduct experiments on two benchmarks of paraphrasing tasks: Parabank and Quora datasets. We compared in detail the pros and cons of models using different pivots in terms of fidelity, fluency, diversity and so on in the experiments. The results show that using the AMR as the pivot can also produce high-quality paraphrases. Besides, the end-to-end framework can reduce the semantic shift when language is the pivot.

In sum, the prime contributions of this paper are as follows:

- We explore to use syntactic and semantic representations as pivots for pivot-based paraphrasing models, which is a more direct way and less likely to incur semantic shift.
- We also investigate applying an end-to-end paraphrasing model instead of the pipeline framework.
- We conduct experiments on two paraphrasing datasets to detailedly investigate the pros and cons of models using different pivots.
- We find out that models taking AMR as pivot can generate better paraphrases compared with taking UD or language as pivot. The end-to-end framework can also reduce the semantic changes when language is used as the pivot. Besides, several unsupervised pivot-based methods can generate paraphrases as good as the supervised encoder-decoder method, indicating that parallel samples may not be essential to generate high-quality paraphrases.

2 Introduction of Pivots

2.1 Language

Using language as the pivot has been widely explored by previous works (Wieting and Gimpel, 2018; Mallinson et al., 2017; Wieting et al., 2017; Guo et al., 2019). There are hundreds of languages in the world, and a sentence has different expressions in different languages. Therefore, we can take the sentence representation in another language as the pivot.

2.2 Abstract Meaning Representation (AMR)

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a rooted, labeled, acyclic graph which abstracts away from syntax and preserves semantics. Nodes in AMR graph are concepts, which are highly related to English words. Edges represent semantic relations between concepts. Since AMR only keeps semantic information, paraphrases can share the same AMR graph.

2.3 Universal Dependencies (UD)

Universal Dependencies (UD) (McDonald et al., 2013) is a framework for consistent annotation of parts of speech, morphological features and syntactic dependencies across human languages. Nodes in UD are tokens in sentences. Edge labels, Different from AMR, represent syntactic information.

2.4 Latent Semantic Representation (LSR)

The latent semantic representation (i.e. a dense vector) is also a simple way of meaning representation. We use a deep neural model to obtain the latent semantic representation of a given sentence.

3 Pipeline Pivot-based Paraphrase Generation

In the pipeline process, we first translate the input texts to pivots (Language, AMR or UD)[1], followed by generating paraphrases from pivots. This process is shown in Figure 1 (a).

3.1 Pipeline-language

We train an English-German and a German-English machine translation model with Transformers

[1]Note that LSR is not suitable to be used for pipeline generation.

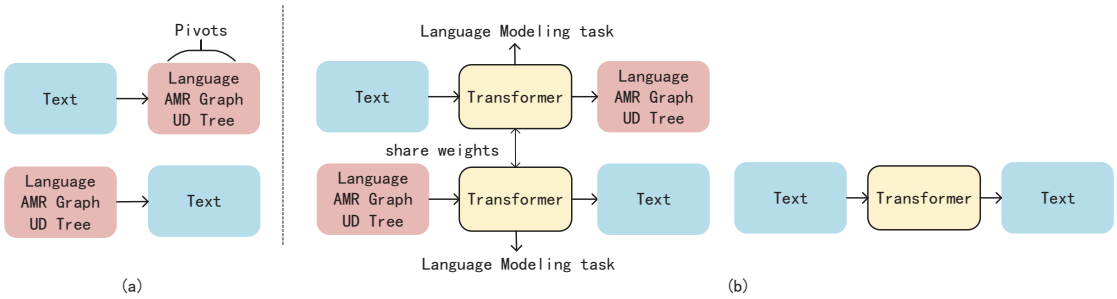

Figure 1: (a) Pipeline pivot-based paraphrase generation. (b) Left: training stage of end-to-end pivot-based paraphrase generation. Right: inference stage of end-to-end pivot-based paraphrase generation.

(Vaswani et al., 2017). The English sentences are first translated into German and then translated back into English. The sentences in German are regarded as the pivot.

3.2 Pipeline-AMR

When parsing texts to AMRs, we employ one of the state-of-the-art AMR parser (Xu et al., 2020). This is a sequence-to-sequence model, since AMR graphs are first linearized. Machine translation and constituent parsing are introduced as auxiliary tasks when training the model. Researchers first generate AMR graphs automatically with an existing AMR parser and construct a larger silver dataset. The seq2seq model is first trained on the silver dataset and fine-tuned on the gold dataset.

As for generating texts from AMRs, we choose the graph-to-text model (Ribeiro et al., 2020). This model is based on T5 (Raffel et al., 2020). It is first trained on a larger task-specific silver dataset and then fine-tuned on the gold English-AMR dataset.

3.3 Pipeline-UD

We apply Stanza toolkit (Qi et al., 2020) to obtain UD. Stanza is a pipeline system with tokenization, sentence and word segmentation, part-of-speech tagging, morphological features tagging, lemmatization and dependency parsing. We omit the model details here, which could be found in Qi et al. (2018).

We use the IMSurReal (Yu et al., 2019) to accomplish the UD-to-text task. The model first linearizes the UD trees and then inflects the lemmas into word forms. At last, the model contracts the tokenized word into one token.

4 Towards End-to-End Paraphrase Generation

The above pivot-based methods are simple and straightforward, but have two disadvantages: (1) It is difficult to control and optimize the pipeline system, and the quality of the generated paraphrases is totally determined by the text-to-pivot and pivot-to-text systems used. (2) The pipeline system is inefficient at the inference stage.

In this paper, we also investigate the feasibility of end2end methods. Different from the supervised paraphrasing models, our model does not involve any explicit paraphrase sentences, so it needs to generate paraphrases in a "zero-shot" way. Inspired by recent work on cross-lingual transfer (Conneau and Lample, 2019), we propose a pre-training framework to endow the model with the ability of zero-shot paraphrasing. Besides, we also experiment using auto-encoder to generate paraphrases. In the auto-encoder model, the encoded latent semantic representation (LSR) can be considered as a kind of semantic pivot.

4.1 LSR

We train a Transformer-based auto-encoder model, and use the encoder to encode the input sentence. The dense representation, which is the output of the encoder and can be considered as the latent semantic representation, is then decoded back to a sentence by the decoder.

4.2 End-to-end Pivot-based Method (E2E-pivot)

For E2E-pivot method, our framework contains only one encoder-decoder (transformer) model, which is learned from parallel text-to-pivot distribution $P(Z|X)$, pivot-to-text distribution $P(Y|Z)$, prior text distribution $P(X)$, $P(Y)$, and prior pivot distribution $P(Z)$. At the inference time, given an

input sentence, we guide the model to produce the output in text form again, which is then considered as the paraphrase of the input. The model architecture of the E2E-pivot method is in Figure 1 (b).

4.2.1 Language Modeling Tasks

Our language modeling task contains two sub-tasks: causal language modeling (CLM) and masked language modeling (MLM). We use CLM and MLM objectives to enable the model to learn a better encoder and decoder. These objectives have been proved effective for cross-lingual transfer in cross-lingual tasks.

Given a sentence, causal language modeling task trains to model the probability of a word given the prefix words: $P(x_t|x_1, x_2, \cdots, x_{t-1}; \theta)$, where x_t denotes the t-th word in sentence X, and θ denotes the model parameters. The training objective is to maximize the log likelihood:

$$\max \ \mathcal{L}_1(X) = \sum_{t=1}^{n} \log P(x_t|x_1, x_2, \cdots, x_{t-1}; \theta) \tag{1}$$

Our masked language modeling task is the same as Devlin et al. 2019a, which is also known as the Cloze task (Taylor, 1953). Concretely, we randomly sample 15% tokens from the input sentence, which are replaced by [MASK] tokens for 80% of the time, by random tokens for 10% of the time, and keep unchanged for 10% of the time. The training objective is to maximize the log reconstruction probability:

$$\max \ \mathcal{L}_2(X) = \log P(X|\tilde{X}; \theta) \tag{2}$$

where $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_t)$ is the corrupt sentence. We recommend readers to refer to Devlin et al. 2019a for more details.

The training objective of language modeling task is to maximize the sum of above two objectives:

$$\max \ \mathcal{L}_{LM}(X) = \mathcal{L}_1(X) + \mathcal{L}_2(X) \tag{3}$$

In our framework, we apply language model pretraining on both texts and pivots. AMR and UD are linearized with depth-first search.

4.2.2 Text-to-Pivot and Pivot-to-Text Tasks

The language modeling tasks only require non-parallel data. To leverage the parallel text-pivot data, we introduce text-to-pivot and pivot-to-text tasks.

Denoting X and Z as a parallel text-pivot sample, the training objective of text-to-pivot (t2p) is to maximize the log likelihood:

$$\max \mathcal{L}_{t2p}(X, Z) = \sum_{j=1}^{m} \log P(z_j|x_1, \cdots, x_n, z_1, \cdots, z_{j-1}; \theta) \tag{4}$$

Similarly, denoting Z and Y as a parallel pivot-text sample, the training objective of pivot-to-text (p2t) is:

$$\max \mathcal{L}_{p2t}(Z, Y) = \sum_{k=1}^{s} \log P(y_k|z_1, \cdots, z_j, y_1, \cdots, y_{k-1}; \theta) \tag{5}$$

The final objective is the sum of $\mathcal{L}_{LM}$, $\mathcal{L}_{t2p}$ and $\mathcal{L}_{p2t}$.

4.2.3 Tag and Indicator Embeddings

We add a special tag at the beginning of each sentence to specify the type of representation. For example, $\langle amr \rangle$ for AMR texts and $\langle en \rangle$ for English sentences.

At the inference stage, we set the first token of the decoder to $\langle en \rangle$ to force the model to produce sentences in text form again, which are then considered as the paraphrases of the input sentences.

However, we find that the tag does not always guarantee the type of the output sentences produced by the model. To keep the consistency, we follow (Conneau and Lample, 2019) to concatenate an *indicator embedding* into the word embedding. Concretely, supposing the word embedding for the i-th AMR token as e_i and the indicator embedding for AMR as a_{amr}, we concatenate the word embedding and the indicator embedding, and feed $[e_i, a_{amr}]$ as the input to the model.

5 Experiments

5.1 Datasets

In this paper, we conduct experiments on Parabank (Hu et al., 2019) and Quora[2] datasets, which are two benchmarks of the paraphrase generation task.

Parabank is a large-scale paraphrasing dataset from the general (news) domain. We use the officially released test set to evaluate the performance

[2] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

of models. The test set contains 36,417 test samples. The average length of sentences in the parabank dataset is 21.34.

Quora dataset contains over 155,000 paraphrased question pairs from the quora forum[3]. We adopt the quora test set to evaluate models' performance. The number of quora test samples is 4,000, and the average length of sentences in quora test set is 10.05.

We utilize WMT14 EN-DE dataset to train the machine translation system. As for AMR and UD, the gold parallel datasets are AMR 2.0 (LDC2017T10) and EWT (LDC2012T13). Since these corpus comes from similar domain as Parabank, Parabank can be regarded as the in-domain test set and Quora can be regarded as the out-of-domain test set. We can evaluate the domain robustness of pivot-based models.

5.2 Competitive Methods

We investigate and compare the performance of pipeline methods as well as end-to-end methods. Pipeline methods include Pipeline-language, Pipeline-AMR and Pipeline-UD, which are mentioned in Section 3. End-to-end methods consist of E2E-language, E2E-AMR and E2E-UD, which leverage language, AMR and UD as the pivot respectively and apply the end-to-end framework mentioned in Section 4. Besides, we also compare these unsupervised methods with a supervised encoder-decoder (Enc-dec) model based on Transformer, which is trained with parallel paraphrase pairs in the training set of ParaBank/Quora.

By analyzing performance of these models, we want to examine (1) whether AMR or UD can serve as the pivot for paraphrase generation, (2) whether end-to-end framework can bring benefit to paraphrase generation, and (3) whether zero-shot methods can obtain paraphrase as good as the supervised model.

5.3 Evaluation Metrics

We evaluate the paraphrasing models from the following aspects: (1) **Fidelity**, i.e., the semantic consistency between generated paraphrase and the original sentence. (2) **Diversity**, i.e., the degree of change in expression between the generated paraphrase and original sentence. (3) **Fluency**, i.e., the fluency of the generated paraphrase. (4) **The number of parallel samples** used for training the paraphrasing system.

To evaluate the fidelity automatically, we use BertScore (Zhang et al., 2020), which has been widely used to evaluate semantic similarity (Mager et al., 2020a; Cao and Wan, 2020; Dong et al., 2021).

To evaluate the diversity automatically, we calculate **"Self-BLEU"**, i.e., the BLEU-4 score between the output and input sentences. A high Self-BLEU score means that the output is similar to the input, and the diversity is poor, vice versa.

Besides the above automatic evaluation metrics, we also conduct the **human evaluation** to evaluate the quality of generated paraphrases of each model. Concretely, we randomly sample 100 test instances from Parabank and 100 test instances from Quora datasets, and ask volunteers to score the outputs from the following aspects: (1) Fidelity, (2) Diversity, and (3) Fluency. The scores range from 1-5, with 5 being the best. We guarantee that each instance is scored by at least 3 human annotators.

5.4 Implementation Details

We use the fairseq toolkit (Ott et al., 2019) to implement Pipeline-language and all end-to-end models. We set the model hidden size, feed-forward hidden size to 512 and 2048 respectively, and set the number of heads, number of layers to 8 and 6 respectively. We use the Adam optimizer (Kingma and Ba, 2014) for training, and adopt the warm-up learning rate (Goyal et al., 2017) technique for the first 4,000 steps.

6 Results and Analysis

The automatic evaluation results are shown in Table 1. The results of human evaluation on the Parabank and Quora test sets are shown in Table 2 and Table 3 respectively. We also calculate kappa coefficient to measure the consistency for each judge's evaluation.

6.1 Fidelity

The results in Table 1, Table 2 and Table 3 show that all models can achieve comparable or superior fidelity scores compared to the reference and the supervised model (Enc-dec), except for the Pipeline-language model. By checking the output files, we find that this is partially because the Pipeline-language model may introduce semantic shift during two-step translation. Compared with

[3] https://www.quora.com/

Method	Parabank		Quora	
	Self-BLEU ↓	BERTScore ↑	Self-BLEU ↓	BERTScore ↑
Source	100.00	100.00	100.00	100.00
Reference	42.42	74.41	32.05	67.34
Enc-dec	46.48	66.54	44.96	71.04
Pipeline-language	39.48	57.47	33.82	55.10
Pipeline-AMR	32.96	60.59	41.18	66.23
Pipeline-UD	82.30	81.87	83.11	93.23
LSR	89.00	82.48	95.08	98.06
E2E-language	43.00	65.80	42.32	66.38
E2E-AMR	42.30	64.47	41.88	67.81
E2E-UD	91.78	84.04	78.51	76.84

Table 1: Experimental results of paraphrase generation on parabank and quora datasets.

Method	Fid. ↑	Div. ↑	Flu. ↑
Reference	3.84	**2.74**	**4.20**
Enc-dec	3.70	2.61	3.99
Pipeline-language	3.18	2.64	3.48
Pipeline-AMR	3.73	2.69	4.03
Pipeline-UD	**4.31**	1.57	4.17
LSR	4.18	1.27	4.10
E2E-language	3.62	2.60	3.88
E2E-AMR	3.55	2.55	3.85
E2E-UD	3.97	1.46	3.89
Cohen's Kappa	0.407	0.422	0.480

Table 2: Results of the human evaluation on the Parabank test set.

Method	Fid. ↑	Div. ↑	Flu. ↑
Reference	3.61	**3.27**	4.51
Enc-dec	3.62	2.46	3.81
Pipeline-language	2.68	2.78	3.60
Pipeline-AMR	4.06	2.38	4.39
Pipeline-UD	4.74	1.41	4.45
LSR	**4.84**	1.16	**4.72**
E2E-language	3.41	2.35	3.93
E2E-AMR	3.25	2.42	3.73
E2E-UD	3.59	1.41	3.54
Cohen's Kappa	0.602	0.415	0.462

Table 3: Results of the human evaluation on the Quora test set.

Pipeline-language, Pipeline-AMR reduces semantic change, since AMR graphs preserve important words as concepts and thus preserve the original meaning. Pipeline-UD, LSR and E2E-UD seem to be able to achieve much higher fidelity scores than other methods, even than the reference. This is due to they produce sentences that are very similar to the source sentence, and sometimes even copy the whole source sentence entirely, which makes their output hardly change the semantics of the sentence, yielding high fidelity scores.

Compared to Pipeline-language, E2E-language achieves much higher scores in terms of fidelity, as it can preserve semantic information since end-to-end models do not require explicitly changing texts into pivots. However, E2E-AMR does not outperform Pipeline-AMR, which also demonstrates that the Pipeline-AMR method does not change semantics substantially.

6.2 Diversity

As for Pipeline-UD, LSR and E2E-UD model, the high score of Self-BLEU in Table 1, and the low score of Diversity in Table 2 and Table 3 reveal that paraphrases predicted by these three models are usually copied from the input texts.

In Parabank, Pipeline-AMR can achieve a similar score in terms of diversity as Pipeline-language. Besides, both Pipeline-AMR and Pipeline-language can achieve better results in diversity than E2E-language, E2E-AMR and Enc-dec in Parabank, revealing that pipeline process can generate more diverse sentences. However, in Quora, the diversity of Pipeline-AMR is similar to E2E-AMR and E2E-language and is far less than Pipeline-language. This is because syntactic information is removed in AMR and thus Pipeline-AMR always produces syntactically diverse sentences. Compared to Pipeline-AMR, Pipeline-language is more likely to replace words or phrases with their synonyms. Texts in Quora are shorter and simpler than ones in Parabank, which is harder for model to produce syntactically diverse output in Quora. Thus the diversity score of Pipeline-AMR is similar to E2E-language and E2E-AMR and is less than Pipeline-language in Quora.

6.3 Fluency

The fluency scores in Table 2 and Table 3 show that all models can generate fluent texts, especially Pipeline-AMR, Pipeline-UD and LSR. With language modeling tasks, E2E-pivot models can also

Source Text: which candidate handled the race question best during the first presidential debate ?
Reference: who provided a better response to the question regarding race relations in the us during the first presidential debate ?
Enc-dec: which candidate deals with the question of the race best in the first presidential debate ?
Pipeline-language: Which candidate has treated the symptoms best in the first half of the year ?
Pipeline-AMR: Which candidate best handled the race question in the first presidential debate ?
Pipeline-UD: which candidate handled the race question best during the first presidential debate ?
LSR: which candidate handled the race question best during the first presidential debate ?
E2E-language: what candidate did the race question best in the first presidential debate ?
E2E-AMR: candidate did the race question best in the first presidential debate ?
E2E-UD: which candidate handled the race question best during the first presidential debate ?

Table 4: An example from the Quora dataset.

generate fluent texts. Pipeline-language performs worst in fluency among these models in Parabank dataset, since the translation systems may sometimes generate irrelevant words and phrases, which both affect fidelity as well as fluency. When it comes to Quora dataset, Pipeline-language and E2E-UD get the lowest scores, which shows that these two methods may generate incoherent sentences and they are sensitive to domains.

6.4 Number of parallel samples required

We also analyze the cost of training each model. In terms of the number of samples used for training, the training of machine translation models in Pipeline-language method requires much more gold parallel training samples than other semantic or syntax based models.

For training AMR-based models, the text-to-AMR model leverages 2M silver training samples and the AMR-to-text model uses 3.9M silver training data. These models both leverage the gold data with 36k samples. For training UD-based models, we only use about 12K training samples. Since training auto-encoder models does not require any parallel samples, we can easily construct auto-encoder training samples from any non-parallel texts. While for training language-based models, we use more than 4.5M training samples to train a well-performed NMT model. It has been proven that the performance of NMT models is greatly limited by the number of training samples (Koehn and Knowles, 2017), when the number of samples is small, the performance of NMT models will be greatly reduced.

Source Text: There may be a hundred crimes in the background , but it is only on this one that they can be tried .
Reference: There may be a hundred felonies in the background , but they can only be tried on this one .
Enc-dec: there may be hundreds of crimes in the background , but they can only be tried on this one .
Pipeline-language: There may be a hundred crimes in the background , but only on this basis can they be condemned .
Pipeline-AMR: 100s of crimes in the background, but they can only be tried in this one.
Pipeline-UD: there may be a hundred crimes in the background , but it is only on this one that they can be tried .
LSR: there may be a hundred crimes in the background , but it is only on this one that they can be tried .
E2E-language: we may have a hundred crimes in the background , but we can only try them on this one .
E2E-AMR: there may be hundreds of crimes in the background , but they can only be tried on this one .
E2E-UD: there may be a hundred crimes in the background , but it is only on this one that they can be tried .

Table 5: An example from the Parabank dataset.

6.5 Summary of Observations

In sum, we have the following conclusions:

- The Pipeline-language method generates paraphrases of low fidelity scores and low fluency scores. Pipeline-language method is more likely to change the semantics of sentences and more sensitive to domains. The E2E-language method can alleviate the semantic changes to some extent, generating paraphrases with good quality.

- The UD-based and LSR methods tend to generate paraphrases with fewer changes in expression compared to the original sentence. However, they require much less human-annotated parallel samples for training compared to other methods.

- AMR-based methods perform well in fidelity, diversity, and fluency, which indicates that language is not the only optional pivot and using

AMR as the pivot is also a good choice for pivot-based paraphrase generation systems.

- Compared to the Enc-dec method, Pipeline-AMR, E2E-language and E2E-AMR methods can generate paraphrases with similar fidelity, diversity and fluency scores, which indicates that parallel paraphrasing data may not be necessary for generating high-quality paraphrases.

7 Case Analysis

Table 4 shows an example of Quora, consisting of paraphrases predicted by all competitive methods mentioned in section 5.2. In this case, Pipeline-UD, LSR and E2E-UD generate the same sentence with the original sentence. Pipeline-language is the only model that fails to preserve semantics, due to the error propagation of machine translation systems.

Table 5 is another example from Parabank. It reveals that Pipeline-AMR tends to paraphrase texts syntactically, while Pipeline-language tends to paraphrase texts by replacing words with their synonyms (e.g. *tried* and *condemned*).

8 Related Work

8.1 Paraphrase Generation

Recently, seq2seq-based methods have been widely used in the task of paraphrase generation and achieve state-of-the-art results. These models include Transformer-based (Prakash et al., 2016; Li et al., 2019; Kajiwara, 2019), Variational Autoencoder-based (Bowman et al., 2016; Shakeri and Sethy, 2019), Generative Adversarial Networks-based (Yang et al., 2019b; An and Liu, 2019; Cao and Wan, 2020), and Reinforcement Learning-based (Li et al., 2018) methods.

Some translation-based models have also been proposed for paraphrase generation (Wieting and Gimpel, 2018; Mallinson et al., 2017; Wieting et al., 2017; Guo et al., 2019). Wieting et al. 2017 and Wieting and Gimpel 2018 select different languages as pivots to generate multiple and diverse paraphrase. Considering that two-step translation may incur semantic shift, Guo et al. 2019 build a Transformer-based language model and pre-train the model on the concatenated bilingual parallel sentences.

8.2 Text-to-AMR and AMR-to-Text

As for AMR parsing, some previous works (Flanigan et al., 2014; Lyu and Titov, 2018; Zhang et al., 2019a) first project words to AMR concepts and then identify the relations. Transition-based models are widely applied (Wang et al., 2015b,a; Damonte et al., 2017; Liu et al., 2018; Guo and Lu, 2018; Naseem et al., 2019; Lee et al., 2020). Because of the rapid development of sequence-to-sequence model, many works leverage it to parse texts into AMRs. Some works (Konstas et al., 2017; van Noord and Bos, 2017; Ge et al., 2019; Xu et al., 2020) linearize AMR graphs and directly use sequence-to-sequence models. Others (Zhang et al., 2019b; Cai and Lam, 2020a) use sequence-to-sequence model to predict concepts. They also jointly train the model to identify the relations.

In AMR-to-text generation, recent methods (Song et al., 2018; Beck et al., 2018; Damonte and Cohen, 2019; Ribeiro et al., 2019) employs GNNs to explicitly encode graph structures. Other approaches (Zhu et al., 2019; Cai and Lam, 2020b; Wang et al., 2020; Yao et al., 2020) encode AMR graph structures through self-attention and Transformers. Ribeiro et al. (2020) and Mager et al. (2020b) utilize pre-trained models and achieve better results.

8.3 Text-to-UD and UD-to-Text

In UD parsing, graph-based models are widely used (Dozat et al., 2017; Straka, 2018; Qi et al., 2018). Besides, many works (Smith et al., 2018; Kulmizev et al., 2019; Grünewald et al., 2020) attempt to make use of contextual embeddings such as ELMO (Peters et al., 2018), Bert (Devlin et al., 2019b), XLM-R (Conneau et al., 2020) and so on.

UD-to-text task is introduced in Surface Realisation Shared Task (Mille et al., 2018, 2019, 2020). Several works (Ferreira et al., 2018; Castro Ferreira and Krahmer, 2019; Elder, 2020; Farahnak et al., 2020) first linearize UD trees without word reordering and then feed the linearized trees to the sequence-to-sequence models or statistical machine translation models to generate texts. Others (Cabezudo and Pardo, 2018; Yu et al., 2019; Recski et al., 2020; Yu et al., 2020) reorder the word in UD trees with neural models first, followed by word inflection.

9 Conclusions and Future Work

In this work, we focus on pivot-based paraphrase generation. Previous works leverage language as the pivot, which may introduce semantic shift. In this work, we explore whether we can use AMR

or UD as pivot. We also explore an end-to-end framework in a zero-shot way, using only parallel text-pivot data. Results of the automatic metrics and human evaluations show that AMR is a good choice of pivot, as AMR graphs preserve important words as concepts and thus preserve semantics. Moreover, replacing two-step pipeline process with the end-to-end framework is beneficial when language is the pivot, reducing the semantic change. Besides, some unsupervised pivot-based methods can perform as well as supervised paraphrase models. In the future, we will focus on zero-shot paraphrase generation task and explore more semantic representations as pivots for pivot-based paraphrase generation.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), Beijing Academy of Artificial Intelligence (BAAI) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Zhecheng An and Sicong Liu. 2019. Towards diverse paraphrase generation using multi-class wasserstein GAN. *CoRR*, abs/1909.13827.

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*, pages 178–186.

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 (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Igor A. Bolshakov and Alexander F. Gelbukh. 2004. Synonymous paraphrasing using wordnet and internet. In *Natural Language Processing and Information Systems, 9th International Conference on Applications of Natural Languages to Information Systems, NLDB 2004, Salford, UK, June 23-25, 2004, Proceedings*, volume 3136 of *Lecture Notes in Computer Science*, pages 312–323. Springer.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. ACL.

Marco Antonio Sobrevilla Cabezudo and Thiago Pardo. 2018. Nilc-swornemo at the surface realization shared task: Exploring syntax-based word ordering using neural models. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 58–64.

Deng Cai and Wai Lam. 2020a. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020b. Graph transformer for graph-to-sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7464–7471.

Yue Cao and Xiaojun Wan. 2020. Divgan: Towards diverse paraphrase generation via diversified generative adversarial network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2411–2421. Association for Computational Linguistics.

Thiago Castro Ferreira and Emiel Krahmer. 2019. Surface realization shared task 2019 (MSR19): The team 6 approach. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 59–62, Hong Kong, China. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing*

Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 7057–7067.

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, Volume 1 (Long and Short Papers)*, pages 3649–3658, Minneapolis, Minnesota. Association for Computational Linguistics.

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: Volume 1, Long Papers*, pages 536–546.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.

Qingxiu Dong, Xiaojun Wan, and Yue Cao. 2021. Parasci: A large scientific paraphrase dataset for longer paraphrase generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 424–434.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Henry Elder. 2020. Adapt at sr'20: How preprocessing and data augmentation help to improve surface realization. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 30–34.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 1156–1165. ACM.

Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1608–1618. The Association for Computer Linguistics.

Farhood Farahnak, Laya Rafiee, Leila Kosseim, and Thomas Fevens. 2020. Surface realization using pretrained language models. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 57–63, Barcelona, Spain (Online). Association for Computational Linguistics.

Thiago Castro Ferreira, Sander Wubben, and Emiel Krahmer. 2018. Surface realization shared task 2018 (sr18): The tilburg university approach. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 35–38.

Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

Donglai Ge, Junhui Li, Muhua Zhu, and Shoushan Li. 2019. Modeling source syntax and semantics for neural amr parsing. In *IJCAI*, pages 4975–4981.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Stefan Grünewald, Annemarie Friedrich, and Jonas Kuhn. 2020. Graph-based universal dependency parsing in the age of the transformer: What works, and what doesn't. *arXiv preprint arXiv:2010.12699*.

Yinpeng Guo, Yi Liao, Xin Jiang, Qing Zhang, Yibo Zhang, and Qun Liu. 2019. Zero-shot paraphrase generation with multilingual language models. *CoRR*, abs/1911.03597.

Zhijiang Guo and Wei Lu. 2018. Better transition-based amr parsing with a refined search space. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1712–1722.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019. PARABANK: monolingual bitext generation and sentential paraphrasing

via lexically-constrained neural machine translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6521–6528. AAAI Press.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6047–6052. Association for Computational Linguistics.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. 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 2755–2768, Hong Kong, China. Association for Computational Linguistics.

Young-Suk Lee, Ramón Fernandez Astudillo, Tahira Naseem, Revanth Gangi Reddy, Radu Florian, and Salim Roukos. 2020. Pushing the limits of amr parsing with self-learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3208–3214.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3865–3878. Association for Computational Linguistics.

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, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3403–3414. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Yijia Liu, Wanxiang Che, Bo Zheng, Bing Qin, and Ting Liu. 2018. An amr aligner tuned by transition-based parser. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2430.

Chunchuan Lyu and Ivan Titov. 2018. Amr parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 397–407.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020a. Gpt-too: A language-model-first approach for amr-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020b. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852, Online. 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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 881–893. Association for Computational Linguistics.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97.

Simon Mille, Anja Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (sr'20): Overview and evaluation results. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 1–20.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (sr'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (sr'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17.

Shashi Narayan, Siva Reddy, and Shay B. Cohen. 2016. Paraphrase generation from latent-variable pcfgs for semantic parsing. In *INLG 2016 - Proceedings of the Ninth International Natural Language Generation Conference, September 5-8, 2016, Edinburgh, UK*, pages 153–162. The Association for Computer Linguistics.

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 Annual Meeting of the Association for Computational Linguistics*, pages 4586–4592.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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.

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 *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2923–2934. ACL.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Chris Quirk, Chris Brockett, and William B. Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 142–149. ACL.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Gábor Recski, Ádám Kovács, Kinga Gémes, Judit Ács, and Andras Kornai. 2020. BME-TUW at SR'20: Lexical grammar induction for surface realization. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 21–29, Barcelona, Spain (Online). Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. 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 3183–3194, Hong Kong, China. Association for Computational Linguistics.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*.

Siamak Shakeri and Abhinav Sethy. 2019. Label dependent deep variational paraphrase generation. *CoRR*, abs/1911.11952.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018. An investigation of the interactions between pre-trained word embeddings, character models and POS tags in dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2711–2720, Brussels, Belgium. 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 (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Felix Stahlberg. 2020. Neural machine translation: A review. *J. Artif. Intell. Res.*, 69:343–418.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7:93–108.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015a. Boosting transition-based amr parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 857–862.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015b. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020. Amr-to-text generation with graph transformer. *Transactions of the Association for Computational 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 Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 451–462. Association for Computational Linguistics.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 274–285. Association for Computational Linguistics.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. Improving amr parsing with sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2501–2511.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Qian Yang, Zhouyuan Huo, Dinghan Shen, Yong Cheng, Wenlin Wang, Guoyin Wang, and Lawrence Carin. 2019a. An end-to-end generative architecture for paraphrase generation. 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 3132–3142, Hong Kong, China. Association for Computational Linguistics.

Qian Yang, Zhouyuan Huo, Dinghan Shen, Yong Cheng, Wenlin Wang, Guoyin Wang, and Lawrence Carin. 2019b. An end-to-end generative architecture for paraphrase generation. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3130–3140. Association for Computational Linguistics.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7145–7154, Online. Association for Computational Linguistics.

Pengcheng Yin, Nan Duan, Ben Kao, Junwei Bao, and Ming Zhou. 2015. Answering questions with complex semantic constraints on open knowledge bases. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1301–1310. ACM.

Xiang Yu, Agnieszka Falenska, Marina Haid, Ngoc Thang Vu, and Jonas Kuhn. 2019. IMSurReal: IMS at the surface realization shared task 2019. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 50–58, Hong Kong, China. Association for Computational Linguistics.

Xiang Yu, Simon Tannert, Ngoc Thang Vu, and Jonas Kuhn. 2020. IMSurReal too: IMS in the surface realization shared task 2020. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*,

pages 35–41, Barcelona, Spain (Online). Association for Computational Linguistics.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019a. Amr parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. Broad-coverage semantic parsing as transduction. *arXiv preprint arXiv:1909.02607*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 834–842. The Association for Computer Linguistics.

Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. 2008. Combining multiple resources to improve smt-based paraphrasing model. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 1021–1029. The Association for Computer Linguistics.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. 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 5459–5468, Hong Kong, China. Association for Computational Linguistics.