Unsupervised Paraphrasing under Syntax Knowledge

Tianyuan Liu, Yuqing Sun[*], Jiaqi Wu, Xi Xu, Yuchen Han, Cheng Li, Bin Gong

School of Software, Shandong University
zodiacg@foxmail.com, sun_yuqing@sdu.edu.cn, oofelvis@163.com,
{sidxu,hanyc}@mail.sdu.edu.cn, 609172827@qq.com, gb@sdu.edu.cn

Abstract

The soundness of syntax is an important issue for the paraphrase generation task. Most methods control the syntax of paraphrases by embedding the syntax and semantics in the generation process, which cannot guarantee the syntactical correctness of the results. Different from them, in this paper we investigate the structural patterns of word usages termed as the *word composable knowledge* and integrate it into the paraphrase generation to control the syntax in an explicit way. This syntax knowledge is pretrained on a large corpus with the dependency relationships and formed as the probabilistic functions on the word-level syntactical soundness. For the sentence-level correctness, we design a hierarchical syntax structure loss to quantitatively verify the syntactical soundness of the paraphrase against the given dependency template. Thus, the generation process can select the appropriate words with consideration on both semantics and syntax. The proposed method is evaluated on a few paraphrase datasets. The experimental results show that the quality of paraphrases by our proposed method outperforms the compared methods, especially in terms of syntax correctness.

Introduction

The paraphrase task aims to generate different expressions having the similar meanings with a given text (McKeown 1983). It requires both understanding the semantics of original text and generating a proper sentence, which is regarded as an advanced natural language processing task. There are some supervised (Li et al. 2019) and unsupervised paraphrasing methods (Wieting, Mallinson, and Gimpel 2017). But they do not consider the syntax correctness, which makes the generated sentences not applicable for many human centered downstream tasks.

In recent years, the syntax controlled paraphrase problem catches much attention from the academia (Kumar et al. 2020; Iyyer et al. 2018). Since there are more and more applications demanding the syntax correctness of the paraphrases, such as text summarization (Zhao et al. 2018), machine translation (Zhou, Sperber, and Waibel 2019), and text simplification (Coster and Kauchak 2011) and recommendation (Xie et al. 2022). For example, a simplified text like *Simple English Wikipedia* serves as a helpful resource for kids or second-language educations. Since this kind of resources requires a lot of human efforts, the syntax controlled paraphrasing could be an alternative way to provide the good quality sentences to such resources.

For the syntax controlled paraphrase generation, there are two aspects: the conformation on the structure-level syntax in the template and the word-level syntax soundness of the generated sentence. The former refers to the high level structure of the sentence, such as the order of core components and clauses. The representative way to achieve this goal is to disentangle the semantic and syntax guidance in paraphrase generation. Some works discard the structural information when encoding the original sentence by removing the word position information in sentence (Huang and Chang 2021). Since the syntax structure guidance is encoded in an implicit way, the paraphrase generator do not check the syntax enforcement explicitly, leading to possible grammatical mistakes in the results. To explicitly evaluate the syntax conformation, some works use a pretrained parser to evaluate the generated sentences and feed the result to the network as a syntactic indicator for training the model (Kumar et al. 2020). These methods emphasize the consistency of the generated text at the top level of syntax template without the explicit word level control in the generation process. Thus, it is difficult to ensure the word level syntactic soundness of the generated paraphrase. But the word-level syntax soundness plays an important role for the generated sentence. For example, in the downstream application of adversarial example generation, a paraphrase with wrong grammar would be easily identified and rejected by the attacked model.

Another kind of representative approaches focuses on the word level syntax soundness of the generated paraphrase. Casas et al. adopt the dependency parsing tree as the form of syntax template. It explicitly traverses the parsing tree and iteratively generates the paraphrase (Casas, Fonollosa, and Costa-jussà 2020). Since the dependency parsing trees provides direct word level syntax relationships, the method generates paraphrase with good syntax quality. However, the generated paraphrases have the semantics problem as the sentence may be drifted from the original sentence meaning. Thus, it is not applicable for many downstream applications.

Inspired by the theory of syntagmatic and paradigmatic relations in linguistics (Bowman et al. 2016), we propose

[*]Corresponding author.

the *word composable knowledge* to learn the composition patterns of word usages. Based on this knowledge, we design an unsupervised syntax controlled paraphrase generation method **CKPara**, which integrates the syntax constraints in the generation process in an explicit manner. Specially, we design a hybrid syntax matching loss, which includes two parts. The local part evaluates the word-level syntax soundness of the paraphrases, which is computed against the word composable knowledge, while the hierarchical structure part ensures the sentence-level consistency with the given syntax template. Compared to the existing unsupervised syntax controlled paraphrasing methods, the hybrid loss function provides not only the consistency to the syntax template, but also the diverse choices of words within and beyond the original sentence.

We verified the **CKPara** on two paraphrasing datasets: Quora and SimpleWiki. The experimental results show that the designed method can generate paraphrases that are more similar to the reference paraphrases than other compared methods. Moreover, the generated paraphrases by our method match the given syntax template better than compared methods. Experiments are also conducted to inspect the contributions of components of our method.

Related Works

Paraphrase Generation Paraphrase generation is a long-standing task in the area of natural language processing (McKeown 1983). While traditional methods rely on hand-crafted rules (Kauchak and Barzilay 2006; Barzilay and Lee 2003), the recent development of neural network based methods has pushed the performance significantly. Most methods treat paraphrasing generation as a sequence-to-sequence translation task and train the model based on parallel data (Li et al. 2019; Egonmwan and Chali 2019; Liu et al. 2020). Recently unsupervised paraphrase draws more attention because of its advantage in training without parallel data and potential to provide diversity. Wieting et al. proposed to use back translation as paraphrasing (Wieting, Mallinson, and Gimpel 2017) and generated ParaNMT-50m dataset (Wieting and Gimpel 2018). There are also methods using statistical methods to sample a paraphrase such as Metropolis-Hastings sampling by Miao et al.(Miao et al. 2019) and stimulated annealing by Liu et al.(Liu et al. 2020). Roy et al. proposed to use variational auto encoders to improve the diversity of generated paraphrases (Roy and Grangier 2019). While these methods generate paraphrases with similar meaning to the original sentence, they lack the ability to explicitly control their expression.

Syntax Controlled Text Generation Syntax guidance information is a promising way to improve the quality of generated text in terms of grammar. Iyyer et al. proposed syntactical controlled paraphrase generation in exploring adversarial example generation (Iyyer et al. 2018). Zhang et al. proposed a syntax infused variantional auto encoder which disentangles the semantic and syntax aspect of text generation (Zhang et al. 2019b). The disentanglement of semantic and syntax representation is widely adopted in following works such as Yuan et al. (2021). Huang et al. uses a transformer without the positional embeddings to reduce the sequential information contained in the semantic encoding (Huang and Chang 2021). In Yang et al. (2021), a pretrained syntax evaluator is used to provide syntax matching training signal for the model training process. Casas et al. designed an iterative generation process to generate explicitly according to given syntax template (Casas, Fonollosa, and Costa-jussà 2020), while the semantics of generated paraphrases drifts significantly due to multi-round generation. Compared to existing work, we used word composable knowledge to evaluate whether the generated paraphrase follows the given syntax guidance in a word-level manner. Furthermore, most of the works use the constituency parsing tree as the form of syntax guidance. In our method, the dependency parsing tree is adopted. It provides explicit word to word relationship guidance compared to constituency parsing tree.

The Syntax Controlled Paraphrasing under Word Composable Knowledge

The theory of syntagmatic and paradigmatic relations is very important in linguistics (Bowman et al. 2016). Comparing to the flexible expression of a sentence, the patterns on word combinations are more stable. To enforce the syntax controlled paraphrase, we design two levels metrics of soundness, namely the word-level combinations and the sentence-level structure, to guide the generation following the given syntax. We choose the dependency parsing syntax since it offers more concrete relationships between words than the constituency parsing. In this section we would present the details on the word composable knowledge and the syntax controlled paraphrasing.

Word Composable Knowledge

Following the theory of syntagmatic and paradigmatic relations (Bowman et al. 2016), we model the word composition patterns as the word composable knowledge. We choose some web available corpus with the annotated dependency parsing information for learning this knowledge. To represent each word, we the pretrained word embeddings (PWE) are used in favor of contextual word embeddings (CWE). Since CWE contain information from the context of each word while the PWE contain the word centered usage information in a static manner and are more suitable for our purpose. By re-using the existing word semantic space, words not in the annotated treebank can also be modeled (Pan et al. 2021). Let (w_h, r, w_t) denote a dependency relationship in the corpus, where w_h, w_t are the words and r is the dependency relation type between them. $\mathbf{w}_h, \mathbf{w}_t$ are the pretrained word embeddings. The soundness of combination (w_h, r, w_t) is modeled as the empirical estimation of its appearance in a proper text, denoted by $p(w_h, r, w_t)$. The functions $f_r^h(\cdot), f_r^t(\cdot)$ project the embeddings of words into syntax spaces according to the relation type and the head or tail position of the word. Then the function $g(\cdot,\cdot)$ computes its likelihood.

$$\mathcal{S}(w_h, r, w_t) = g(f_r^h(\mathbf{w}_h), f_r^t(\mathbf{w}_t)) \tag{1}$$

$$p(w_h, r, w_t) \propto \mathcal{S}(w_h, r, w_t) \tag{2}$$

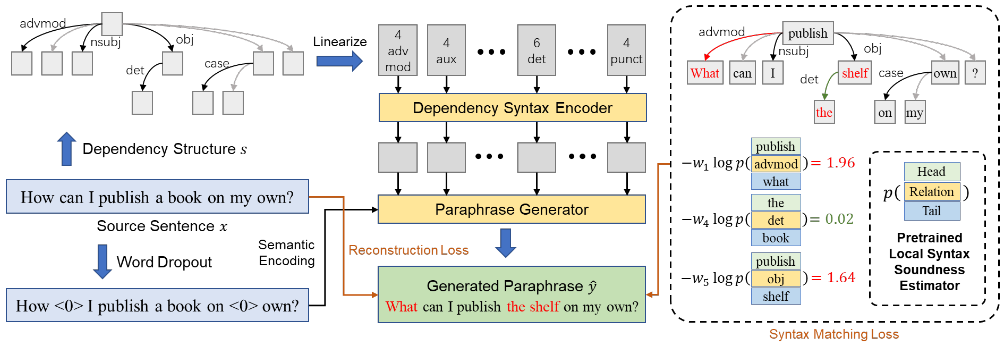

advmod
nsubj
obj
det
case
Linearize
4 adv mod
4 aux
6 det
4 punct
Dependency Syntax Encoder
Dependency Structure s
How can I publish a book on my own?
Source Sentence x
Word Dropout
Semantic Encoding
How <0> I publish a book on <0> own?
Paraphrase Generator
Reconstruction Loss
Generated Paraphrase $\hat{y}$
What can I publish the shelf on my own?
publish
What
can
I
shelf
the
own
on
my
?
$-w_1 \log p($publish advmod what$) = 1.96$
$-w_4 \log p($the det book$) = 0.02$
$-w_5 \log p($publish obj shelf$) = 1.64$
$p($Head Relation Tail$)$
Pretrained Local Syntax Soundness Estimator
Syntax Matching Loss

Figure 1: The model architecture

We design two loss functions to match the above objective. The first is a contrastive loss inspired by noise contrastive estimation technique (Gutmann and Hyvärinen 2010). The energy functions are trained by distinguishing proper syntax relationship triplets with negative ones.

$$\begin{aligned}\ell_1 = &- \sum_{w_h, r, w_t \in C} \log \sigma(\mathcal{S}(w_h, r, w_t)) \\ &- \sum_{w'_h, r', w'_t \in C'} \log \sigma(-\mathcal{S}(w'_h, r', w'_t))\end{aligned} \tag{3}$$

where C is the training set, C' is the set of negative samples. The negative samples are generated in three possible ways: 1) reverse the head and tail word; 2) change the syntax relation and 3) change the head or tail word. Such that the generated negative samples are unlikely to be proper syntax relationships.

The second loss function is to differentiate the syntax relation spaces projected by $f_r^h(\cdot), f_r^t(\cdot)$ to avoid the spaces converging to a less meaningful solution, i.e., the spaces are very close to each other. Such that each transform function $f_r^{\cdot}$ focuses on a word's syntax characteristics that is related to r. For a triplet $< w_h, r, w_t >$, the loss ℓ_2 is designed to minimize the probability of combining the head vector under r and the tail vector under a different relation, i.e. $r' \neq r$.

$$\ell_2 = - \sum_{u,v \in V, r, r' \in R} \log \sigma(-g(f_r^h(\mathbf{w}_h), f_{r'}^t(\mathbf{w}_t)) \tag{4}$$

Let Θ denote the parameters of model, the final loss function for learning the composable knowledge is defined as the following equation, where $||\Theta||_2^2$ is a regularization component, α and λ are the hyper-parameters.

$$\Theta^* = \arg\min_{\Theta} \{\ell_1 + \alpha \ell_2 + \lambda ||\Theta||_2^2\} \tag{5}$$

The functions $f_r^h(\cdot), f_r^t(\cdot), g(\cdot, \cdot)$ are the formed word composable knowledge.

Syntax Controlled Paraphrasing

The syntax controlled paraphrase task takes a source sentence $\mathbf{x} = (w_1, w_2, ..., w_n)$ and a dependency syntax structure $\mathbf{s} = (s_1, s_2, ...s_m)$ as inputs, and generates a sentence $\mathbf{y} = (y_1, y_2, ..., y_m)$ with the same meaning as $\mathbf{x}$ and with the syntax structure $\mathbf{s}$. We propose the syntax controlled paraphrase generation method with considerations of semantics and syntax, as the architecture illustrated in Fig.1, which consists of three neural network parts. The syntax encoder encodes the given dependency tree $\mathbf{s}$ as the syntax guidance in the paraphrase generation. The original sentence $\mathbf{x}$ is encoded with a pretrained text encoder as the semantics guidance. The text generator generates the target sentence under the above guidance.

Text Encoder It encodes the given original sentence into a semantic representation, denoted by z_x. There are a few neural methods applicable for the text encoder, such as BERT or GPT, etc. It is preferable that the text encoders are pretrained on a large corpus to ensure the effectiveness on representing text semantics. In our implementation we pretrained a BiRNN based auto encoder on WikiText dataset as a baseline. There are also scenarios requiring a domain-specific text encoder for better text representations, which would be discussed in the experiments.

$$z_x = TextEnc(\mathbf{x}) \tag{6}$$

Dependency Syntax Encoder The dependency parsing tree of the target sentence is given in the form of $\mathbf{s} = (s_1, s_2, ..., s_m)$. Each s_i corresponds to a word placeholder in the target sentence. $s_i =< h_i, rel_i >$, where h_i is the head index of word i, i.e. i pointing to h_i with the dependency relation rel_i. When h_i is zero, word i is the root word of the sentence. This form matches to a unique dependency parsing tree, where each node corresponds to a word in the generated sentence.

We design two trainable lookup tables to embed the position information h_i and the relation information rel_i, denoted as E_{pos} and E_{rel}, respectively. For each s_i, the embedding is the concatenation of $E_{pos}(h_i)$ and $E_{rel}(rel_i)$, which is then fed to a multi layer bidirectional recurrent neural network. This information guides the generated words in the sentence following the explicit syntax relationships.

$$e_{s,i} = E_{pos}(h_i) \oplus E_{rel}(rel_i) \tag{7}$$
$$H_{syn} = BiGRU(E_{syn}) \tag{8}$$

Paraphrase Generator The paraphrase generator takes z_x and H_{syn} as inputs and generates the final paraphrase, which satisfies the constraints on the same meaning with the original sentence and the given syntax template. To enforce the given parsing tree in an explicit way, the generator has the same time steps as H_{syn}. We adopt the QKV attention scheme to extract information of syntax guidance on each time step i in the text generation process:

$$q_i = W^Q(E_{word}(y_{i-1}) \oplus z_x \oplus H_{syn,i}) + b^Q \tag{9}$$

$$k_i = W^K H_{syn,i} + b^K \tag{10}$$

$$v_i = W^V H_{syn,i} + b^V \tag{11}$$

$$\alpha_{ij} = \frac{\exp(q_i \cdot k_j)}{\sum_{p=1}^{m} \exp(q_i \cdot k_p)} \tag{12}$$

$$H_{att,i} = \sum_{j=1}^{m} \alpha_{ij} v_i \tag{13}$$

A one directional RNN decoder is chosen as the paraphrase generator that takes the concatenation of the following representations as inputs: 1) The embedding of the last generated word $E_{word}(y_{i-1})$, 2) The semantic representation of the original sentence z_x, 3) The syntax representation of the current step $H_{syn,i}$, 4) An attention fused syntax representation of the current step $H_{att,i}$, as the following equations. The first step takes a special token $< BOS >$ as input. The generated paraphrase is denoted by $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_m)$, which has the same length as $\mathbf{s}$.

$$x_{syn,i} = E_{word}(y_{i-1}) \oplus z_x \oplus H_{syn,i} \oplus H_{att,i} \tag{14}$$

$$\hat{\mathbf{y}} = RNN(X_{syn}) \tag{15}$$

Unsupervised Training with Hybrid Loss Functions

The unsupervised training process includes two parallel aspects: semantics and syntax enforced paraphrase generation. We design two loss functions to satisfy these two objectives and use this hybrid loss to train the paraphrase model. Since there is not parallel labeled data, the training is in an unsupervised way with the target of reconstructing the sentence and with the pretrained word composable knowledge.

Reconstruction Loss For the semantics enforcement, we design the reconstruction loss, which accepts the original sentence $\mathbf{x}$ and its syntax structure $\mathbf{s}_x$ as input, re-generate the original sentence $\mathbf{x}$ as the target, i.e. $\mathbf{y} = \mathbf{x}$. We use the negative likelihood of the words in the reference sentences that encourages the model to generate the target paraphrase.

$$loss_{rec} = -\sum_{i=1}^{m} \log p(\hat{y}_i = y_i | \mathbf{x}, \mathbf{s}_x, y_1, ...y_{i-1}) \tag{16}$$

Since the model is trained in an unsupervised scheme, it tends to choose the words that ever appeared in the original sentence. This phenomenon influences the diversity of paraphrase. Thus, we adopt the word dropout mechanism designed in Huang and Chang (2021) during unsupervised training. Words in the sentence $\mathbf{x}$ are randomly dropped, and the model is trained to generate a complete sentence.

Syntax Matching Loss We design a syntax matching loss function with the word composable knowledge, which evaluates how much the generated paraphrase follows the given dependency parsing tree in a word-to-sentence mode.

For the word-level syntax, the composable knowledge evaluates the soundness of a dependency triplet (h, r, t), i.e. the dependency relationship between two words, denoted by $p_s(h, r, t)$. It provides an explicit gradient for each generated word during paraphrasing. For a given dependency relationship $s_i = (j, rel_i) \in \mathbf{s}$ in the syntax template, each placeholder is replaced by the generated word corresponding to the word positions, and the syntactical soundness of the relationship is computed by $\log p_s(\hat{y}_j, rel_i, \hat{y}_i)$, where $\hat{y}_i, \hat{y}_j$ are the words at the positions i, j in the generated sentence, as shown in Fig.1. For efficient training, we adopt the ground truth word at the head position for each syntax relationship as follows. Since the pretrained word composable knowledge relies on the word embeddings, the Gumbel-Softmax trick(Jang, Gu, and Poole 2017) is adopted to make sure the loss function is differentiable and the model is trainable.

$$loss_{syn} = -\frac{1}{|s|} \sum_{s_i \in \mathbf{s}} \log p_s(y_j, rel_i, \hat{y}_i) \tag{17}$$

To ensure a sentence-level consistency to the syntax template in the generation, we further take the hierarchical structure of parsing tree into consideration. From the linguistic point of view, the more important syntactical role a word takes, the more it approaches the root. Thus, we define the loss function that assigns a higher weight to such words in addition to Eq.17. Let L_i denote the depth of a word w_i in the dependency parsing tree, and let L_{max} denote the max depth. The range of L_i is $[1, L_{max}]$ considering the virtual root node. The final syntax matching loss function is:

$$loss_{syn} = -\frac{1}{|s|} \sum_{s_i \in \mathbf{s}} w_i \log p_s(y_j, rel_i, \hat{y}_i) \tag{18}$$

$$w_i = \frac{\exp(L_{max} - L_i)}{\exp(L_{max})} \tag{19}$$

The complete hybrid loss function is as follows, where λ is the weight of the syntax matching loss.

$$loss = loss_{rec} + \lambda * loss_{syn} \tag{20}$$

Experiments

Datasets For training and evaluating the proposed method, we adopted two paraphrasing datasets: Quora and SimpleWiki. The Quora (Iyer et al. 2017) dataset is a widely used dataset for paraphrase generation. The source and target sentences are both questions from Quora, which are identified as the same question by the users. The Simplewiki (Coster and Kauchak 2011) is a dataset for text simplification. It is generated from aligning the English Wikipedia with Simple English Wikipedia.

To be mentioned here, both datasets are parallel paraphrase datasets. Since our method adopts unsupervised training, the source and target sentence in the original training set are used separately as unparalleled training samples.

Datasets	#Train	#Test	#Tokens
Quora	134336	14927	3338818
SimpleWiki	123506	13723	6147602
WikiText	3544073	N/A	98640547
Universal Dependency	200159	N/A	N/A

Table 1: Statistics of datasets

For pretraining the word composable knowledge, we use the English Web Treebank from Universal Dependencies [1] to obtain high quality syntax paring annotations. For pretraining semantic encoder, we used the WikiText corpus which contains 98 million tokens. We also pretrained semantic encoder on the training set of each dataset and conducted experiments, which will be discussed later. The statistics of used dataset are given in Table.1.

Evaluation Tasks and Metrics The evaluated task is syntax controlled paraphrase generation. Similar to previous works, for each test paraphrase pair $(\mathbf{x}, \mathbf{y})$, the source sentence $\mathbf{x}$ and the dependency parsing structure $\mathbf{s}_y$ of the target sentence $\mathbf{y}$ are used as inputs of the model. The model output $\hat{\mathbf{y}}$ is expected to be similar with $\mathbf{y}$. The following metrics are adopted for evaluating the generated paraphrase:

- **BLEU** (Papineni et al. 2002). We adopt two metrics, BLEU-ref is computed against the target sentences and BLEU-ori against the source sentence.
- **BertScore** (Zhang et al. 2019a). We evaluate the F1 BertScore of the paraphrase against the original sentence, since embedding based evaluation is more accurate than n-gram based metrics in terms of semantic similarity.
- **DSM** (Dependency Syntax Matching). It is a metric designed by us to evaluate whether the paraphrase conforms the given dependency syntax tree. This metric is inspired by the evaluations of dependency parser. For a generated sentence $\hat{\mathbf{y}}$ given $\mathbf{x}, \mathbf{s}_y$ as input, we first get its dependency parsing tree $\mathbf{s}_{\hat{y}}$. Then we evaluate the UAS and LAS score (Dozat and Manning 2016) between $\mathbf{s}_{\hat{y}}$ and $\mathbf{s}_y$ as the evaluation of syntax template matching.
- **SSM** (Soft Syntax Matching). We also proposed a relaxed syntax matching metric, since most compared methods don't guarantee the length of generated paraphrase. Considering the cases that parts of the ground truth appear in a different position of the generated paraphrase, we match syntax relationships independently and only consider the relation type and the depth. Define the matching score between two syntax relationships $s_i = (r_i, l_i) \in \mathbf{s}_y$ and $s_j = (r_j, l_j) \in \mathbf{s}_{\hat{y}}$, where $r.$ denotes the syntax relation type and $l.$ denotes the depth of the syntax relationship.

$$Score(s_i, s_j) = \begin{cases} 1, & r_i = r_j, l_i = l_j \\ \frac{1}{|l_i - l_j| + 1}, & r_i = r_j, l_i \neq l_j \\ 0, & Otherwise \end{cases} \quad (21)$$

We compute the SSM score by first finding matches from $\mathbf{s}_{\hat{y}}$ for each edge in $\mathbf{s}_y$, similar to the recall score, and then compute an average of the matching scores in a sentence.

[1] https://universaldependencies.org

Comparison Methods The compared methods are all capable of integrating syntax control information in paraphrase generating. Most methods adopt top levels of the constituency parsing tree as syntax template in their original implementation. To make fair comparison, we use the full constituency parsing tree in the following experiments.

- **Copy**. A baseline method which copies the original sentence as a paraphrase. It serves as a quantified view of the characteristics of the original dataset.
- **SUP** (Yang et al. 2021). A syntactically-informed unsupervised paraphrasing model based on conditional variational auto-encoder (CVAE). The authors designed a two-phase training process to improve the model's ability to disentangle semantics and syntax without parallel training data.
- **SynPG** (Huang and Chang 2021). A syntax controlled unsupervised paraphrasing model using an encoder-decoder structure, which disentangles the semantics and syntax of a given sentence. The results with models trained by us are denoted by **SynPG**. The authors also provided model parameters pretrained on ParaNMT dataset, whose results are denoted by **SynPG-PT**.
- **SCPN** (Iyyer et al. 2018). A syntactically controlled supervised paraphrase model based on an encoder-decoder structure. It also adopted copy mechanism (See, Liu, and Manning 2017) in generating paraphrase, which will copy words directly from the original sentence.
- **CKPara**. Our method, and the variants:
  - **-TreeLvl** uses Eq.17 instead of Eq.18 for syntax matching loss.
  - **-SynLoss** is trained without the syntax matching loss.
  - **+SemEnc** uses text encoder pretained on the corresponding dataset instead of a large corpus.

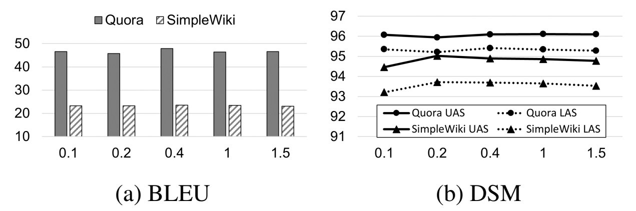
Implementation Our method is implemented with PyTorch 1.10.2 with CUDA 11.3. The dependency parsing trees used are generated with Stanza 1.4.0 (Qi et al. 2020). The text encoder uses 300 dimension GloVe (Pennington, Socher, and Manning 2014) embeddings and produces a 300 dimension vector as the semantic representation. For the parsing tree encoder, the dimension size of syntax element embeddings is chosen from $\{150, 200, 300, 500, 750\}$. For the paraphrase generator, the dimension size of the RNN hidden vector is chosen from $\{150, 200, 300, 500, 750\}$. The weight of the syntax matching loss is chosen from $[0, 1.5]$. The performance results are chosen from the best parameter combinations, while the model analysis and ablation study experiments are conducted with both dimension sizes set to 300 and the weight set to 0.2 if not stated otherwise.

Performance Comparison

The performances of methods on both datasets are shown in Table.2. First as indicated by the performance of baseline Copy, the characteristics of the datasets are very different. The BLEU-ref on SimpleWiki is much higher than on

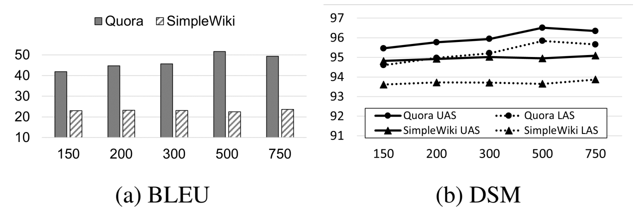
Model	DSM-UAS(↑)	DSM-LAS(↑)	SSM(↑)	BLEU-ref(↑)	BLEU-ori(↓)	BertScore(↑)	Br/Bo(↑)
			Quora				
Copy	14.02	12.69	71.27	31.55	91.47	**95.94**	0.34
SUP	63.25	59.48	81.21	34.48	28.83	90.18	1.20
SynPG-PT	77.1	75.18	87.24	28.27	**19.95**	89.19	1.42
SynPG	69.27	67.48	88.08	40.01	23.77	89.09	1.68
CKPara	**96.51**	**95.84**	**96.6**	**51.69**	24.39	89.9	**2.12**
-TreeLvl	96.04	95.32	96.1	46.48	23.95	89.76	1.94
-SynLoss	96.09	95.37	96.04	45.87	24.89	89.9	1.84
+SemEnc	94.67	93.44	94.54	41.58	28.78	**90.25**	1.44
SCPN *	79.4	78.26	90.34	**55.99**	22.84	90.23	**2.45**
			SimpleWiki				
Copy	34.91	34.37	80.75	**57.08**	95.86	**97.69**	0.60
SUP	2.76	1.62	57.42	10.58	15.02	85.35	0.70
SynPG-PT	27.45	26.7	69.14	14.12	**10.5**	86	1.34
SynPG	32.56	31.57	84.83	15.31	15.43	86.25	0.99
CKPara	**94.97**	**93.74**	**93.39**	**27.51**	19.24	**88.13**	**1.43**
-TreeLvl	94.62	93.27	93.27	22.65	13.61	87.16	**1.66**
-SynLoss	94.96	**93.8**	**93.5**	22.86	14.29	87.33	1.60
+SemEnc	94.75	93.38	93.11	**28.56**	20.32	**88.31**	1.41
SCPN *	53.4	52.58	81.06	**53.19**	43.54	**90.92**	1.22

Table 2: Paraphrasing performance comparison. Br/Bo is the ratio of BLEU-ref against BLEU-ori. Methods marked with star * are supervised paraphrasing models to serve as a reference upper bound.



(a) BLEU (b) DSM

Figure 2: The influence of syntax matching loss weight λ

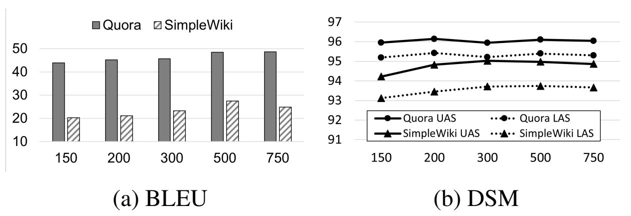


(a) BLEU (b) DSM

Figure 3: The influence of syntax embedding dimension

Quora. The target sentence of SimpleWiki is often a sentence with easier words and simpler syntax structure and thus has a high n-gram overlap with the original sentence.

The results show a significant advantage of our proposed method over compared unsupervised methods both semantically and syntactically, which shows the effectiveness of the word composable knowledge on dependency syntax. On Quora our method approaches the supervised method SCPN, which is a very promising performance. On SimpleWiki, our method performs better than unsupervised method but worse than the Copy baseline. This is probably due to that there are many proper nouns and entities in the dataset. The com-



(a) BLEU (b) DSM

Figure 4: The influence of the hidden dimension in the paraphrase generator

pared constituency parsing based methods tend to expand the structure and generate redundant contents. Such result shows the potential educational application of our method.

Model Analysis

Hyper-parameter Influence We performed experiments to analyze how the hyper-parameters affect the performance of our model to have a better understanding on the influential factors in our method. The tested hyper-parameters are the weight of the syntax matching loss λ, the dimension of syntax element embeddings E_{pos}, E_{rel} in the dependency syntax encoder, and the dimension of the RNN in the paraphrase generator. The results are shown in Fig.2 to 4. We choose the metrics BLEU-ref to reflect the overall paraphrase quality and DSM to reflect the ability of syntax matching. In the figures of DSM, the solid line is used for UAS and the dot line for LAS, and the statistics of Quora are marked by circle and SimpleWiki by triangle.

The weight of syntax matching loss. As shown in Fig.2a, on the BLUE-ref metric, the model performs best when λ is set to 0.4 on both datasets. It can also be seen in Fig.2b

Model	Sentence 1 (Quora)	Sentence 2 (SimpleWiki)
Source Sent.	what workout clothes did guys wear in the summer back in the year 1990?	a legal case is a dispute between opposing parties resolved by a court, or by some equivalent legal process.
Reference	what is the best outfit a guy would wear for working out in the summer like it's the year 1990?	a legal case is a dispute between two parties that is resolved by a court or other legal process.
CKPara	what is a good suit a guy would wear for working out in the summer like it's the year 1990?	a legal case is a dispute between two parties that is determined by a court or other legal process.
SynPG	what is the best the the workout wear to wear in front of the year 1990 in the year 1990?	the same question is a dispute between two parties that is resolved by the court or legal legal court.
SUP	what 's the best workout a collection would happen to holding down in the summer if he lots the 1990?	legal legal dispute is a legal legal legal dispute between legal legal legal legal legal legal legal legal legal legal legal legal.

Table 3: Case study from two datasets

Method	Quora		SimpleWiki	
	Overall	Syntax	Ovarall	Syntax
CKPara	1.03	0.95	0.45	0.95
SynPG	0.87	0.91	0.2	0.77
SUP	0.85	0.86	0.17	0.48

Table 4: Score of human evaluation on both datasets

that the higher λ leads to better DSM as matching the given syntax parsing tree is emphasized.

The syntax embedding dimension. As shown in Fig.3, on Quora the model performs better with larger dimension size, with the peak at 500 dimensions. While on the SimpleWiki dataset, the performance on BLEU-ref drops. This is probably due to the target sentences of the SimpleWiki dataset have simpler syntax structure, and thus the difficulty lies more on preserving the meaning of the original sentence.

The hidden dimension in paraphrase generator. On both datasets, increasing the dimension size leads to higher performance on BLEU-ref and slightly better DSM as shown in Fig.4. This indicates that the capacity of the paraphrase generator is vital for preserving the original meaning.

Ablation Study We also conducted ablation study to analyze the impact of different components in our model. The results are shown in Table.2. We can see that both -TreeLvl and -SynLoss has lower performance on DSM on both datasets, which shows the effectiveness of the syntax matching loss and the introduction of the hierarchical weights. The +SemEnc variant achieves better performance on SimpleWiki than on Quora. This is due to that the samples in SimpleWiki contains many proper nouns or OOV words. This shows that a semantic encoder pretrained in the corresponding domain is beneficial if the language has unique characteristics in that domain. Otherwise a text encoder pretrained on a large corpus is more preferred.

Human Evaluation

We also perform human evaluation to evaluate the quality of paraphrases. Following previous works (Yang et al. 2021), 100 generated samples by each method are randomly selected from testing sets. Each sample is evaluated by three annotators according to the following scoring: the generated sentence is 0) not a valid paraphrase; 1) a paraphrase with grammatical errors; 2) grammatically good paraphrase. A 0/1 syntax matching score is also given separately to show the model's ability on matching the given syntax structure.

The results are shown in Table.4. Our method achieves better results than compared methods in both overall score and syntax matching score. The overall score on SimpleWiki is much lower than Quora. As described above, the samples in SimpleWiki often contain proper nouns. Thus, using the wrong word is considered invalid paraphrase though the syntax of generated paraphrase matches the reference sentence.

In Table.3, examples from both datasets are shown to provide an intuitive view of the generated paraphrases. Benefiting from the explicit syntax guidance, our method generates paraphrases with correct syntax and without the problem of repeating. In sentence 2 the paraphrase generated by our method used a different verb word, which may be considered a false paraphrase in a professional domain. The performance will be further improved if the copy mechanism (See, Liu, and Manning 2017) is used.

Conclusion

In this paper, we proposed an unsupervised syntax controlled paraphrase generation method **CKPara**. It generates syntax controlled paraphrase of a sentence based on a dependency parsing tree as the form of syntax guidance. We pretrained word composable knowledge to learn the word composition patterns. We design a hybrid syntax matching loss, which evaluates the word-level syntax soundness of the paraphrases using the word composable knowledge, and ensures the sentence-level consistency with the given syntax template using a hierarchical structure. Experiments are conducted thoroughly on two real world paraphrase datasets, where the results demonstrate that the proposed method performs better than existing methods in terms of both preserving the original meaning and matching the given syntax template. We also conducted detailed analysis of our method on both datasets. The performance of **CKPara** shows high potential in text simplification and education-related text generation where grammar correctness is well emphasized.

Acknowledgments

This work was supported by the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007).

References

Barzilay, R.; and Lee, L. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. *arXiv preprint cs/0304006*.

Bowman, S. R.; Gupta, R.; Gauthier, J.; Manning, C. D.; Rastogi, A.; and Potts, C. 2016. A fast unified model for parsing and sentence understanding. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, 1466–1477. Association for Computational Linguistics (ACL).

Casas, N.; Fonollosa, J. A.; and Costa-jussà, M. R. 2020. Syntax-driven Iterative Expansion Language Models for Controllable Text Generation. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, 1–10.

Coster, W.; and Kauchak, D. 2011. Simple English Wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 665–669.

Dozat, T.; and Manning, C. D. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Egonmwan, E.; and Chali, Y. 2019. Transformer and seq2seq model for Paraphrase Generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 249–255. Hong Kong: Association for Computational Linguistics.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 297–304. JMLR Workshop and Conference Proceedings.

Huang, K.-H.; and Chang, K.-W. 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: Main Volume*, 1022–1033.

Iyer, S.; Dandekar, N.; Csernai, K.; et al. 2017. First quora dataset release: Question pairs. http://data.quora.com.

Iyyer, M.; Wieting, J.; Gimpel, K.; and Zettlemoyer, L. 2018. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *Proceedings of NAACL-HLT*, 1875–1885.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparametrization with Gumble-Softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net.

Kauchak, D.; and Barzilay, R. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, 455–462.

Kumar, A.; Ahuja, K.; Vadapalli, R.; and Talukdar, P. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8: 330–345.

Li, Z.; Jiang, X.; Shang, L.; and Liu, Q. 2019. Decomposable Neural Paraphrase Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3403–3414.

Liu, M.; Yang, E.; Xiong, D.; Zhang, Y.; Meng, Y.; Hu, C.; Xu, J.; and Chen, Y. 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*, 2310–2321.

McKeown, K. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1): 1–10.

Miao, N.; Zhou, H.; Mou, L.; Yan, R.; and Li, L. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6834–6842.

Pan, W.; Liu, T.; Zhang, W.; and Sun, Y. 2021. Learning New Word Semantics with Conceptual Text. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; and Manning, C. D. 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*.

Roy, A.; and Grangier, D. 2019. Unsupervised Paraphrasing without Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6033–6039.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083.

Wieting, J.; and Gimpel, K. 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 (Volume 1: Long Papers)*, 451–462.

Wieting, J.; Mallinson, J.; and Gimpel, K. 2017. Learning Paraphrastic Sentence Embeddings from Back-Translated Bitext. In *EMNLP 2017: Conference on Empirical Methods in Natural Language Processing*, 274–285. Association for Computational Linguistics.

Xie, Y.; Li, W.; Sun, Y.; Bertino, E.; and Gong, B. 2022. Subspace Embedding Based New Paper Recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 1767–1780. IEEE.

Yang, E.; Liu, M.; Xiong, D.; Zhang, Y.; Meng, Y.; Hu, C.; Xu, J.; and Chen, Y. 2021. Syntactically-Informed Unsupervised Paraphrasing with Non-Parallel Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2594–2604.

Yuan, W.; Ding, L.; Meng, K.; and Liu, G. 2021. Text Generation with Syntax-Enhanced Variational Autoencoder. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019a. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Zhang, X.; Yang, Y.; Yuan, S.; Shen, D.; and Carin, L. 2019b. Syntax-Infused Variational Autoencoder for Text Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2069–2078.

Zhao, S.; Meng, R.; He, D.; Saptono, A.; and Parmanto, B. 2018. Integrating Transformer and Paraphrase Rules for Sentence Simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3164–3173.

Zhou, Z.; Sperber, M.; and Waibel, A. 2019. Paraphrases as Foreign Languages in Multilingual Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 113–122.