Keep the Primary, Rewrite the Secondary: A Two-Stage Approach for Paraphrase Generation

Yixuan Su[♠] **David Vandyke**[♣] **Simon Baker**[♠] **Yan Wang**[♢] **Nigel Collier**[♠]
[♠]Language Technology Lab, University of Cambridge
[♣]Apple
[♢]Tencent AI Lab
{ys484,sb895,nhc30}@cam.ac.uk
dvandyke@apple.com, brandenwang@tencent.com

Abstract

Paraphrase generation is an important and challenging NLG problem. In this work, we propose a new Identification-then-Aggregation (IA) framework to tackle this task. In the identification step, the input tokens are sorted into two groups by a novel Primary/Secondary Identification (PSI) algorithm. In the aggregation step, these groups are separately encoded, before being aggregated by a custom designed decoder, which autoregressively generates the paraphrased sentence. In extensive experiments on two benchmark datasets, we demonstrate that our model outperforms previous studies by a notable margin. We also show that the proposed approach can generate paraphrases in an interpretable and controllable way.

1 Introduction

Paraphrases refer to text (often sentences) that share the same meaning but use different choices of words and their ordering. Automatic generation of paraphrases is a longstanding problem that is important to many downstream NLP applications such as question answering (Dong et al., 2017; Buck et al., 2018), machine translation (Cho et al., 2014), and semantic parsing (Su and Yan, 2017). Most early research adopts the sequence-to-sequence model (Prakash et al., 2016; Cao et al., 2017; Li et al., 2018) to map the input text to its paraphrase by processing and generating each word in a uniform way. Rather than processing each word uniformly, some recent studies tackles this task in a decomposable manner. For instance, Li et al. (2019) adopt an external word aligner to extract paraphrasing patterns at different levels of granularity and then perform generation. Fu et al. (2019) first use source words to predict their neighbors and then organize the predicted neighbors into a complete sentence.

How are baby elephants called? What is a name for a baby elephant?
What is the best way to use time? How can we utilize time effectively?
Many people are relaxing on the pretty beach. A group of people relaxing on the sand at a beach.
A refrigerator and a stove on a street. A yellow stove and refrigerator and a building.

Figure 1: Examples of paraphrase pair sampled from Quora and MSCOCO datasets in which the words in red refer to the primary content and the rest of the words make up the secondary content.

In this work, we investigate decomposable paraphrase generation from a different perspective. Specifically, we consider using a non-parametric approach to label each token in an input sentence as either (i) primary, or (ii) secondary. Intuitively, the primary content of a sentence refers to the factual information that defines the shared meaning of the paraphrase pair. All other content is deemed as secondary, and typically controls the structure of the sentence. In practice, this distinction is determined by an algorithm that decides whether tokens are primary or secondary, as described in §3. To better illustrate our idea, in Figure 1, we show some examples sampled from Quora and MSCOCO (Lin et al., 2014) datasets. We see that, for many cases, the paraphrase pairs maintain the similar primary content (e.g., the phrases "*baby elephants*" and "*baby elephant*" in the first example) while the secondary content can be rephrased in several different ways.

Based on the above observation, we propose an Identification-then-Aggregation (IA) framework to address the paraphrase generation task. Given an input sentence, generating a paraphrase follows a two-stage process. First, the primary and secondary content of the input sentence is identified via a novel Primary/Secondary Identification (PSI) algo-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 560–569

rithm which is based on a common non-parametric rank coefficient. Second, a new neural paraphrase generation model aggregates the identified information and generates the result. Specifically, the proposed model consists of (1) two encoders which separately process the identified primary and secondary content; and (2) an aggregation decoder which integrates the processed results and generates the paraphrased sentence.

We test the proposed approach on two benchmark datasets with automatic and human evaluation. The results show that our approach outperforms previous studies and can generate paraphrases in an interpretable and controllable way.

2 Related Work

The automatic generation of paraphrases is important for many downstream NLP applications and it has attracted a number of different approaches. Early researches included rule-based approaches (McKeown, 1979; Meteer and Shaked, 1988) and data-driven methods (Madnani and Dorr, 2010). With the advances of neural networks, recent approaches tackle this problem by treating it as a sequence-to-sequence language generation task. Prakash et al. (2016) proposed to modify the networks structure to improve the generation quality. Cao et al. (2017), Wang et al. (2019), and Kazemnejad et al. (2020) proposed to improve the model performance by leveraging external resources, including phrase dictionary, semantic annotations, and an off-the-shelf pre-trained neural retriever. Other works proposed to adopt techniques like reinforcement learning (Li et al., 2018) and unsupervised learning (Roy and Grangier, 2019) for this task.

While achieving satisfactory results, these above methods do not offer users the way to control the generation process in a fine-grained way. To incorporate controllability into the generation model, different approaches have been proposed. Iyyer et al. (2018) trained the model to produce the paraphrased sentence with a given syntax. Li et al. (2019) proposed to adopt an external word aligner to train the model to generate paraphrases from different levels. In Fu et al. (2019)'s work, the model generates paraphrases by planning the neighbour of words and realizing the complete sentence.

3 Primary/Secondary Identification

Given an input sentence, our goal is to identify the primary content that are likely to appear in the paraphrased sentence. To this end, we propose a Primary/Secondary Identification (PSI) approach which dynamically evaluates the importance of different parts of the input sentence. The parts with high importance are deemed the primary content, while the rest parts are deemed secondary content.

Token Importance Formally, given a paraphrase pair $\mathbf{X}$ and $\mathbf{Y}$, we define their pairwise similarity as $\mathcal{F}(\mathbf{X}, \mathbf{Y})$. To determine the importance of the i-th token x_i of $\mathbf{X}$ in relation to $\mathbf{Y}$, we first compute the pairwise similarity between $\mathbf{X}' = \mathbf{X} \ominus x_i$ and $\mathbf{Y}$ as $\mathcal{F}(\mathbf{X}', \mathbf{Y})$, where the $\ominus$ operator removes the token x_i from $\mathbf{X}$. We assume that if the token x_i belongs to the primary content that is maintained in both $\mathbf{X}$ and $\mathbf{Y}$, then removing it from $\mathbf{X}$ will cause a significant drop in the pairwise similarity between $\mathbf{X}$ and $\mathbf{Y}$. Based on this assumption, we measure the importance of x_i as the ratio of change in the pairwise similarity score as

$$\mathcal{G}(x_i; \mathbf{X}, \mathbf{Y}) = \frac{\mathcal{F}(\mathbf{X}, \mathbf{Y}) - \mathcal{F}(\mathbf{X}', \mathbf{Y})}{\mathcal{F}(\mathbf{X}, \mathbf{Y})}. \quad (1)$$

Intuitively, a higher $\mathcal{G}(x_i; \mathbf{X}, \mathbf{Y})$ means a larger decrease in the pairwise similarity, indicating a higher importance of the token x_i and vice versa.

Similarity Measurement We now describe the details of the function $\mathcal{F}(\cdot, \cdot)$. Inspired by Zheleznyak et al. (2019), we measure the pairwise similarity between $\mathbf{X}$ and $\mathbf{Y}$ based on a non-parametric rank correlation coefficient. Specifically, given $\mathbf{X}$ and $\mathbf{Y}$, we first transform them into the representation matrices $\mathcal{M}(\mathbf{X}) \in \mathbb{R}^{|\mathbf{X}| \times D}$ and $\mathcal{M}(\mathbf{Y}) \in \mathbb{R}^{|\mathbf{Y}| \times D}$ via a D-dimensional pretrained embeddings. Then, the matrices are mapped into fixed size context vectors $\hat{x} \in \mathbb{R}^{1 \times D}$ and $\hat{y} \in \mathbb{R}^{1 \times D}$ via an element-wise max-pooling operation. Finally, the pairwise similarity $\mathcal{F}(\mathbf{X}, \mathbf{Y})$ is measured using Spearman's correlation coefficient $\hat{\rho}$ of the context vectors $\hat{x}$ and $\hat{y}$ as

$$\mathcal{F}(\mathbf{X}, \mathbf{Y}) = 1 - \frac{6 \times \sum_{j=1}^{D} (r[\hat{x}_j] - r[\hat{y}_j])^2}{D \times (D^2 - 1)} \quad (2)$$

where $r[\hat{x}_j]$ denotes the integer rank of $\hat{x}_j$ in the context vector $\hat{x}$ (similarly $r[\hat{y}_j]$).

For a better illustration, in Table 1, we show sentence sampled from Quora and MSCOCO datasets along with their pairwise similarities. We see that the numerical results are highly correlated with human judgement which empirically demonstrate the effectiveness of our measurement method.

Sentence 1	Sentence 2	Pairwise Similarity
What are the best games you can play with just paper?	Which games can you play on paper in your free time with your friends?	0.745
	What can I do to become a professional chess player?	0.438
	Should animals be used for testing medicines and drugs?	0.140
Three bears walking on dried grass toward the green wooded area.	Two brown bears walking through a green, grassy area.	0.743
	A simple plain clear vase with a dead twig and water inside.	0.439
	A man using a phone next to a motorcycle.	0.128

Table 1: Examples of different sentence pairs $(\mathbf{X}, \mathbf{Y})$ and their corresponding pairwise similarity scores $\mathcal{F}(\mathbf{X}, \mathbf{Y})$.

Algorithm 1: Primary/Secondary Identification

Input : Input sentence $\mathbf{X} = (x_1, ..., x_{|\mathbf{X}|})$;
Paraphrased sentence $\mathbf{Y} = (y_1, ..., y_{|\mathbf{Y}|})$;
Primary content threshold α_p;
Importance measurement function $\mathcal{G}(\cdot, \cdot, \cdot)$.

$\mathbf{X}_p \leftarrow \{\}$; $\mathbf{X}_s \leftarrow \{\}$;
for $i = 1$ *to* N **do**
  $\mathbf{X}' \leftarrow \mathbf{X} \ominus x_i$;
  if $\mathcal{G}(x_i; \mathbf{X}, \mathbf{Y}) > \alpha_p$ **then**
    $\mathbf{X}_p \leftarrow x_i$ and $\mathbf{X}_s \leftarrow$ [MASK];
  else
    $\mathbf{X}_p \leftarrow$ [MASK] and $\mathbf{X}_s \leftarrow x_i$;
  end
end
$\mathbf{X}_p \leftarrow$ joinmask$(\mathbf{X}_p)$; $\mathbf{X}_s \leftarrow$ joinmask$(\mathbf{X}_s)$.;

Output : Primary Content $\mathbf{X}_p$;
Secondary Content $\mathbf{X}_s$.

Putting this together, the detailed description for splitting the input sentence $\mathbf{X}$ into the primary content $\mathbf{X}_p$ and secondary content $\mathbf{X}_s$ is given in Algorithm 1, where the token [MASK] is used as a special placeholder and the threshold α_p is tuned based on the performance on the validation set[1]. The joinmask$(\cdot)$ operation joins consecutive [MASK] tokens into a single [MASK] token. We note that the incorporation of the [MASK] token is crucial. Because, in this way, the generation model could have access to the original source sentence structure by simply overlapping the primary and secondary content. In the experiments, we found that removing [MASK] from the identified content causes a significant drop in model performance as the model can no longer have access to the original sentence structure.

In Figure 2, we show the computed results from PSI of an example presented in Figure 1. We can see that the primary content is effectively identified.

Inference During inference, given an input sentence, the primary and secondary content could not be directly identified as $\mathbf{X}_p, \mathbf{X}_s = \text{PSI}(\mathbf{X}, \mathbf{Y})$ since we do not have access to the target sentence $\mathbf{Y}$. To this end, we propose two alternative approaches. For the first one, we simply identify the primary and secondary content using the input sentence $\mathbf{X}$ as $\mathbf{X}'_p, \mathbf{X}'_s = \text{PSI}(\mathbf{X}, \mathbf{X})$. For the second one, we train a neural sequence tagger $\mathcal{S}$ based on the labels provided by $\text{PSI}(\mathbf{X}, \mathbf{Y})$. Then we extract the content using the input $\mathbf{X}$ as $\mathbf{X}'_p, \mathbf{X}'_s = \mathcal{S}(\mathbf{X})$. In the experiment section, we provide more detailed comparisons between these approaches.

[1]In this work, we set α_p as 0.1 for all experiments based on the model performance on the validation set.

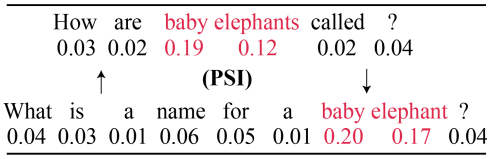

Figure 2: For each token, the score from the PSI algorithm is presented. The words in red is the identified primary content and the rest words make up the secondary content.

4 Neural Paraphrase Generator

Overview Given the input sentence $\mathbf{X}$, it is first partitioned into the primary and secondary content using the PSI algorithm. Then the identified content is independently processed by the primary encoder and the secondary encoder. Finally, an aggregation decoder integrates the outputs from both encoders and generates the result. In Figure 3, we provide an illustration of the proposed framework.

Encoder Stacks In this work, we use the transformer architecture (Vaswani et al., 2017) to construct the primary and secondary encoders. Formally, the Multi-Head Attention is defined as $\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are query, key and value. Each encoder has N_E layers. Given the input $\mathbf{X}$, the first layer operates as

$$\mathbf{V}_{\mathbf{X}}^{(1)} = \text{MultiHead}(E(\mathbf{X}), E(\mathbf{X}), E(\mathbf{X})), \quad (3)$$

$$\mathbf{O}_{\mathbf{X}}^{(1)} = \text{FFN}(\mathbf{V}_{\mathbf{X}}^{(1)}), \quad (4)$$

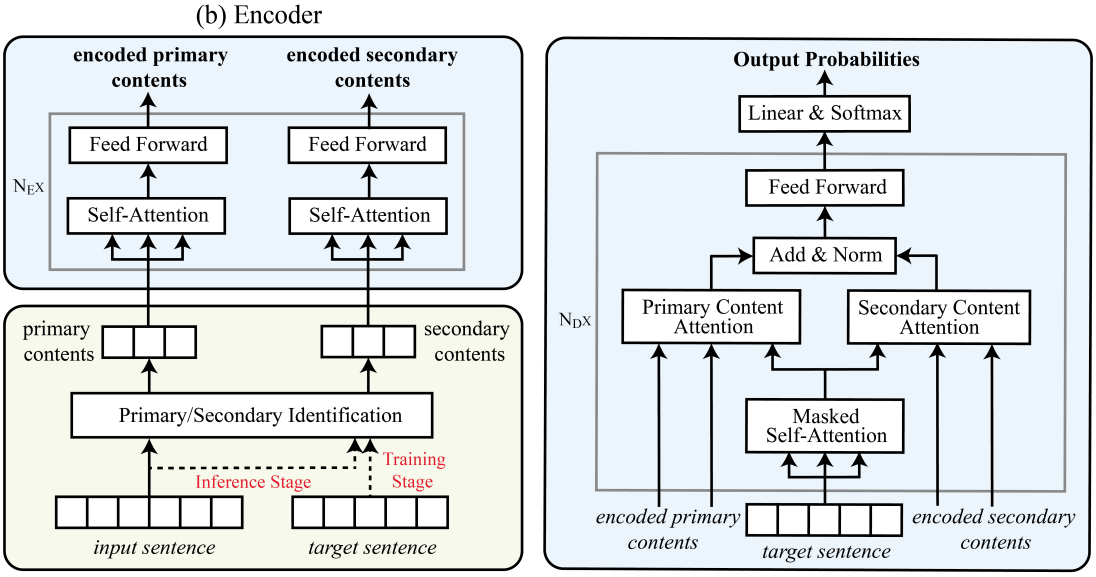

Figure 3: Overview of the proposed framework: *Italic* and **Boldface** denote the inputs and outputs at each stage. For a better illustration, we separately draw the encoder and decoder of the paraphrase generator. **(a)** During training, the primary and secondary content of the input sentence are identified by the PSI algorithm using the input and target sentence. During inference, the content is identified solely based on the input sentence. **(b)** The identified results are then encoded by separate encoders. **(c)** The aggregation decoder takes the encoded primary and secondary content as input and produces the probability of the target sentence. It should be noted that, during the training stage, the encoder and decoder are jointly trained in an end-to-end fashion.

where $E(\mathbf{X})$ is the input sequence embedding and $\text{FFN}(\cdot)$ is a feed-forward layer. For other layers:

$$\mathbf{V}_{\mathbf{X}}^{(n)} = \text{MultiHead}(\mathbf{O}_{\mathbf{X}}^{(n-1)}, \mathbf{O}_{\mathbf{X}}^{(n-1)}, \mathbf{O}_{\mathbf{X}}^{(n-1)}), \tag{5}$$

$$\mathbf{O}_{\mathbf{X}}^{(n)} = \text{FFN}(\mathbf{V}_{\mathbf{X}}^{(n)}), \tag{6}$$

where $n = 2, ..., N_E$.

Given the primary content $\mathbf{X}_p$ and secondary content $\mathbf{X}_s$ of the input sequence, their representations $\mathbf{O}_{\mathbf{X}_p}^{(N_E)} \in \mathbb{R}^{|\mathbf{X}_p| \times d}$ and $\mathbf{O}_{\mathbf{X}_s}^{(N_E)} \in \mathbb{R}^{|\mathbf{X}_s| \times d}$ are computed by the primary and secondary encoder respectively and d is the model size.

Decoder Stacks We design an aggregation decoder to integrate information coming from both encoders. Given the target sentence $\mathbf{Y}$, it is first encoded via a masked multi-head attention as

$$\mathbf{V}_{\mathbf{Y}_m}^{(1)} = \text{Mask-MultiHead}(E(\mathbf{Y}), E(\mathbf{Y}), E(\mathbf{Y})). \tag{7}$$

Then, the primary content attention module takes the encoded primary content $\mathbf{O}_{\mathbf{X}_p}^{(N_E)}$ and $\mathbf{V}_{\mathbf{Y}_m}^{(1)}$ as input and produces the intermediate result $\mathbf{V}_{\mathbf{Y}_m^p}^{(1)}$ as

$$\mathbf{V}_{\mathbf{Y}_m^p}^{(1)} = \text{MultiHead}(\mathbf{V}_{\mathbf{Y}_m}^{(1)}, \mathbf{O}_{\mathbf{X}_p}^{(N_E)}, \mathbf{O}_{\mathbf{X}_p}^{(N_E)})). \tag{8}$$

Similarly, the result $\mathbf{V}_{\mathbf{Y}_m^s}^{(1)}$ from the encoded secondary content $\mathbf{O}_{\mathbf{X}_s}^{(N_E)}$ is computed as

$$\mathbf{V}_{\mathbf{Y}_m^s}^{(1)} = \text{MultiHead}(\mathbf{V}_{\mathbf{Y}_m}^{(1)}, \mathbf{O}_{\mathbf{X}_s}^{(N_E)}, \mathbf{O}_{\mathbf{X}_s}^{(N_E)})). \tag{9}$$

The first layer output $\mathbf{O}_{\mathbf{Y}}^{(1)}$ is then acquired as

$$\mathbf{V}_{\mathbf{Y}_i}^{(1)} = \text{LayerNorm}(\mathbf{V}_{\mathbf{Y}_m^p}^{(1)} + \mathbf{V}_{\mathbf{Y}_m^s}^{(1)}), \tag{10}$$

$$\mathbf{O}_{\mathbf{Y}}^{(1)} = \text{FFN}(\mathbf{V}_{\mathbf{Y}_i}^{(1)}). \tag{11}$$

The final output $\mathbf{O}_{\mathbf{Y}}^{(N_D)} \in \mathbb{R}^{|\mathbf{Y}| \times d}$ is computed via a stack of N_D layers. The final probability of $\mathbf{Y}$ is produced by a linear softmax operation.

Learning Finally, given the input primary content $\mathbf{X}_p$, secondary content $\mathbf{X}_s$ and the target sequence $\mathbf{Y}$, the learning objective is defined as

$$\mathcal{L} = \sum_{t=1}^{|\mathbf{Y}|} \log p(\mathbf{Y}_t | \mathbf{Y}_{<t}, \mathbf{X}_p, \mathbf{X}_s). \tag{12}$$

5 Datasets

We test our approach on two benchmark paraphrase generation datasets: (1) Quora dataset[2] and (2) MSCOCO dataset (Lin et al., 2014).

[2]https://www.kaggle.com/c/quora-question-pairs

The Quora dataset was developed for the task of duplicated question detection. Each data instance consists of one source sentence and one target sentence. In the experiment, we randomly select one sentence as the source and the other as the target.

The MSCOCO dataset was originally developed for the image captioning task. In this dataset, each image is associated with five human-written captions. Although there is no guarantee that these captions must be paraphrases as they could describe different objects in the image, most of these captions are generally close to each, therefore the overall quality of this dataset is favorable and it is widely used for the paraphrase generation task.

Following Li et al. (2019) and Fu et al. (2019), for the Quora dataset, we split the size of training, validation and test sets as 100k, 4k and 20k. The MSCOCO dataset is split into 93k, 4k and 20k. The maximum sentence length for these two datasets is set as 16. The vocabulary size of the Quora and MSCOCO datasest are set to be 8k and 11k.

6 Experiments

6.1 Model Comparisons

We compare the proposed model with several representative baselines, including Residual-LSTM (Prakash et al., 2016), β-VAE (Higgins et al., 2017), Transformer (Vaswani et al., 2017), DNPG (Li et al., 2019), LBOW-Topk and LBOW-Gumbel (Fu et al., 2019)[3]. To compare different inference approaches, three variants of our model are used.

IANet+X: Given the input sentence $\mathbf{X}$, this model extracts the primary and secondary content using the approximated PSI$(\mathbf{X}, \mathbf{X})$ algorithm. Then, the paraphrase generator produces the paraphrased sentence using the identified content.

IANet+$\mathcal{S}$: In this case, a neural sequence tagger $\mathcal{S}$ is first trained based on the labels provided by PSI$(\mathbf{X}, \mathbf{Y})$. During inference, the model extracts the primary and secondary content of the input as $\mathbf{X}'_p, \mathbf{X}'_s = \mathcal{S}(\mathbf{X})$ and then perform generation.

IANet+ref: In contrast to previous variants, this model obtains the primary and secondary content using the exact PSI$(\mathbf{X}, \mathbf{Y})$ algorithm against the reference $\mathbf{Y}$. The reason to include this model is that, besides our proposed alternatives, there are other options that we can use. We will explore these options in the future work. But by evaluating IANet+ref we can show an upper bound on how much could be improved in this way.

[3]The hyperparameter setups and optimization in all baseline models are the same as their original works. For methods that do not release their code, we directly use the results in their original papers.

6.2 Implementation Details

We implement our model with PyTorch (Paszke et al., 2017). For the primary and secondary encoders, we use a 3-layer transformer with model size of 256 and heads of 8. Since the decoder has to integrate the information from both encoders, we build it with a larger capacity. The number of layers is set to 4. The model size and the attention heads are set to be 512 and 8. For the sequence tagger $\mathcal{S}$ that is used in the IANet+$\mathcal{S}$ model, we use a 2-layer LSTM with hidden size of 512.

In the experiments, we adopt pretrained 300-dimensional FastText Embeddings (Bojanowski et al., 2017) to perform the PSI algorithm. During training , we use Adam (Kingma and Ba, 2015) to optimize our model with a learning rate of 1e-4. In all experiments, we set α_p in Algorithm 1 as 0.1 based on the performance on the validation set.

6.3 Evaluation Metrics

Following previous studies (Prakash et al., 2016; Fu et al., 2019; Li et al., 2019), we report results on several automatic metrics, including BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). All lower n-gram metrics (1-4 grams in BLEU and 1-2 grams in ROUGE) are reported. In addition, we include iBLEU (i-B) (Sun and Zhou, 2012) as another evaluation metric, which penalizes repeating the source sentence in its paraphrase.

6.4 Main Results

Table 2 lists the results on both datasets. We see that the transformer baseline already achieves pretty strong results. This is because the capacity of transformer model is large enough to fit the datasets quite well. Nonetheless, in most of the evaluation metrics, our model outperforms previous studies by a notable margin, demonstrating the effectiveness of the proposed approach.

By comparing different variants of our model, we see that IANet-ref achieves the best results on all metrics. This is expected as it uses the reference sentence in determining the primary and secondary content. It is worth emphasising that the IANet+ref model, like everything else, does not receive the target sentence, it just gets inputs $\mathbf{X}$ and has their primary and secondary content more accurately

Quora

Models	B-1	B-2	B-3	B-4	i-B	R-1	R-2	R-L
Residual-LSTM (Prakash et al., 2016)	53.59	39.49	30.25	23.69	15.93	55.10	33.86	53.61
β-VAE, $\beta = 10^{-4}$ (Higgins et al., 2017)	47.86	33.21	24.96	19.73	10.28	47.62	25.49	45.46
Transformer (Vaswani et al., 2017)	53.56	40.47	32.11	25.01	17.98	57.82	32.58	56.26
DNPG (Li et al., 2019)	-	-	-	25.03	18.01	**63.73**	**37.75**	-
LBOW-Topk (Fu et al., 2019)	55.79	42.03	32.71	26.17	19.03	58.79	34.57	56.43
LBOW-Gumbel (Fu et al., 2019)	55.75	41.96	32.66	26.14	18.97	58.60	34.47	56.23
IANet+$\mathbf{X}$	56.06	42.69	33.38	26.52	19.62	59.33	35.01	57.13
IANet+$\mathcal{S}$	**56.72**	**43.21**	**33.96**	**27.09**	**20.11**	59.98	36.02	**58.01**
IANet+ref (upperbound)	58.32	44.81	35.46	28.71	21.76	61.89	38.86	59.43

MSCOCO

Models	B-1	B-2	B-3	B-4	i-B	R-1	R-2	R-L
Residual-LSTM (Prakash et al., 2016)	70.24	48.65	34.04	23.66	18.72	41.07	15.26	37.35
β-VAE, $\beta = 10^{-4}$ (Higgins et al., 2017)	70.04	47.59	32.29	22.54	18.34	40.72	14.75	36.75
Transformer (Vaswani et al., 2017)	71.31	49.86	35.55	24.68	19.81	41.49	15.84	37.09
LBOW-Topk (Fu et al., 2019)	72.60	51.14	35.66	25.27	21.07	42.08	16.13	38.16
LBOW-Gumbel (Fu et al., 2019)	72.37	50.81	35.32	24.98	20.92	42.12	16.05	38.13
IANet+$\mathbf{X}$	72.10	52.22	37.39	26.06	21.28	43.81	16.35	39.65
IANet+$\mathcal{S}$	**73.01**	**53.09**	**38.12**	**26.90**	**22.03**	**44.66**	**17.13**	**40.58**
IANet+ref (upperbound)	75.29	55.09	41.01	29.65	24.72	46.36	19.13	42.23

Table 2: Evaluation results on the Quora and MSCOCO dataset. B for BLEU and R for ROUGE. Where possible we copy results from DNPG (Li et al., 2019) as they did not release their code.

identified. This suggests that the deomposition of our approach is beneficial, and further work can be focused more on the identification step. On the other hand, without using the target sentence, both IANet+$\mathbf{X}$ and IANet+$\mathcal{S}$ must use an approximated approach at the inference time, which inevitably introduces noise in the identified content. As a result, the performance is lower than IANet+ref. We will provide more analysis in the analysis section.

6.5 Human Evaluation

We also conduct a human evaluation to assess our model, using graders proficient in English from an internal grading platform. We randomly select 150 examples from the Quora test set and compare our model with three representative baselines[4]. Three annotators are asked to rate the generated results from different models on a 3-point Likert scale (0, 1, or 2) with respect to the following features[5]:

- **Fluency**: Whether the generated paraphrase is grammatically correct and easily understood.
- **Accuracy**: Whether the content in the generated paraphrase is consistent with the content in the original sentence.
- **Diversity**: Whether the generated sentence structure differs from the reference sentence.

	Fluency	Accuracy	Diversity
Agreement	0.582	0.543	0.498
Residual-LSTM	1.57	1.41	1.29
Transformer	1.63	1.50	1.40
LBOW-Topk	1.61	1.49	1.43
IANet+$\mathcal{S}$	**1.71**	**1.57**	**1.61**
Reference	1.85	1.78	1.68

Table 3: Human Evaluation Results

To measure the agreement between the annotators, we use the Fleiss′ kappa coefficient (Fleiss et al., 1971). The agreement results are shown in the first row of Table 3, indicating moderate agreement between annotators on all metrics.

From Table 3, we see that our model achieves the best result on all metrics, which demonstrates the effectiveness of the proposed approach. Especially, on the diversity metric, our model significantly outperforms other baselines (Sign Test, with p-value < 0.05) and performs comparably with the

[4]Because the authors of DNPG (Li et al., 2019) did not release their code, thus we are not able to reproduce their results and are not able to include this model in the human evaluation.

[5]More details of the human evaluation guideline can be found in the supplementary material.

	Rec.(%)	Prec.(%)	F1(%)	B-4	R-L
PSI($\mathbf{X}, \mathbf{X}$)	71.2	88.5	78.9	26.52	57.13
Tagger $\mathcal{S}$	78.1	90.1	83.7	27.09	58.01
PSI($\mathbf{X}, \mathbf{Y}$)	**87.4**	**91.3**	**89.3**	**28.71**	**59.43**

Table 4: Comparisons between the identification performances of different algorithms on Quora dataset

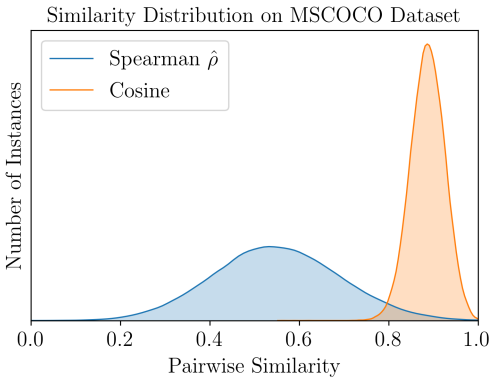

Figure 4: Comparison of similarity distribution

reference sentence (p-value $= 0.23$). The improvement in the diversity metric mainly comes from the two-step nature of our generation framework. By first determining which parts of the sentence to keep (primary) or to change (secondary), our model could then focus on maintaining the primary content while rewriting the secondary content, resulting in a more accurate and diverse paraphrase.

6.6 Further Analysis

In this section, we present further discussions and empirical analysis of the proposed approach.

6.6.1 Inference Algorithms Comparison

As shown in Table 2, IANet-ref outperforms IANet-$\mathbf{X}$ and IANet-$\mathcal{S}$ on both datasets. Our analysis is that IANet-ref could more accurately identify the primary content from the source comparing with the other variants. To provide more analysis, we separately use PSI($\mathbf{X}, \mathbf{X}$), the sequence tagger $\mathcal{S}$, and PSI($\mathbf{X}, \mathbf{Y}$) to identify the words that both appear in the source and target sentences in the Quora dataset. The results are shown in Table 4. We see that all three methods perform comparably on the precision (prec.) metric. However, PSI($\mathbf{X}, \mathbf{Y}$) significantly outperforms other methods on the recall (rec.) metric, showing that it can accurately extract more primary content from the source. From Table 4, we also observe that better identification results lead to better generation performances. Therefore, to improve the model performance, future work could be focused more on the identification step.

Metric	B-2	B-4	R-2	R-L
Cosine	50.34	24.91	14.78	37.62
Spearman $\hat{\rho}$	**53.09**	**26.90**	**17.13**	**40.58**

Table 5: Result comparison between the cosine similarity and the Spearman's $\hat{\rho}$ on the MSCOCO dataset.

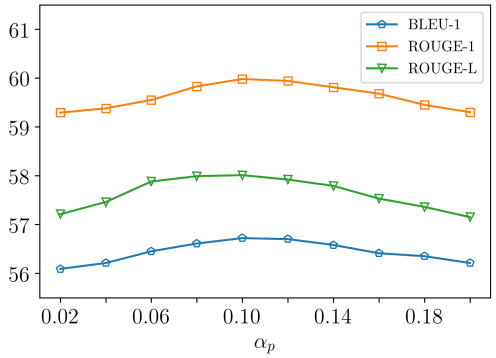

Figure 5: Effect of α_p on the model performance

6.6.2 Similarity Measurement Comparison

In this part, we analyze the differences between different similarity measurements. As described in Eq. (1) and Algorithm 1, the pairwise similarity measurement $\mathcal{F}(\mathbf{X}, \mathbf{Y})$ is the basis of the PSI algorithm. To see how different similarity measurements affect the system performance, we compare the adopted Spearman's $\hat{\rho}$ with the cosine similarity which is commonly used for measuring text similarity. We use both metrics to measure the training pair similarity of the MSCOCO dataset and the results are shown in Figure 4. As it can be seen that the distribution of cosine similarity is condensed in a much smaller interval comparing with the one from Spearman's $\hat{\rho}$, showing that the Spearman's $\hat{\rho}$ is more discriminative and can detect more subtle differences between the sentence pairs. Therefore, it can better identify the primary content, leading to better model performance. For further analysis, we run experiments on the MSCOCO dataset using cosine similarity as the measurement approach. The results of IANet+$\mathcal{S}$ using both metrics are shown in Table 5 which also demonstrate the fact that a more discriminative measurement approach leads to a better model performance.

6.6.3 Effect of α_p in PSI

As described in Algorithm 1, the proposed PSI algorithm relies on a predefined threshold α_p to perform the extraction of primary and secondary content. In this part, we examine the effect of different α_p on the model performance. We vary the value of α_p and measure the results of IANet+$\mathcal{S}$

	Input Sentence	Reference Sentence
	what are some of the best young adult fiction novels ?	which are the best young adult novels / films ?
	Input Sentence with Identified Components	**Generated Paraphrase**
IANet+ref	what are some of the best young adult fiction novels ?	which are the best young adult novels ?
IANet+$\mathcal{S}$	what are some of the best young adult fiction novels ?	which is a good young adult fiction novel you have read ?
IANet+**X**	what are some of the best young adult fiction novels ?	which are the best fiction novels ever written ?
Controlled Paraphrase Generation		
	what are some of the best young adult fiction novels ?	which are some good adult fiction novels ?
	what are some of the best young adult fiction novels ?	which is the best fiction novel ?
	what are some of the best young adult fiction novels ?	which are some good novels of all time ?

Table 6: **Paraphrase Generation Samples from Quora dataset**: To compare different inference algorithms, we present the results of different model variants for the sampled instance. To examine the generation controllability of the proposed model, we generate sentences by manually decomposing the input sentence. Specifically, we choose different parts of the input sentence as the primary content (highlighted in blue) and the rest as the secondary content. The results on the right side are the corresponding generated paraphrases.

model on the Quora dataset. The results of three metrics (B-1, R-1, and R-L) are depicted in Figure 5. We see that the optimal value of α_p is 0.1 and by further decreasing or increasing α_p, the model performance drops. Our analysis is that, when α_p is too small, the words that only cause small variation in the pairwise similarity (Eq. (1)) will be misclassified as primary. Therefore, extra noise might be introduced to the model input which in turns decreases the model performance. On the other hand, when α_p is too large, some important words that should be classified as primary content might be excluded by the PSI algorithm, which also leads to the decrease of model performance.

6.6.4 Case Study

As described in section §6.6.1, the reason why IANet+ref outperforms IANet+**X** and IANet+$\mathcal{S}$ is that it can more accurately identify the primary content in order to generate a paraphrase that is similar to the reference sentence. On the other hand, both IANet+**X** and IANet+$\mathcal{S}$ adopt an approximated algorithm which would inevitably introduce extra noise in the identified content.

For a better illustration, we sample one test case from Quora dataset and present the results generated by our different model variants in Table 6. We can see that, given the input sentence, all model variants can generate a sentence that is similar to the reference paraphrase. By further comparing the primary content (words in blue), we can see that only the IANet+ref successfully identifies all the primary content that are also contained in the reference sentence. On the other hand, IANet+**S** misses the word *best* and IANet+**X** ignores the words *young* and *adult*. As a result, IANet+ref can generate paraphrase that is closer to the reference sentence, leading to higher performances in different evaluation metrics as shown in Table 2.

6.6.5 Controllable Paraphrase Generation

Since the identification of the primary and secondary content of the input are separated from the neural generator, we therefore have the flexibility to manually choose these content. In this way, we can more precisely control the generation process.

To examine the controllability of the proposed approach, we manually select the primary and secondary content of the sampled instance and use the IANet+$\mathcal{S}$ model to generate paraphrases accordingly. The results based on different selections are presented in Table 6. As demonstrated by the examples, our model is flexible to generate different paraphrases given different combinations of the primary and secondary content. We can observe in the generated paraphrases that the selected primary content is largely maintained while the secondary content is properly rephrased.

This controllable attribute could make our model useful for other tasks such as task-oriented dialogue generation. Suppose we want to generate more utterances with the same meaning of the user utterance "*book a great* restaurant *in* London *tonight*". The slot values can be fixed as the primary content and our model could produce more utterances with the same intent, e.g. "*make a reservation at the best* London restaurant *for this evening*". This remains to be rigorously tested in future work.

7 Conclusion

In this work, we propose a novel IA framework to tackle the paraphrase generation task. Addition-

ally, we design a new neural paraphrase generator which works coherently under the proposed framework. We conduct extensive experiments on two benchmark datasets. The results of quantitative experiments and human evaluation demonstrate that our approach improves upon previous studies. The qualitative experiments show that the generation of the proposed model is interpretable and controllable. In the future, we would like to investigate a better inference algorithm to further bridge the gap between the IANet+$\mathcal{S}$ and IANet+ref models.

Acknowledgments

The authors wish to thank Jialu Xu, Deng Cai, and Sihui Wang for their insightful discussions and support. Many thanks to our anonymous reviewers for their suggestions and comments.

Ethical Statement

We honor and support the ACL Code of Ethics. Paraphrase generation aims at automatically generating paraphrased sentence given the source sentence. The generation process does not involve any bias towards the users. All datasets used in this work are from previously published works, and in our view, do not have any attached privacy or ethical issues.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the right questions: Active question reformulation with reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3152–3158.

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.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 875–886.

J.L. Fleiss et al. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Yao Fu, Yansong Feng, and John P. Cunningham. 2019. Paraphrase generation with latent bag of words. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13623–13634.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1875–1885.

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6010–6021. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

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.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*.

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*, pages 740–755.

Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Kathleen R. McKeown. 1979. Paraphrasing using given and new information in a question-answer system. In *17th Annual Meeting of the Association for Computational Linguistics, 29 June - 1 July 1979, University of California at San Diego, La Jolla, CA, USA*.

Marie Meteer and Varda Shaked. 1988. Strategies for effective paraphrasing. In *Proceedings of the 12th International Conference on Computational Linguistics, COLING '88, Budapest, Hungary, August 22-27, 1988*, pages 431–436.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

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.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. 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 6033–6039.

Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1235–1246.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual SMT system for paraphrase generation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 38–42.

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, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A task in a suit and a tie: Paraphrase generation with semantic augmentation. 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 7176–7183.

Vitalii Zhelezniak, April Shen, Daniel Busbridge, Aleksandar Savkov, and Nils Hammerla. 2019. Correlations between word vector sets. 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 77–87.