Writing Polishment with Simile: Task, Dataset and A Neural Approach

Jiayi Zhang, Zhi Cui, Xiaoqiang Xia, Yalong Guo,
Yanran Li, Chen Wei and Jianwei Cui
Xiaomi AI Lab, Beijing, China
{zhangjiayi3, cuizhi, liyanran, weichen, cuijianwei}@xiaomi.com

Abstract

A simile is a figure of speech that directly makes a comparison, showing similarities between two different things, e.g. "Reading papers can be dull sometimes, *like watching grass grow*". Human writers often *interpolate* appropriate similes into proper locations of the plain text to vivify their writings. However, none of existing work has explored neural simile interpolation, including both locating and generation. In this paper, we propose a new task of Writing Polishment with Simile (WPS) to investigate whether machines are able to polish texts with similes as we human do. Accordingly, we design a two-staged Locate&Gen model based on transformer architecture. Our model firstly *locates* where the simile interpolation should happen, and then *generates* a location-specific simile. We also release a large-scale Chinese Simile (CS) dataset containing 5 million similes with context. The experimental results demonstrate the feasibility of WPS task and shed light on the future research directions towards better automatic text polishment.

Introduction

Figurative language, or a figure of speech (修辞), is phrasing that goes beyond the literal meaning of words to get a message or point across. Writers and poets use figurative language to build imagery and elicit aesthetic experiences (Citron and Zervos 2018). In computational linguistics, figurative language processing (FLP) has long been an interesting research topic, including both detection (Li and Sporleder 2010; Klebanov et al. 2020) and generation tasks (Mishra, Tater, and Sankaranarayanan 2019; Yu and Wan 2019; Liu et al. 2019).

There exist a handful of figurative types that help make concepts become vivid and graspable, including but not limited to simile (明喻), metaphor (隐喻), irony, etc. Among them, similes play a vital role for human writings to be attractive. Different from metaphors' using implicit comparisons, a simile is a description that uses "like" or "as" to make a clear comparison between two separate concepts. As shown in Table 1, human writers add coherent similes into proper locations of the original text to vivify plain writings. Such an *interpolation*-based text polishing process is especially unique for similes, since most polishing objectives clearly requires text rephrasing, e.g., grammar error correction for fluency polishment, text editing for irony style transfer, etc. Distinctly, interpolating similes is like putting proper ingredients to an unflavored dish, instead of totally re-cooking a new one based on a different recipe. Despite the importance of simile, only a few work has explored simile recognition (Liu et al. 2018; Zeng et al. 2020). To the best of our knowledge[1], none of existing work has ever investigated simile generation given a plain text, which is indispensable for amplifying writing with similes.

Writing Polishment with Simile		
Before	Looking at his bloodthirsty eyes, everyone felt horrible and couldn't help but step back.	
After	Looking at his bloodthirsty eyes, everyone felt horrible **as if they were being stared at by a serpent**, and couldn't help but step back.	
Other Figurative Language Generation		
Task	Status	Text
Metaphor	Before	A metaphorical pair of (Enjoy, Devour)
	After	She devoured his novels.
Irony	Non-ironic	Tried to leave town and my phone died, what a failure.
	Ironic	**Nice** to leave town and my phone died, **definition of success**.

Table 1: Example of WPS and other related tasks.

Although sequence-to-sequence models work well for story generation (Liu et al. 2020), irony generation (Zhu, Yu, and Wan 2019), or metaphor and personification generation (Liu et al. 2019), it is non-trivial for these models to generate proper and creative simile for a given text. In particular, writing polishment with similes is a unique task because it requires to together address the challenges listed below:

- Locating is critical for simile interpolation. A simile inserted at a wrong place will impact language fluency and result in weird reading experience.
- To polish writing appropriately, the generated simile must be coherent to the context and diverse in the semantics

[1]We encourage readers to also refer to a contemporary work by Chakrabarty, Muresan, and Peng (2020), which shares a different point of view of simile generation.

and expressions.

- Since a simile is disentangled from its context, existing methods are hardly applicable (Gu, Wang, and Zhao 2019; Malmi et al. 2019; Su et al. 2019) since they do not target on interpolation-based text editing.
- Like most text style transfer or figurative language generation tasks (Rao and Tetreault 2018; Prabhumoye et al. 2018; Zhu, Yu, and Wan 2019; Yu and Wan 2019), there is no large corpus suitable for learning simile interpolation.

To this end, we propose a new task of Writing Polishment with Simile (WPS)—to firstly decide *where* to put a simile within plain input text, then figure out *what* content to generate as a coherent simile. To facilitate our research, we propose a new Chinese Simile (CS) dataset, which contains roughly 5.5 million similes in fictional contexts. We also set up a benchmark model Locate&Gen to validate the feasibility and potentials of WPS task. Locate&Gen model is a two-stage biased generation model upon the framework of transformer encoder-decoder (Vaswani et al. 2017). At the first step, it *locates* a pointer position for simile interpolation, and then *generates* a location-specific simile using a novel insertion bias. The two-stage design allows both automatic and semi-automatic inference modes to assist writing polishment flexibly. To summarize, our contributions are three-folded:

- We introduce the task of Writing Polishment with Simile (WPS), which we believe is a critical step towards figurative writing polishment.
- We develop[1] a large-scale Chinese Simile (CS) dataset for public research, which contains millions of similes with contexts extracted from Chinese online fictions.
- We establish benchmark model Locate&Gen and compare it with several SOTA models on the CS dataset, by which we analyze the task in detail.

Related Work

Figurative Language Generation As a figure of speech, simile generation is related to general figurative language generation. Yu and Wan (2019) first studied on end-to-end framework for metaphor generation, and Zheng et al. (2019) integrated template-based and CBOW-based metaphor generation into chatbots. Mishra, Tater, and Sankaranarayanan (2019) proposed multi-staged reinforced seq2seq and retrieval framework for irony generation, while Zhu, Yu, and Wan (2019) applied semi-supervised back translation approach with trained irony classifier. Chakrabarty et al. (2020) proposed retrieve-and-edit framework for sarcasm generation. All these works applied unsupervised learning due to data insufficiency, and none of them considered the scenario of text polishing as WPS. Despite that Liu et al. (2019) explored metaphor and personification over modern Chinese poetry generation, their model is only able to generate new lines instead of polishing existing ones. Similar shortcomings are also observed in most works on story generation (Tu et al. 2019; Fan, Lewis, and Dauphin 2019). Guan et al. (2020) recently devised a knowledge-enhanced model, but it is hard to generate figurative language without explicit goals. Especially for simile, Harmon (2015) developed a rule-based approach to form a simile given two comparable concepts, and a few work performed simile recognition more recently (Liu et al. 2018; Zeng et al. 2020). Our work differs from them in that we aim to polish text with simile interpolation, where context is important for the generated similes to be coherent.

Text Style Transfer Writing polishment is also related to text style transfer, where the core is the disentanglement of content and implicit attributes called "style". Wang, Hua, and Wan (2019) and Dathathri et al. (2020) explored plug-and-play schema for controllable text editing. Prabhumoye et al. (2018) and Yang et al. (2018) proposed back-translation and language discriminators respectively to deal with unsupervised text style transfer. Notably, three important differences distinguish our WPS task from previous ones: **1)** WPS strongly depends on contexts, i.e. the original plain text. **2)** Unlike the latent "style", a simile is clearly separable from the semantic content of the original writing. In other words, WPS task requires simile interpolation instead of sentence rephrasing. **3)** WPS is unique in the need of locating where to insert a simile. Hence, the challenges for WPS include positioning accuracy, contextual coherence and aestheticness of the generated simile.

Text Editing at Specific Location Past work designed sequence operations to refine a given piece of text for Grammar Error Correction and Contextual Query Rewrite (Su et al. 2019). Gu, Wang, and Zhao (2019) developed Levenshtein Transformer that supports deletion and insertion operations through iterative decoding schema. Su et al. (2019) applied copy mechanism for transformer encoder-decoder to accomplish utterance rewriting. These methods are designed to process the whole input sequence, which is not suitable for generating similes at specific positions. Malmi et al. (2019) developed LASERTAGGER that combines sequence tagging of keep, delete and swap with local sequence generation based on fixed common vocabulary. However, WPS involves only a single simile insertion place, and a common vocabulary would impact simile diversity. In this work, inspired by Li et al. (2020) who proposed a unified MRC framework for named entity recognition, we adopt a simple but effective approach called Locate&Gen to realize writing polishment with similes.

Chinese Simile Dataset

In this section, we present our large-scale Chinese Simile Dataset on its collection, processing as well as qualitative and quantitative analysis.

Data Collection Although Liu et al. (2018) has introduced a simile recognition dataset based on student essays, it only contains 5,088 simile samples extracted from student essays, which is insufficient for generating diverse and high-quality similes. In comparison, Chakrabarty, Muresan, and Peng

[1] https://github.com/mrzjy/writing-polishment-with-simile.git

Data Split			Avg. Length		# Uniq.
Train	Dev	Test	Context	Simile	Simile
5,485,721	2,500	2,500	52.7	8.3	$\sim$ 3M
Position Distribution			Complexity Distribution		
Start	Middle	End	Single	Comb.	Clause
4.2%	82.3%	13.5%	37%	29%	34%

Table 2: Statistics and properties of Chinese Simile Dataset.

Samples from CS dataset	
#1	他{像幽灵一样出现}那里，单手握门框上，挡住了那女的退路。耀眼如银丝般的长发下... He appeared there {like a ghost}, holding the door frame with one hand, blocking the her retreat...
#2	那些狼首怪物休想靠近她，一旦接近十米的范围，就{像是被流沙沼泽吞噬掉似的}陷入地下。 Those wolf-head monsters ... would suddenly sink into the ground {as if swallowed by a quicksand swamp} when getting close to her in 10 meters range.

Table 3: Samples from Chinese Simile Dataset. The extracted ground-truth similes within context are enclosed in braces. Translation is provided for non-Chinese speakers.

(2020) collected comments containing similes from Reddit and then auto-constructed a parallel simile corpus thanks to the commonsense knowledge inference by COMET (Bosselut et al. 2019), and finally obtained 80k+ self-labeled human written similes. However, their similes are all limited in a "like a" pattern and appear only at the end of a sentence, which is still different from real-world setting. Instead, we choose as our data sources the online free-access fictions that are tagged with sci-fi, urban novel, love story, youth, etc, because fiction writers often use figurative language to engage their audience. The raw corpus we accessed and crawled contains 22,463 online Chinese fictions for 60GB of text with roughly 470 million characters.

Data Processing We split long paragraphs into pieces of sentences and re-combine them to our contextualized simile samples, whose max length is set to 128. Thus, each sample in our CS dataset typically corresponds to several continuous sentences or a whole paragraph from fictions. We automatically extract sentences that match a dozen of rich Chinese simile start and ending patterns (e.g. "好像", "仿佛", "宛若", "俨然", "如同", "似的", "一般", etc., all meaning "as if" or "like"). Similes containing personage names are eliminated (e.g. "as beautiful as Elizabeth") by Jieba's name recognition[1]. For similes that occur more than 100 times, we downsample them by a keep ratio of 100/occurrence so as to mitigate the problem of generating frequent but meaningless similes. Detailed statistics could be found in Table 2.

Data Analysis We ask three annotators to assess the properties of 500 random samples from CS dataset on multiple aspects, and present the assessment in Table 2.

[1]https://github.com/fxsjy/jieba

- **Data Quality.** Although Liu et al. (2018) claimed that neural methods lead to higher recognition accuracy than rule-based ones, all their simile data is constructed using a single pattern, i.e., sentences containing the comparator "像". In contrast, we adopt more complex regex patterns and ensure an extraction precision of 92% approximately. Incorrect extractions are either due to over-recall on simile patterns or failures of Jieba's name recognition. Given the massive unique similes, we didn't pursue recalling more similes at the risk of hurting precision.
- **Variety of Simile Position.** We investigate whether the simile appears at the start, middle or end of the entire content. It turns out that 82.3% of similes are at somewhere middle of a text, demonstrating the strong feature of interpolation.
- **Generation Challenge.** To understand the generation difficulty of the task, We distinguish the simile complexity on three levels. Ranging from simple to hard, we define similes containing only a noun or a verb as single, "ADJ. + noun" or "ADV. + verb" as combined, and attributive clause as clause. The statistics show that a decent amount of similes in fictions include complex expressions, such as sample #2 shown in Table 3. Drawing on the overall analysis, we find that human writers interpolate similes at various positions in different shapes. This implies that for WPS task, simile diversity should be also considered during evaluation.

Model

Task Definition and Overview

The task of Writing Polishment with Simile (WPS) is formulated as follows: Given a sequence of context tokens $X = \{x_0, ..., x_N\}$, the goal is to insert a coherent simile $Y = \{y_1, ..., y_T\}$ at a proper position of X. In order to mimic human behaviors, we cast this task as a two-staged process: The first stage is simile positioning, which predicts where to insert a simile within X. The insertion position is denoted as i_{Ins}, where $0 \leq i_{\text{Ins}} \leq N$. The second stage is simile generation, which generates a simile Y word-by-word considering the context X and the predicted i_{Ins}. Hence, the probability of generating a simile Y can be depicted as:

$$P_Y = \prod_{i=1}^{T} P(y_t | y_{0:t-1}, i_{\text{Ins}}, X) \tag{1}$$

Locate&Gen Architecture

Encoder As shown in Figure 1, we design Locate&Gen framework. Typically, we apply BERT (Devlin et al. 2019) as our encoder. It takes X as input and encode it into hidden vector sequence $\mathbf{h} = \{h_0, h_1, ..., h_N\}$ based on multi-head self-attention, where $h \in \mathbb{R}^H$ and H is the hidden size. This process is depicted as follows:

$$\begin{aligned} \mathbf{e} &= \{\text{Embed}(x_0), ..., \text{Embed}(x_N)\} \\ \mathbf{h} &= \text{BERT}(\mathbf{e}) \end{aligned} \tag{2}$$

where $Embed$ is the embedding function (e.g., the sum of word, segment and position embedding) that transforms a

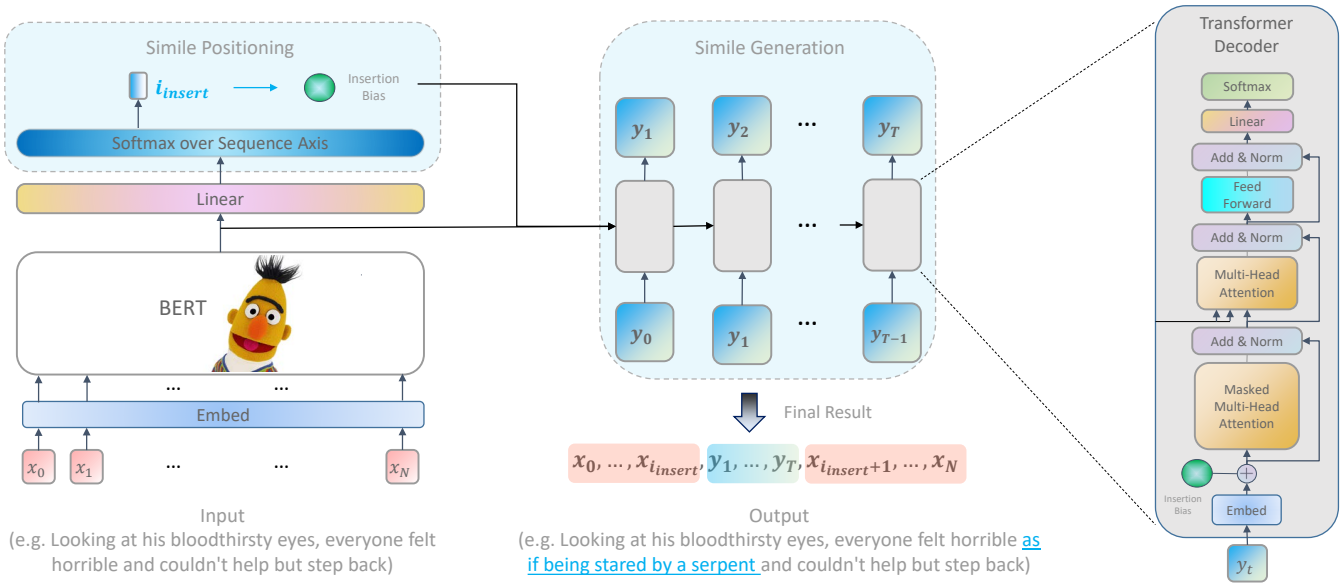

Figure 1: Illustration of Locate&Gen model that is derived from transformer encoder-decoder framework. The first stage of simile positioning is shown at the upper left, the generation stage is at the right. An insertion vector is obtained based on positioning result that bias the decoder towards desired sequence generation.

token x_i into a vector $e_i \in \mathbb{R}^E$. Note that dimension E is set to be equal to hidden size H for standard transformer model.

Simile Positioning To compute the probability of a position $i \in [0, N]$ being the insertion position pointer i_{Ins}, we apply softmax over all of projected hidden vectors along sequence axis:

$$P_i^{\text{Ins}} = \frac{\exp(h_i W_{\text{Ins}})}{\sum_j^N \exp(h_j W_{\text{Ins}})}, i \in [0, N] \tag{3}$$

where $W_{\text{Ins}} \in \mathbb{R}^{H \times 1}$ is a pointer weight matrix, and h_i is the i-th hidden vector of encoder outputs $\mathbf{h}$ according to equation 2. Note that a special token "[CLS]" (denoted as x_0) is prepended to each input sequence during BERT pretraining phase, which could also be used as a null pointer for cases when no possible insertion location exists. The insertion position i_{Ins} is simply calculated as follows:

$$i_{\text{Ins}} = \underset{i}{\text{argmax}}(\{P_i^{\text{Ins}} | i \in [0, N]\}) \tag{4}$$

Simile Generation The decoder is adapted from standard transformer decoder. The decoding process involves not only the self attention among previously decoded tokens (a.k.a. causal attention), but also the encoder-to-decoder attention as well. It takes the encoder output $\mathbf{h}$ and the previously decoded hidden vectors $\{s_0, ..., s_{t-1}\}$ as input, and autoregressively produce one token y_t at each time step.

Insertion Bias In order to guide the decoder generation with the signal of simile positioning, we compute an insertion bias vector $k \in R^H$ by projecting the i_{Ins}-th encoder hidden vector as follows:

$$k = h_{i_{\text{Ins}}} W_{\text{IB}} \tag{5}$$

where $W_{\text{IB}} \in R^{H \times H}$ is a weight matrix. By considering the insertion bias, the probability P_{y_t} of generating y_t at each step is finally modeled as:

$$\begin{aligned} e_{t-1} &= \text{Embed}(y_{t-1}) + k \\ s_t &= \text{TransformerBlock}(\mathbf{h}, s_{0:t-1}, e_{t-1}) \\ P_{y_t} &= \text{softmax}(s_t W_e) \end{aligned} \tag{6}$$

where y_0 is a special start token, $W_e \in R^{E \times |V|}$ is the word embedding matrix with vocabulary size of $|V|$. The final selection of token y_t could be processed based on auxiliary decoding strategies such as greedy or sampling methods.

Training and Inference

Loss Since equation 4 is not differentiable, we treat the training of positioning and generation as multi-task learning, and minimize the cross-entropy loss respectively for the optimization of the two-staged Locate&Gen model. The loss is calculated as:

$$\mathcal{L}_{\text{Total}} = \underbrace{(-\sum_{i=0}^{N} i_{\text{Ins}} \log P_i^{\text{Ins}})}_{\text{Positioning Loss}} + \underbrace{(-\sum_{t=0}^{T} y_t \log P_{\hat{y}_t})}_{\text{Generation Loss}} \tag{7}$$

where i_{Ins} and y_t here denote the ground-truth insertion position and gold simile token at t-th time step, respectively.

Teacher Forcing Similar to previous approaches, there also exists an exposure bias in Locate&Gen framework since during training, we expose ground-truth target, i.e., the real

pointer position and gold previous sequence tokens. However, we only observed limited improvement by using techniques such as scheduled sampling (Bengio et al. 2015) or Gumbel Softmax Approximations (Jang, Gu, and Poole 2017). The reasons why exposure bias is not that harmful in our case might be several-folds. For one cause, simile is relatively easy to locate and the simile length is often shorter than normal text generation tasks. Also, the strong capacity of the BERT encoder also results in robust performance (Hsieh et al. 2019), which help alleviate the issue of exposure bias (Kasai et al. 2020). We leave the usage of more complex training techniques to future works.

Inference Since WPS is interpolation-based, two inference modes can be adopted. **1) Automatic Mode.** Ideally, our Locate&Gen model is able to complete the simile positioning and generation given a plain text in a fully automatic way, without the need of explicitly telling the model where to add similes. **2) Semi-automatic Mode.** Meanwhile, thanks to the design of insertion bias, it's also possible to put simile at any arbitrary location other than the most probable one. In such cases, the model takes an additional input of i_{Ins} so as to directly start the biased generation. We investigate both of these two modes in the experiments.

Experiments

In this section, we benchmark and analyze the performances of baseline models including Locate&Gen as well as retrieval approaches.

Baseline Models

Locate&Gen We set up Locate&Gen model as introduced in the previous section. To investigate the significance of each component, we also compare with several variants: Locate&Gen (**+ beam n**) is an enhanced version where we apply beam search decoding with beam size of n. Locate&Gen (**- w/o Pretrain**) is an ablated model, in which the BERT encoder is randomly initialized and trained from scratch. Locate&Gen (**- w/o InsBias**) is another ablated version where we eliminate the insertion bias from the decoder and retrain the model.

Retrieve&Match It also makes sense to select proper simile(s) using retrieve-then-rerank techniques. **1) Retrieve.** Our assumption is that a simile coherent to a given context is also likely to be suitable for similar contexts. In the first step, we treat as context the 16 surrounding characters around the insertion position and retrieve 100 most similar contexts from training corpus using **BM25** ranking scores (Robertson and Zaragoza 2009). Note that setting the retrieval context length to 16 is an empirical choice, since either shorter or longer context ends in inferior retrieval performance. If no further step exists, the simile with the highest BM25 score is chosen as the model output. **2) Match.** To improve the retrieval performance, we also collect the top 100 similes from the first step as candidates, and acquire two matching models to re-rank them. **MatchCBOW** calculates cosine similarity of sentence embeddings (i.e., average of word embeddings) between simile and the original writing. We also adopt **MatchBERT**, a BERT matching model whose input is the concatenation of context and simile. The simile with the highest matching score is returned as the final output. For fair comparison, all retrieval-based methods share the position predictions from Locate&Gen model.

Evaluation Metrics

Positioning Accuracy We evaluate the simile insertion accuracy compared with ground-truth simile position. The prediction is scored as correct only when it exactly matches the ground-truth location. Note that insertions located elsewhere do not necessarily lead to a bad simile interpolation.

Word Overlap (BLEU) BLEU[1] (Papineni et al. 2002) reflects the word overlap between the generated and the ground-truth text. Typically, we use BLEU_n ($n = 1, 2, 3$) considering the n-gram overlaps of similes.

Embedding Similarity To assess similes' contextual relevance, we use pretrained embeddings[2] to calculate embedding average (EA), greedy match (GM) and vector extrema (VE) (Liu et al. 2016) between simile and context.

Perplexity, Diversity, Length Perplexity (PPL.) measures how well a language model estimates the probability distribution over entire sentences (Adiwardana et al. 2020) (lower is better). Meanwhile, diversity computes the ratios of distinct unigrams, bigrams and sentences, denoted as Dist-1, Dist-2 and Dist-S, respectively. Besides, we adopt the generation length as a simple indicator of content richness.

Human Evaluation We conduct human evaluation in order to assess the overall performance of WPS, namely the simile positioning as well as generation. We adopt in total 4 aspects for human judgments. Fluency (Flu.) and creativity (Crea.) (Yu and Wan 2019) are to examine grammatical correctness of a simile given the surrounding context, and whether it's interesting and creative instead of dull or bland, respectively. In addition to Crea., we adopt informativeness (Info.) to investigate whether a simile is rich in content. Note that a creative simile does not need to be really rich in content, meanwhile a content-rich simile might still be universal or dull. Besides, we propose coherence (Coh.) to assess whether a simile coherently fits its context on semantic aspect. We ask 5 well-educated annotators to assess 100 random samples from test set and rate each simile with $\{0, 1\}$ score on the 4 aspects.

Implementation Details

All models are implemented in Tensorflow[3] and trained on Nvidia Tesla V100 GPU. We apply standard BERT-

[1] We use the public multi-bleu.perl script for BLEU calculation.
[2] https://ai.tencent.com/ailab/nlp/en/embedding.html
[3] https://www.tensorflow.org

Model	Positioning	Simile Generation										
	Accuracy	PPL	Word Overlap (With G.T.)			Len.	Contextual Similarity			Diversity		
			BLEU1	BLEU2	BLEU3		EA	GM	VE	Dist-1	Dist-2	Dist-S
BM25			20.00	1.37	0.25	8.6	0.777	0.177	0.540	0.088	0.472	**0.996**
MatchCBOW			25.13	2.96	0.56	**14.4**	**0.889**	0.168	**0.608**	0.060	0.402	0.988
MatchBERT			25.55	4.45	1.14	9.6	0.800	0.179	0.561	0.088	**0.494**	0.990
Locate&Gen			38.96	17.96	7.02	8.2	0.796	0.187	0.564	0.057	0.239	0.878
+ beam 2	**0.769**	**6.469**	39.73	18.82	7.68	7.9	0.791	0.188	0.560	0.059	0.247	0.887
+ beam 20			**41.28**	**19.77**	**8.32**	7.2	0.777	**0.190**	0.553	0.064	0.270	0.855
- w/o Pretrain	0.713	7.344	36.36	16.25	5.75	8.0	0.792	0.185	0.559	0.051	0.207	0.818
- w/o InsBias	0.726	6.870	36.98	16.42	5.85	8.1	0.791	0.187	0.559	0.053	0.217	0.831
G.T.						8.3	0.782	0.183	0.552	**0.090**	0.473	**0.996**

Table 4: Evaluation on automatic metrics of baselines on Simile Polishment. Highest scores are in bold. Metric scores of ground-truth (G.T. for short) similes are shown as well. Note that no positioning accuracy is reported for retrieval methods since they share the same simile position predictions from Locate&Gen.

Model	Flu.	Crea.	Coh.	Info.
BM25	0.89	0.55	0.54	0.64
MatchCBOW	0.80	0.56	0.43	**0.83**
MatchBERT	0.96	0.59	0.64	0.79
Locate&Gen	0.95	0.54	0.77	0.58
GroundTruth	**0.99**	**0.66**	**0.85**	0.61

Table 5: Human evaluation results. The human agreement is validated by Kappa coefficient (Fleiss 1971) of 0.45, indicating a "moderate agreement" among annotators.

base[4] settings as the encoder. For the decoder, we set as 2-layer transformer of 768 hidden size. We use standard BERT tokenizer to perform char-level Chinese tokenization with a vocab size of 21,128. During training, we apply embedding weight-tying as well as label smoothing technique (Müller, Kornblith, and Hinton 2019) to improve training speed and robustness. We use dropout of 0.1, and Adam optimizer (Kingma and Ba 2015) with a mini-batch of 128. The max context and target length are set to 128 and 16 respectively. For MatchBERT, we train with random negative sampling of size 5. Without hyper-parameter tuning, we set the learning rate to 5e-5 and train for maximum of 15 epochs with early stopping on Dev set.

Results and Analysis

Overall Analysis We first examine the results based on automatic metrics. At a glimpse of Table 4, the best scores for positioning and simile generation are promising but imperfect, suggesting the feasibility and potential of our WPS task. When we examine the contextual similarity and diversity scores (the last six columns), things become interesting. The similarity scores for ground-truth similes are comparable and even lower than the best scores achieved by benchmark models. It indicates that contextual similarity is a necessary but not sufficient measurement for WPS task. Moreover, the three similarity metrics favour different models due to different calculation manners. Both EA and VE compute sentence-level embeddings and thus models outputting long similes will gain higher EA and VE scores. It is because long similes often share more similar words with context. GM instead prefers similes resembling the key words in the contexts (Liu et al. 2016). Thus, GM is a more reasonable indicator of contextual similarity especially for noisy contexts. In terms of diversity, there is a clear gap between generated and human-written similes. Overall speaking, similes produced by retrieval-based methods are often more diverse but less coherent than generation-based ones, as implied by their higher Dist-n but poorer BLEU and GM scores.

Human Assessment In Table 5, we observe the human judgments are consistent with automatic evaluations that generative methods have advantages over retrieval ones in terms of coherence, which we assume is a critical and indispensable goal of WPS task. Surprisingly, the similes selected by BM25 and MatchCBOW are annotated with lower Flu. scores, even though they are human-written sentences by nature. We conjecture that these methods are prone to focus on context noise during context-simile semantic matching, hence the selected similes even break the overall language fluency when considering the context together. As such, MatchBERT is more capable of distinguishing the features and noise, which yields better performance. Note that even ground-truth simile is not always perfect, and its creativity score of 0.66 and informativeness of 0.61 partially reveal the need of automatic polishing to help humans with their writings.

Ablation Study

We investigate the impact of beam search (beam 2 and beam 20), BERT pretraining (versus w/o Pretrain) as well as insertion bias (versus w/o InsBias) on WPS task.

Beam Search Changing from greedy decoding to beam search results in gains on BLEU scores but losses on length and Dist-S. It is a common trade-off between blandness and diversity to when decoding with beam search.

BERT Pretraining Large pretrained language models (Radford et al. 2019; Devlin et al. 2019) have achieved impressive progress on NLP tasks (Wang et al. 2018; Brown

[4]We adopt pretrained Chinese BERT-wwm-ext checkpoint from https://github.com/ymcui/Chinese-BERT-wwm.

Case	Model	Simile Generation	Original Writing
#1	GroundTruth	like an upside-down lid	He was standing right beside a giant irregularly round cavity **[insert]**, only an underground river passes through its center. He just escaped from there.
	BM25	like a huge cave	
	MatchCBOW	like a huge cave	
	MatchBERT	as if in a karst cave	
	Locate&Gen	like a huge funnel	
#2	GroundTruth	like being frozen	And Zi Jiang's movements stopped abruptly **[Insert]**. She just paused and did not react any more.
	BM25	like a wooden puppet	
	MatchCBOW	like a person who has his neck severed	
	MatchBERT	as if someone pressed her stop button	
	Locate&Gen	like being casted a freezing charm	
#3	GroundTruth	like frightened quails	The few Titans who were knocked in the air finally came to their senses, and turned into vigilant state. However they suddenly prostrated **[Insert]** after seeing their king glaring at them.
	BM25	like a mess of mud	
	MatchCBOW	like being knocked down	
	MatchBERT	like worshiping god	
	Locate&Gen	like mice confronting a cat	
#4	GroundTruth	like a wronged little daughter-in-law	Leyao Su remained silent and looked pitiful **[Insert]**, but she could not actually talk about it.
	BM25	like a big dog about to be taken away	
	MatchCBOW	like a wronged little daughter-in-law	
	MatchBERT	like a puppy afraid of being abandoned	
	Locate&Gen	like an abandoned puppy	

Table 6: Model comparison of simile generation. For fair comparison, we choose cases where model prediction of simile position exactly matches ground-truth position (noted as **[insert]**). Note that the original texts are in Chinese, we however only show English translations due to space limit.

Automatic / Semi-Automatic Polishment
梁师这一脚好生了得，王宇感觉{像是被火车撞了一样}难受，身体就要不顾自己的控制[像断线的风筝一样]飞出悬崖，还好他及时用双手抓在水泥地上，只[如同鹰爪一样]抓划着地面发出了令人发指的声音。 Wang Yu was kicked so hard by Liang's feet and felt uncomfortable {like being hit by a train}, he almost seemed to be kicked off the cliff [like a broken kite]. Fortunately, he grabbed the concrete floor [as eagle claws] in time, with ground-scratching sound.

Table 7: Polishing modes. Generated similes in automatic and semi-automatic mode are enclosed in braces and brackets, respectively. We apply beam search decoding for reproducibility.

et al. 2020). From Table 4, we can see that BERT pretraining is also of great contribution for WPS. It suggests that the results might be further improved if BERT is pretrained on large-scale fiction corpora, which we leave as future work.

Insertion Bias We observe that insertion bias also counts a lot. Injecting the bias to the decoder grants Locate&Gen the control over insertion on arbitrary positions. Generation models without it would neglect the predicted insertion position and could not help producing identical similes for a given context. The improvement yielded by the bias show that in addition to the encoder-decoder attention mechanism, equipping the decoder with interpolation signals is also beneficial for generating coherent similes.

Case Study

As shown in Table 6, we perform case study to delve into the advantages and shortcomings of our Locate&Gen model.

Making Meaningful Comparisons For case #1, all retrieval methods select similes containing the word "cave", resulting in similes as "cavity is like a cave", which fail to elicit aesthetic experiences. In comparison, both Locate&Gen and ground-truth produce engaging similes that compare a cave with funnel and lid, respectively.

Approaching Ground-Truth All models perform well in case #2 whereas the simile by MatchCBOW is a bit too specific given the current non-informative context. Note that Locate&Gen yields the best similarity with ground-truth since they both apply the word "freeze" in similes to describe "movements stopped abruptly".

Enhancing Drama Case #3 is even more interesting. Both ground-truth and Locate&Gen generate not simply a simile, but an oxymoron (反衬) at the same time, which is another figure of speech where contradictory terms appear in conjunction. Specifically, ground-truth and Locate&Gen use words like "quail" or "mouse" to describe "Titans", respectively. Moreover, "mouse confronting a cat" generated by Locate&Gen is dramatic and coherent with the context "Titans seeing their king".

Lacking Content Richness For case #4 however, we observe that retrieval methods especially MatchBERT outper-

form Locate&Gen model in terms of generation diversity and richness. As a matter of fact, the overall performances across Table 4 and Table 6 suggest the potentials of ensembling retrieval and generation methods.

Inference Mode

As introduced before, there are two inference modes available for WPS task. The studies on them are shown in Table 7. In automatic mode (boxed in yellow), the model firstly predicts the insertion position, and then generates a coherent simile. However, there are inevitably cases where the model's own simile interpolation is not as satisfying as expected, such as the simile "like being hit by a train", which seems a bit weird. In this case, we could thus change to semi-automatic mode (boxed in orange) where the model takes an extra input of desired insertion position and directly starts the biased simile generation. The design of insertion bias grants us the ability to control the simile interpolation at any desired position with contextual coherence.

Shortcomings

In spite of the interesting results, Locate&Gen model still possesses several drawbacks.

Lack of Explainability on which target (a.k.a tenor (本体)) in the context is the predicted simile to describe. Take #2 in Table 6 as example. The simile "like being frozen" is inserted after a comma, but it means to describe the target Zi Jiang's movement. Attention visualization might be indicative to some point, but it's not always feasible for interpretability (Serrano and Smith 2019). Since the insertion position and attention weights are not always aligned with the tenor, it remains an open question on how to explain the simile predicted by models to better assist human writings. One desirable way might be to first identify a target tenor as an additional input feature for Locate&Gen model before performing simile interpolation.

Lack of Diversity and aestheticness of the generated simile, which is partly determined by the language complexity of training examples. As introduced before, we've distinguished the simile complexity with three levels, i.e., single, combined and clause as shown in Table 2. Intuitively, similes with distinct levels of complexity accord with different styles of contents. Hence, another promising way to improve simile quality is to feed extra features of complexity and styles into the model. We believe it is a critical research problem to assist humans with simile polishment.

To Polish or Not To Polish In real-world application, there are definitely cases where an input writing is already engaging enough and thus one does not need to gild the lily by adding more similes. As mentioned in the model section, our Locate&Gen framework is actually compatible with such a scenario since there is a null pointer location that could be used to indicate unnecessary insertions, as long as proper negative training samples exist for the model to learn. Such case is however beyond the scope of this work and we plan to explore it in the future.

Conclusion and Future Work

In this paper, we introduce a new task, Writing Polishment with Similes, and curate a large-scale Chinese simile dataset. Our experiments demonstrate the feasibility and potential of the task, which we consider as a first step towards figurative writing polishment in a real-world setting. We establish Locate&Gen model and benchmark it on the developed dataset.

Future works include but not limited to:

- Dataset refinement (e.g., neural simile recognition).
- Better model designs (e.g., retrieval-generation ensemble model has potentials, and tenor extraction as an additional feature).
- Focus on writing polishment tasks for other figurative types.

Furthermore, from an AI writing assistant perspective, we surmise that assisting humans with writing polishment is more likely to develop the potentials of current AI models than just letting AIs write on the fly (which has however become a typical slogan for cutting-edge generative models such as GPT3). Given that figurative language is an essential creative aspect of language use, we encourage the use of the CS dataset in various contexts and look forward to the emergence of intelligent writing assistant tools **like magic**[1] in the future.

References

Adiwardana, D.; Luong, M.; So, D. R.; Hall, J.; Fiedel, N.; Thoppilan, R.; Yang, Z.; Kulshreshtha, A.; Nemade, G.; Lu, Y.; and Le, Q. V. 2020. Towards a Human-like Open-Domain Chatbot. *CoRR* abs/2001.09977.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *NIPS*, 1171–1179.

Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Çelikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *ACL*, 4762–4779.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165.

Chakrabarty, T.; Ghosh, D.; Muresan, S.; and Peng, N. 2020. R^3: Reverse, Retrieve, and Rank for Sarcasm Generation with Commonsense Knowledge. In *ACL*, 7976–7986.

[1]We applied our Locate&Gen model to generate this simile, which is "如同魔术般的"(智能写作助手) in Chinese before being translated to English.

Chakrabarty, T.; Muresan, S.; and Peng, N. 2020. Generating similes effortlessly like a Pro: A Style Transfer Approach for Simile Generation. In *EMNLP*, 6455–6469.

Citron, F.; and Zervos, E. A. 2018. A neuroimaging investigation into figurative language and aesthetic perception. In *Sensory Perceptions in Language, Embodiment and Epistemology*. Springer.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *ICLR*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 4171–4186.

Fan, A.; Lewis, M.; and Dauphin, Y. N. 2019. Strategies for Structuring Story Generation. In *ACL*, 2650–2660.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5).

Gu, J.; Wang, C.; and Zhao, J. 2019. Levenshtein Transformer. In *NeurIPS*, 11179–11189.

Guan, J.; Huang, F.; Huang, M.; Zhao, Z.; and Zhu, X. 2020. A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation. *Trans. Assoc. Comput. Linguistics* 8: 93–108.

Harmon, S. 2015. FIGURE8: A Novel System for Generating and Evaluating Figurative Language. In *ICCC*, 71–77.

Hsieh, Y.; Cheng, M.; Juan, D.; Wei, W.; Hsu, W.; and Hsieh, C. 2019. On the Robustness of Self-Attentive Models. In *ACL*, 1520–1529.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR (Poster)*.

Kasai, J.; Pappas, N.; Peng, H.; Cross, J.; and Smith, N. A. 2020. Deep Encoder, Shallow Decoder: Reevaluating the Speed-Quality Tradeoff in Machine Translation. *CoRR* abs/2006.10369.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Klebanov, B. B.; Shutova, E.; Lichtenstein, P.; Muresan, S.; Leong, C. W.; Feldman, A.; and Ghosh, D., eds. 2020. *Proceedings of the Second Workshop on Figurative Language Processing, Fig-Lang@ACL 2020, Online, July 9, 2020*.

Li, L.; and Sporleder, C. 2010. Using Gaussian Mixture Models to Detect Figurative Language in Context. In *HLT-NAACL*, 297–300.

Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; and Li, J. 2020. A Unified MRC Framework for Named Entity Recognition. In *ACL*, 5849–5859.

Liu, C.; Lowe, R.; Serban, I.; Noseworthy, M.; Charlin, L.; and Pineau, J. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *EMNLP*, 2122–2132.

Liu, D.; Li, J.; Yu, M.; Huang, Z.; Liu, G.; Zhao, D.; and Yan, R. 2020. A Character-Centric Neural Model for Automated Story Generation. In *AAAI*, 1725–1732.

Liu, L.; Hu, X.; Song, W.; Fu, R.; Liu, T.; and Hu, G. 2018. Neural Multitask Learning for Simile Recognition. In *EMNLP*, 1543–1553.

Liu, Z.; Fu, Z.; Cao, J.; de Melo, G.; Tam, Y.; Niu, C.; and Zhou, J. 2019. Rhetorically Controlled Encoder-Decoder for Modern Chinese Poetry Generation. In *ACL*, 1992–2001.

Malmi, E.; Krause, S.; Rothe, S.; Mirylenka, D.; and Severyn, A. 2019. Encode, Tag, Realize: High-Precision Text Editing. In *EMNLP/IJCNLP (1)*, 5053–5064.

Mishra, A.; Tater, T.; and Sankaranarayanan, K. 2019. A Modular Architecture for Unsupervised Sarcasm Generation. In *EMNLP/IJCNLP*, 6143–6153.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 4696–4705.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

Prabhumoye, S.; Tsvetkov, Y.; Salakhutdinov, R.; and Black, A. W. 2018. Style Transfer Through Back-Translation. In *ACL*, 866–876.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog (Accessed on 2019-05-16)* 1(8).

Rao, S.; and Tetreault, J. R. 2018. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *NAACL-HLT*, 129–140.

Robertson, S.; and Zaragoza, H. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Serrano, S.; and Smith, N. A. 2019. Is Attention Interpretable? In *ACL*, 2931–2951.

Su, H.; Shen, X.; Zhang, R.; Sun, F.; Hu, P.; Niu, C.; and Zhou, J. 2019. Improving Multi-turn Dialogue Modelling with Utterance ReWriter. In *ACL*, 22–31.

Tu, L.; Ding, X.; Yu, D.; and Gimpel, K. 2019. Generating Diverse Story Continuations with Controllable Semantics. In *NGT@EMNLP-IJCNLP*, 44–58.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *BlackboxNLP@EMNLP*, 353–355.

Wang, K.; Hua, H.; and Wan, X. 2019. Controllable Unsupervised Text Attribute Transfer via Editing Entangled Latent Representation. In *NeurIPS*, 11034–11044.

Yang, Z.; Hu, Z.; Dyer, C.; Xing, E. P.; and Berg-Kirkpatrick, T. 2018. Unsupervised Text Style Transfer using Language Models as Discriminators. In *NeurIPS*, 7298–7309.

Yu, Z.; and Wan, X. 2019. How to Avoid Sentences Spelling Boring? Towards a Neural Approach to Unsupervised Metaphor Generation. In *NAACL-HLT*, 861–871.

Zeng, J.; Song, L.; Su, J.; Xie, J.; Song, W.; and Luo, J. 2020. Neural Simile Recognition with Cyclic Multitask Learning and Local Attention. In *AAAI*, 9515–9522.

Zheng, D.; Song, R.; Hu, T.; Fu, H.; and Zhou, J. 2019. "Love Is as Complex as Math": Metaphor Generation System for Social Chatbot. In *CLSW*, volume 11831, 337–347.

Zhu, M.; Yu, Z.; and Wan, X. 2019. A Neural Approach to Irony Generation. *CoRR* abs/1909.06200.