Probing Simile Knowledge from Pre-trained Language Models

Weijie Chen[1*], Yongzhu Chang[2*], Rongsheng Zhang[2*], Jiashu Pu[2]
Guandan Chen[2], Le Zhang[2], Yadong Xi[2], Yijiang Chen[1] and Chang Su[1†]
[1] School of Informatics, Xiamen University, Xiamen, China
[2] Fuxi AI Lab, NetEase Inc., Hangzhou, China
chenwj@stu.xmu.edu.cn, {changyongzhu,zhangrongsheng}@corp.netease.com

Abstract

Simile interpretation (SI) and simile generation (SG) are challenging tasks for NLP because models require adequate world knowledge to produce predictions. Previous works have employed many hand-crafted resources to bring knowledge-related into models, which is time-consuming and labor-intensive. In recent years, pre-trained language models (PLMs) based approaches have become the defacto standard in NLP since they learn generic knowledge from a large corpus. The knowledge embedded in PLMs may be useful for SI and SG tasks. Nevertheless, there are few works to explore it. In this paper, we probe simile knowledge from PLMs to solve the SI and SG tasks in the unified framework of simile triple completion for the first time. The backbone of our framework is to construct masked sentences with manual patterns and then predict the candidate words in the masked position. In this framework, we adopt a secondary training process (Adjective-Noun mask Training) with the masked language model (MLM) loss to enhance the prediction diversity of candidate words in the masked position. Moreover, pattern ensemble (PE) and pattern search (PS) are applied to improve the quality of predicted words. Finally, automatic and human evaluations demonstrate the effectiveness of our framework in both SI and SG tasks.

1 Introduction

The simile, which is a special type of metaphor, is defined as a figurative expression in which two fundamentally different things are explicitly compared, usually using "like" or "as" (Israel et al., 2004; Zeng et al., 2020). It is widely used in literature because it can inspire the reader's imagination (Paul, 1970) by giving a vivid and unexpected analogy between two objects with similar attributes.

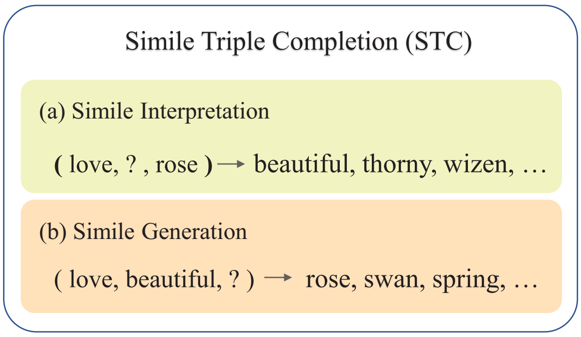

Simile Triple Completion (STC)
(a) Simile Interpretation
(love, ? , rose) → beautiful, thorny, wizen, ...
(b) Simile Generation
(love, beautiful, ?) → rose, swan, spring, ...

Figure 1: In the form of triple, the tasks of Simile Interpretation and Simile Generation can be unified into Simile Triple Completion.

A simile sentence usually contains three key elements: the **tenor**, the **attribute** and the **vehicle**,[1] which can be defined in the form of a triple (*tenor*, *attribute*, *vehicle*) (Song et al., 2021). For example, the simile sentence "Love is as thorny as rose" can be extracted as the triple (love, thorny, rose), where the tenor is "love", the vehicle is "rose", and the attribute is "thorny". Note that a simile triple can produce different simile sentences with different templates. For the example triple above, the simile sentences can be also constructed as "*love is thorny like rose*" with the pattern "*tenor* is *attribute* like *vehicle*".

The study of simile is benefit to many downstream tasks, like sentiment analysis (Rentoumi et al., 2012), question answering (Zheng et al., 2020), writing polishment (Zhang et al., 2021) and creative writing (Gero and Chilton, 2019). **Simile interpretation** (SI) (Qadir et al., 2016; Su et al., 2016) and **simile generation** (SG) (Yu and Wan, 2019) are the two important tasks in the study of simile (Tong et al., 2021). The SI task is to find suitable attributes as a mediator between the tenor

[*] Equal contribution. Work is done during Weijie's internship at NetEase Inc..

[†] Corresponding author. E-mail: suchang@xmu.edu.cn.

[1] Tenor: the logical subject of the comparison, usually a noun phrase. Attribute: what things being compared have in common, usually an adjective. Vehicle: the logical object of the comparison, usually a noun phrase.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5875 - 5887
May 22-27, 2022 ©2022 Association for Computational Linguistics

and vehicle. Likewise, the SG task is to select a proper vehicle for the tenor with the given attribution. And these two tasks can be unified into the form of **simile triple completion** (STC) (Song et al., 2021) as shown in Figure 1.

Previous works on the SI and SG tasks relied on a limited training corpus or labor-intensive knowledge base, which leads to an upper limit on the diversity of results. (Song et al., 2021) collected sentences containing comparator words from a Chinese essays corpus and manually annotated them to obtain the simile triple. Some works (Stowe et al., 2021; Gero and Chilton, 2019; Veale et al., 2016) relied on a knowledge base such as ConceptNet[2], FrameNet[3], which are scarce to other languages because it is time-consuming and labor-intensive to construct such a knowledge base. It is notable that pre-trained language models (PLMs) (Devlin et al., 2019; Radford et al., 2019) have made significant progress recently in many NLP tasks since it learns generic knowledge such as grammar, common sense from a large corpus (Davison et al., 2019; Liu et al., 2021a,b). Considering the sufficient existence of simile in the large corpus, it's reasonable to assume that PLMs are equipped with rich knowledge of similes during the pre-training stage. However, few works have explored directly probing the knowledge of simile from the PLMs.

In this paper, we propose a unified framework to solve the SI and SG tasks by mining the knowledge in PLMs, which does not require fine-labeled training data or knowledge graphs. The backbone of our method is to construct masked sentences with manual patterns from an incomplete simile triple, and then use language models with MLM heads to predict the masked words over the task-specific vocabulary. We take the K words with the highest probability as the result words. However, there are problems with this crude approach. Firstly, the predicted words should be creative and surprised for the simile sentence. On the contrary, the PLMs tend to predict common words (e.g., good, bad) with a higher probability. To address this issue, we introduce a secondary pre-training stage - Adjective-Noun mask Training (ANT), where only the noun or adjective contained in the *amod* dependencies (Nivre et al., 2017) could be masked in the MLM training process and the number of words masked times are limited. Secondly, the words predicted by MLM have a preference for different patterns. For this reason, we employ a pattern ensemble to obtain high-quality and robust results. Finally, we also introduce a prompt-search method to improve the quality of the simile component predictions.

Our main contributions are as follows:

- We propose a unified framework to solve both the simile interpretation (SI) and simile generation (SG) tasks based on pre-trained models. To the best of our knowledge, it is the first work to introduce pre-trained language models to unify these tasks.

- We propose a secondary pre-training stage that effectively improves the prediction diversity. Further, we employ the pattern-ensemble and pattern-search approaches to obtain better results.

- We compare our models on both automated metrics and manual measures, and the results show that our approach outperforms the baselines in terms of diversity and correctness.

2 Related Work

2.1 Simile Interpretation and Generation

Simile interpretation and simile generation are the two main directions of the simile study (Yu and Wan, 2019). The SI task (Shutova, 2010; Su et al., 2017) aims at finding a suitable attribute when given the tenor and vehicle, while the SG task (Yu and Wan, 2019) is to find a proper vehicle when given the tenor and its attribute. For simile interpretation, some works (Zheng et al., 2020; Bar et al., 2018; Xiao et al., 2016; Gagliano et al., 2016; Qadir et al., 2016) applied word vectors to decide which attribute words can fit into the tenor and vehicle domains and some other works (Gero and Chilton, 2019; Stowe et al., 2021) introduced knowledge base (Baker et al., 1998; Speer et al., 2017) to help find intermediate attributes. For simile generation, some works focused on constructing limited training corpus to finetune a sequence-to-sequence model (Lewis et al., 2020) by pattern-based (Zhang et al., 2021; Bollegala and Shutova, 2013) or knowledge-based approaches (Chakrabarty et al., 2020, 2021; Stowe et al., 2021). There are also some works (Abe et al., 2006; Hervás et al., 2007; Zheng et al., 2020) that focused more on the relationships between concepts (i.e.,

[2] https://conceptnet.io/

[3] https://framenet.icsi.berkeley.edu/fndrupal/

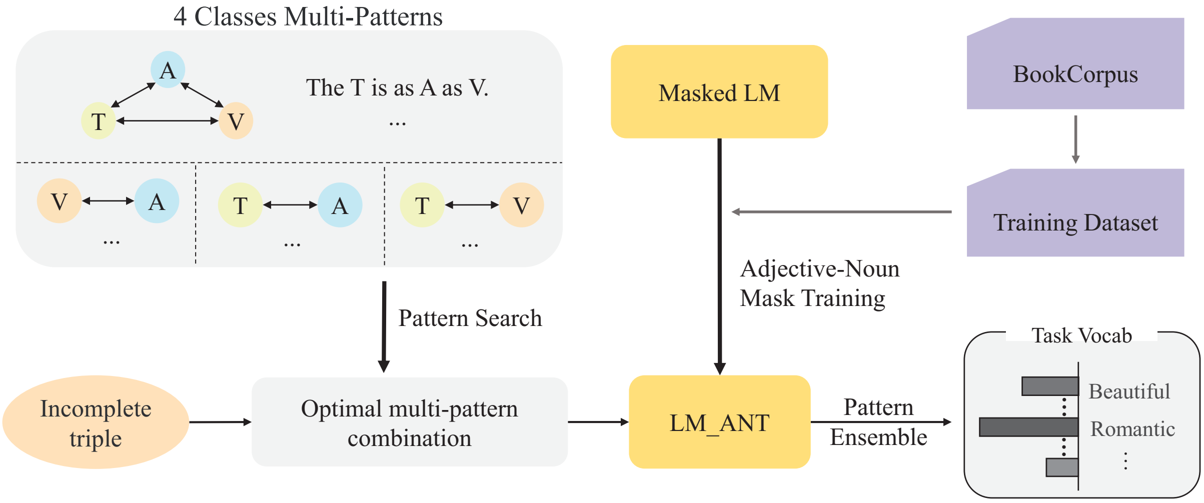


Figure 2: The unified framework for STC. The incomplete triple will be transfer to masked sentences by a multi-pattern combination, which is searched from four classes well-designed patterns. And LM_ANT (obtained by using unlabeled corpus to perform Adjective-Noun Mask Training) predicts the missing simile element over the task-specific vocabulary based on the masked sentences.

tenor and vehicle) and attribute. However, our paper carries out the task of simile interpretation and generation uniformly in the form of simile triples. And instead of extracting the simile triples from the limited corpus using designed templates or a hand-crafted knowledge base, we probe simile-related knowledge from PLMs.

2.2 Explore knowledge from PLMs

Pre-trained language models such as Bert and GPT (Devlin et al., 2019; Radford et al., 2019) are trained on the large-scale unlabeled corpus. Many recent works (Manning et al., 2020; Ettinger, 2020; Petroni et al., 2019; Shin et al., 2020; Haviv et al., 2021; Jiang et al., 2020; Zhong et al., 2021; Wang et al., 2022a,b; Li and Liang, 2021) focused on exploring the rich knowledge embedded in these PLMs. Manning et al. (2020) and Ettinger (2020) learned the syntactic and semantic knowledge from PLMs. Among these works, one branch of works(Petroni et al., 2019; Shin et al., 2020; Haviv et al., 2021; Jiang et al., 2020) designed discrete patterns to explore the common sense and world knowledge embedded in PLMs. In addition, some works (Zhong et al., 2021; Li and Liang, 2021) probed knowledge by searching the best-performing continuous patterns in the space of embedding. Inspired by the above works, in this paper, we probe the knowledge of simile in these pre-trained models and further apply pattern ensemble and pattern search to improve the results.

3 Backbone

3.1 Simile Triple Completion

As shown in Figure 1, the simile triple complete consists of two tasks: simile interpretation (SI) and simile generation (SG). Each simile sentence can be abstracted into the form of a triple. Therefore, we define a triple: $(\mathcal{T}, \mathcal{A}, \mathcal{V})$, where $\mathcal{T}$, $\mathcal{V}$ are mainly nouns or noun phrases and represent the tenor and vehicle in the simile sentence, respectively. $\mathcal{A}$ is the attribute in simile sentences, which is an adjective. If the $\mathcal{A}$ is None in the triple, i.e. $(\mathcal{T}, None, \mathcal{V})$, we define it as the simile interpretation task. Similarly, if the $\mathcal{V}$ is None, i.e. $(\mathcal{T}, \mathcal{A}, None)$, this will be the task of simile generation.

3.2 Masked Language Model

The masked language model (MLM) (Devlin et al., 2019; Taylor, 1953) randomly masks the words in the input sentence and feeds the masked sentence into the pre-trained models to make predictions by other visible words. For example, given a sentence $s = [w_1, w_2, \dots, w_i, \dots, w_m]$, where the w_i means the i-th word in the sentence. We can randomly mask s and feed the masked sequence $\widetilde{s}$ into the PLMs e.g. BERT (Devlin et al., 2019) to obtain the masked words by Equation:

$$\widetilde{s} = f_{mask}(s, i, v) \tag{1}$$

$$P = f_{\theta}(\widetilde{s}) \tag{2}$$

where the v means the Vocabulary for pre-trained models, and the i denotes the position of the masked word in Equation 1. The θ is the parameters of PLMs in Equation 2. We can select the word corresponding to the maximum probability in P as the output of the model.

3.3 Probe Simile Knowledge with MLM

To probe the simile knowledge in pre-trained masked language models, the intuitive solution is: (1) Construct a sentence that contains the simile triple in Section 3.1 with the given pattern. (2) Mask the attribute $\mathcal{A}$ or vehicle $\mathcal{V}$ in this simile sentence. (3) Predict the words in the masked position with MLM. For example, when given a pattern *The $\mathcal{T}$ is as $\mathcal{A}$ as $\mathcal{V}$*, the input sentence of MLM is *The $\mathcal{T}$ is as [MASK] as $\mathcal{V}$* for the SI task while *The $\mathcal{T}$ is as $\mathcal{V}$ as [MASK]* for the SG task.

To formulate this problem, we define the pattern function as $p(\tau)$, where $\tau \in \{SG, SI\}$. The pre-trained MLM is denoted as $\mathcal{M}$ and the predicted distribution Q over vocabulary V can be formulated as:

$$Q(w|p(\tau)) = \frac{exp(\mathcal{M}(w|p(\tau)))}{\sum_{w' \in V} exp(\mathcal{M}(w'|p(\tau)))} \quad (3)$$

4 Method

In this section, we will introduce our proposed method of probing simile knowledge from pre-trained models. Our method first introduces a secondary pre-training stage - Adjective-Noun mask Training (ANT) based on pre-trained language models to acquire diverse lexical-specific words. Then two modules of pattern ensemble and pattern search are used to obtain the high-quality predictions. The framework of our method is shown in Figure 2 in detail[4].

4.1 Adjective-Noun Mask Training (ANT)

For the MLM task, pre-trained models prefer to output high-frequency words as candidate words since the objective of the training is to minimize the cross-entropy loss (Gehrmann et al., 2019). However, the components of simile triples are usually nouns or adjectives and the simile sentences are appealing due to their creativity and unexpectedness. Therefore, to predict more diverse and specific words of simile component, we introduce a secondary pre-training stage - Adjective-Noun mask Training (ANT) that fine-tune the pre-trained model with specially designed datasets. First, we utilize *trankit* (Nguyen et al., 2021) to construct the training set by selecting sentences from BookCorpus (Zhu et al., 2015) that contains *amod*[5] dependencies (Nivre et al., 2017). Second, we mask a word at the end of *amod* relation, instead of randomly masking, and all words are masked no more than 5 times. Finally, the pre-trained model is fine-tuned on the constructed dataset with MLM loss. In this way, the pre-trained model will avoid the bias to high-frequency words and have a higher probability of generating diverse and novel words.

4.2 Pattern Ensemble (PE)

Since words predicted by MLM have a preference for different patterns and only using one pattern is insufficient, we apply the pattern ensemble to obtain better performance where different types of patterns are designed as shown in Table 1. Specifically, the class I describes the relationship between the three-element $\mathcal{T}$, $\mathcal{V}$ and $\mathcal{A}$. However, the similes tend to highlight an obvious attribute between tenor and vehicle (Israel et al., 2004). We further design the class II and class III to find the attribute corresponding to the tenor and vehicle, respectively. Finally, the attributes of simile sentences are sometimes omitted and thus the class IV is designed to deal with this case. Additionally, we also design three patterns for each class to obtain high-quality and robust results.

The output distribution Q_{PE} of pattern ensemble can be formulated as

$$Q_{PE}(w|P) = \frac{1}{|P|} \sum_{p(\tau) \in P} log(Q(w|p(\tau))) \quad (4)$$

where P is the set of patterns $p(\tau)$ for specific task τ. Note that though we design four classes of patterns in Table 1, some classes of patterns are not required for the SI or SG task. Specifically, The patterns of Class IV are not used for the SI task because the attribute $\mathcal{A}$ is missed in Class IV. Likewise, the patterns of Class III are not used for the SG task due to the lack of vehicle $\mathcal{V}$.

[4] we released our code at https://github.com/nairoj/Probing-Simile-from-PLM.

[5] An adjectival modifier of a noun (or pronoun) is any adjectival phrase that serves to modify the noun (or pronoun). The relation applies whether the meaning of the noun is modified in a compositional way (e.g., large house) or an idiomatic way (hot dogs).

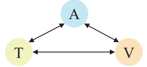
Class	Relationship	Pattern	
I		The {tenor} is as {attribute} as {vehicle}.	p_1
		{vehicle} is very {attribute}, so as {tenor}.	p_2
		{tenor} is like {vehicle}, because they are both {attribute}.	p_3
II		The {attribute} {vehicle}.	p_4
		{vehicle} is very {attribute}.	p_5
		{vehicle} is {attribute}.	p_6
III		The {attribute} {tenor}.	p_7
		{tenor} is very {attribute}.	p_8
		{tenor} is {attribute}.	p_9
IV		{tenor} is similar to {vehicle}.	p_{10}
		{tenor} is like {vehicle}.	p_{11}
		{tenor} and {vehicle} are alike.	p_{12}

Table 1: All patterns and corresponding classes. Class I models the relationship between three elements, and other classes model relationships between two elements. Every pattern is denoted as the right side symbol p_i.

4.3 Pattern Search (PS)

The prediction of pattern ensemble in Section 4.2 is averaged by adding up the output distributions of all the patterns. Conversely, the hand-designed patterns are heuristic, which may lead to suboptimal results. Therefore, it is worth studying how these patterns can be combined to obtain better performance. To solve this problem, we introduce an approach of pattern search (PS) to find the best combination of different patterns. Specifically, given a simile dataset $\mathcal{D}_{PS}$, we calculate Equation 4 on $\mathcal{D}_{PS}$ by iterating all subsets of the patterns. Finally, we select the optimal subset p_{best} as the input of MLM to predict simile components.

5 Experiments

5.1 Dataset

Dataset for ANT: We constructed our train set of ANT from BookCorpus. We first extracted the sentences with length less than 64 and then masked nouns or adjectives in them based on *amod* dependencies (Nivre et al., 2017). Meanwhile, we limited the frequency of masked words to less than 5. Finally, we got 98k sentences as the dataset of ANT, which contains 68k noun-masked sentences and 30k adjective-masked sentences.

Dataset for PE and PS: We evaluate our method on the dataset proposed in (Roncero and de Almeida, 2015). As the samples in Table 2, there are multiple attributes for each ($\mathcal{T}$, $\mathcal{V}$) pair. For example, the pair of (anger, fire) has the attributes of dangerous, hot, and red. In addition, we followed the previous work (Xiao et al., 2016) to filter the dataset by reversing simile triples with attribute frequencies greater than 4. Eventually, we obtain the train set with 533 samples and the test set with 145 samples. Notice that the train set is the $\mathcal{D}_{PS}$ in Section 4.3 used for the pattern search and the test set is used for evaluating all the approaches in this paper.

Triple	Frequency
(Anger, Dangerous, Fire)	8
(Anger, Hot, Fire)	8
(Anger, Red, Fire)	5
(Love, Beautiful, Rainbow)	10
(Love, Beautiful, Melody)	2
(Love, Beautiful, Rose)	9

Table 2: Some samples from the dataset. Frequency represents the number of annotators who consider the attribute is suitable for the Tenor-Vehicle pair.

5.2 Implementation Details

Details for ANT: In adjective-nouns mask training, we utilized Adam as our optimizer and the learning rate is 5e-5. The batch size is set to 32 and the max sequence length is set to 64, respectively. Further, we utilize the Bert-Large[6] with 340M parameters as the basic model to perform adjective-nouns mask training and the number of training epoch is 3.

Vehicle Filtering: For simile generation, we filter the predicted vehicles that are similar to the tensor by calculating the semantic similarity with Glove embedding. For instance, given the sentence “The child is as tall as [MASK]", we will filter out the word “father" as its vehicle due to not meeting the simile definition[7]. To solve this problem, we compute the similarity score of the tenor and vehicle and filter the predicted vehicle whose score is above the threshold 0.48[8].

[6] https://huggingface.co/Bert-large-uncased

[7] Using something with similar characteristics to give an analogy to another thing

[8] The threshold is the maximum similarity score of tenor and vehicle in the train set

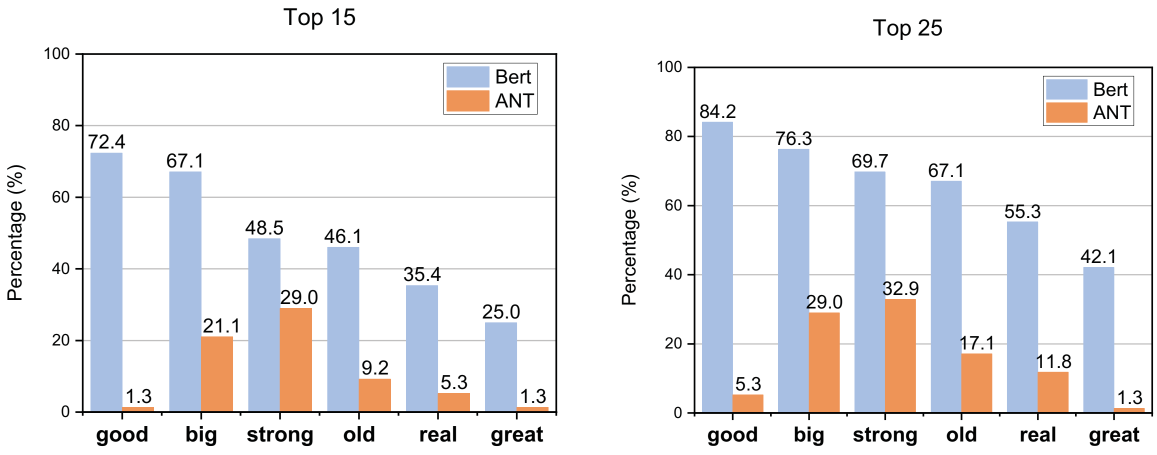


Figure 3: Percentage of samples whose top K predicted words contain a given common word. The horizontal coordinates are some common adjectives.

5.3 Evaluating the effectiveness of ANT

In this section, we will demonstrate that ANT could improve the diversity of predicted words for both the SI and SG tasks. We compare the predicted results of MLM (i.e., Bert) before and after ANT, which use the patterns "The $\mathcal{T}$ is as [MASK] as $\mathcal{V}$" for the SI task and "The $\mathcal{T}$ is as $\mathcal{A}$ as [MASK]" for the SG task.

Metric: We evaluate the diversity of the MLM predictions by calculating the proportion of unique words in the predicted Top K results on the test set. It can be formulated as

$$p@K = \frac{Num(Unique\_words)}{K * N} \tag{5}$$

where the $Num(Unique\_words)$ means the number of unique words, and the N represents size of the test set.

Result: To illustrate the effectiveness of ANT, We evaluate the results on the test set based on Equation 5. As shown in Table 3, the diversity of predicted words significantly improves after ANT for different $p@K$, specifically about 100% improvement for the SI task and about 50% for the SG task. Additionally, Figure 3 plots the percentage of samples on the test set, where a given common word (e.g., good, big, strong) appears in the list of the top $k = 15, 25$ predicted words. We can observe that the frequency of common words decreases significantly after ANT. For example, the frequency of the common word *good* decreases from 72.37% to 1.32% when $k = 15$.

	Method	p@5	p@10	p@15	p@25
SI	Bert	0.263	0.216	0.189	0.163
	ANT	0.492	0.412	0.382	0.312
SG	Bert	0.232	0.201	0.182	0.158
	ANT	0.370	0.299	0.256	0.216

Table 3: The results of diversity on both the SI and SG tasks. The method *Bert* and *ANT* separately represent the results before and after the Adjective-Noun mask training.

5.4 Evaluating the effectiveness of PE and PS

5.4.1 Baselines

We compare the proposed approaches with the following baseline:

(1) **Meta4meaning (Xiao et al., 2016)**: It uses the trained LSA vector representation according to the degree of abstraction and salience imbalance to select appropriate attributes. (2) **GEM (Bar et al., 2018)**: A method calculates the cosine similarity and normalized PMI between each attribute and tensor/vehicle based on Glove representing to obtain the best attribute with ranking. (3) **Bert (Devlin et al., 2019)**: Directly use pre-trained MLM to predict the simile component with a single pattern as Section 3.3. In this paper, we utilize the *bert-large-uncased* as the basic pre-trained MLM. (4) **ConScore (Zheng et al., 2020)**: A connecting score is proposed to select an attribute word $\mathcal{A}$ for $\mathcal{T}$ and $\mathcal{V}$.

Our proposed approaches are denoted as:

(1) **ANT**: Perform Adjective-Noun mask Training based on a pre-trained MLM with the datasets mentioned in Section 5.1. (2) **ANT+PE**: Based on ANT, the output distribution over vocabulary is pre-

Task	Method	MRR	R@5	R@10	R@15	R@25	R@50
SI	Meta4meaning	N/A	0.221	0.303	0.339	0.397	0.454
	GEM	**0.312**	0.198	0.254	0.278	0.405	0.562
	ConScore	0.078	0.076	0.138	0.172	0.269	0.386
	Bert	0.266	$\underline{0.338}$	$\underline{0.428}$	0.448	0.538	$\underline{0.641}$
	ANT	0.245	0.310	0.407	$\underline{0.455}$	0.510	0.614
	ANT+PE	0.241	0.331	0.400	0.448	$\underline{0.552}$	0.628
	ANT+PS+PE	$\underline{0.270}$	**0.379**	**0.490**	**0.524**	**0.579**	**0.655**
SG	ConScore	0.036	0.055	0.09	0.103	0.145	0.200
	Bert	$\underline{0.064}$	$\underline{0.076}$	$\underline{0.124}$	**0.159**	$\underline{0.207}$	0.283
	ANT	0.049	0.069	0.117	$\underline{0.145}$	0.186	**0.303**
	ANT+PE	0.036	0.034	0.083	0.097	0.131	0.172
	ANT+PS+PE	**0.095**	**0.124**	**0.145**	**0.159**	**0.214**	$\underline{0.290}$

Table 4: Automatic evaluation for SI and SG tasks. The best results are in bold, and the second best results are underlined.

Task	Method	Top5	Top10	Top15
SI	ConScore	0.192†	0.169†	0.172†
	Bert	0.411†	0.364†	0.326†
	ANT	0.471	0.396†	0.365†
	ANT+PE	$\underline{0.494}$	**0.469**	**0.456**
	ANT+PS+PE	**0.496**	$\underline{0.433}$†	$\underline{0.398}$†
SG	ConScore	0.780†	0.690†	0.673†
	Bert	0.597†	0.667†	0.629†
	ANT	0.867†	$\underline{0.868}$†	$\underline{0.808}$†
	ANT+PE	$\underline{0.887}$†	0.805†	0.751†
	ANT+PS+PE	**1.123**	**1.052**	**0.973**

Table 5: The average score of human evaluation for STC. The best results are in bold, and the second best results are underlined. † denotes significant difference with the best result (t-test, p-value<0.05).

dicted by average on all the corresponding patterns in Table 1. (3) **ANT+PS+PE**: Based on ANT, first the pattern search is to decide which patterns in Table 1 are applied, and then the pattern ensemble is used over these selected patterns.

5.4.2 Metrics

We use both automatic evaluation and human evaluation to compare our approaches with baselines.

Automatic Evaluation:

(1) Mean Reciprocal Rank (MRR): average on the reciprocal of the ranking r_i of label words in the predicted candidates, denoted as

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i} \tag{6}$$

(2) $R@K$: the percentage of the label words appear in the top K predictions. Note that, following previous works (Xiao et al., 2016; Bar et al., 2018), we consider a predicted word as the correct answer if it is a synonym of label word n in WordNet (Miller, 1992). It can be formulated as

$$cor(w) = \begin{cases} 1 & w \in Synonyms(L_i) \\ 0 & w \notin Synonyms(L_i) \end{cases} \tag{7}$$

$$R@K = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{w \in K_i} cor(w)}{K} \tag{8}$$

where K_i denotes the list of predicted words, L_i denotes the list of label words and $Synonyms(L_i)$ represents the synonyms of a word.

Human Evaluation: To further prove our approaches are better than baselines, human evaluation is used to evaluate the quality of predicted simile triples from three levels (0, 1, 2). 0 - The triple is unacceptable. 1 - The triple is acceptable. 2 - The triple is acceptable and creative. Given a simile triple, annotators need to score it according to their subjective judgment and each triple is annotated by three annotators independently. We use the average score of three annotators as the quality of a simile triple.

5.4.3 Results

Automatic and Human Evaluation: The results of both automatic and human evaluation are shown in Table 4 and Table 5. The agreement between annotators is measured using Fleiss's kappa κ (Randolph, 2005). The κ value is 0.68 (substantial agreement) for the SI task and 0.48 (moderate agreement) for the SG task.

From the results, we can conclude

Task	Subset of Patterns	MRR	R@5	R@10	R@15	R@25
SI	$\{p_1, p_5\}$	**0.100**	**0.126**	**0.184**	**0.233**	**0.281**
	$\{p_1, p_2, p_3, p_4, p_5, p_9\}$	0.095	0.107	0.171	0.203	0.268
	$\{p_1, p_2, p_3, p_4, p_5, p_7, p_8\}$	0.095	0.099	0.163	0.206	0.274
	$\{p_1, p_4, p_5\}$	0.094	0.094	0.163	0.203	0.261
SG	$\{p_3, p_4\}$	**0.056**	0.068	**0.105**	**0.135**	**0.159**
	$\{p_1, p_4\}$	0.056	**0.071**	0.092	0.120	0.154
	$\{p_1, p_3, p_4\}$	0.052	0.06	0.105	0.128	0.163
	$\{p_1, p_2, p_4\}$	0.052	0.058	0.096	0.116	0.137

Table 6: The top 4 best performing pattern subsets for SI and SG tasks (see Table 1 for which class the pattern p_i belongs to). The best results are in bold. More results of pattern search are shown in the Appendix A
.

(1) For both SI and SG tasks, our proposed approaches (i.e., ANT, ANT+PE, ANT+PS+PE) significantly outperform the baselines on both automatic and human evaluations. It proves that our methods not only enhance the diversity of predicted simile components in Section 5.3 but also their quality.

(2) Pre-trained MLM-based methods (i.e., Bert, ANT, ANT+PE and ANT+PS+PE) perform better than the traditional methods (i.e., GEM, Meta4meaning, ConScore). It shows the potential of pre-trained models in probing simile knowledge.

(3) Compared ANT with Bert, we found that though ANT improves the diversity of predicted words in Table 3, the average scores on automatic and human evaluations decrease because the simile knowledge is not involved in the ANT training process. However, our proposed PE and PS compensate for the performance.

(4) The scores of automatic evaluation metrics on the SI task are remarkably higher than the SG task. Yet, the scores of human evaluation metrics are significantly lower than on the SG task. We conjecture that this may be because the list of candidate words of attribute predicted by SI are smaller than that of the vehicle for the SG task. For example, given the SI sample "(Cloud, $None$, Cotton)", the attribute words are almost restricted to the physical properties of the vehicle, such as "Soft", while the choices of vehicle words are more varied and unexpected given the SG sample "(Cloud, soft, $None$)" such as "cotton, silk, towel".

Discussion for PS: Compared ANT+PS+PE to ANT+PE, it can be included that pattern search brings a great improvement to the results on both automatic and human evaluations. To have a deeper insight into PS, the pattern subsets with high performance are listed in Table 6. For the SI task, the optimal multi-pattern combination is $\{p_1, p_5\}$, which support the hypothesis proposed by (Ortony, 1979) considers that the highlighted attribute of a simile triple is more salient in the vehicle domain despite it is commonly shared by both tenor and vehicle domains. Specifically, pattern p_1 belonging to the Class I, models the relationship of all three simile components while the pattern p_5 belonging to Class II requires the candidate words to be the salient attribute of the vehicle. Similarly, for SG task, optimal multi-pattern combination is $\{p_3, p_4\}$, which is also a combination of the Class I pattern and the Class II pattern.

6 Conclusion and Future work

In this paper, from the perspective of simile triple completion, we propose a unified framework to solve the SI and SG tasks by probing the knowledge of the pre-trained masked language model. The backbone of our method is to construct masked sentences with manual patterns from an incomplete simile triple, and then use language models with MLM heads to predict the masked words. Moreover, a secondary pre-training stage (the adjective-noun mask training) is applied to improve the diversity of predicted words. Pattern ensemble (PE) and pattern search (PS) are further used to improve the quality of predicted words. Finally, automatic and human evaluations demonstrate the effectiveness of our framework in both SI and SG tasks. In future work, we will continue to study how to mine broader or complex knowledge from pretrained models, such as metaphor, common sense and we expect more researchers to perform related research.

Acknowledgements

This work is supported by the Key Research and Development Program of Zhejiang Province (No. 2022C01011) and National Natural Science Foundation of China (Project 61075058). We would like to thank the anonymous reviewers for their excellent feedback.

References

Keiga Abe, Kayo Sakamoto, and Masanori Nakagawa. 2006. A computational model of the metaphor generation process. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 28.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Kfir Bar, Nachum Dershowitz, and Lena Dankin. 2018. Metaphor interpretation using word embeddings. *International Journal of Computational Linguistics and Applications*.

Danushka Bollegala and Ekaterina Shutova. 2013. Metaphor interpretation using paraphrases extracted from the web. *PloS one*, 8(9):e74304.

Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. 2020. Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6455–6469, Online. Association for Computational Linguistics.

Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. 2021. MERMAID: Metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4250–4261, Online. Association for Computational Linguistics.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Andrea Gagliano, Emily Paul, Kyle Booten, and Marti A. Hearst. 2016. Intersecting word vectors to take figurative language to new heights. In *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*, pages 20–31, San Diego, California, USA. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Katy Ilonka Gero and Lydia B. Chilton. 2019. Metaphoria: An algorithmic companion for metaphor creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, page 296. ACM.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. BERTese: Learning to speak to BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3618–3623, Online. Association for Computational Linguistics.

Raquel Hervás, Rui Ponte Costa, Hugo Costa, Pablo Gervás, and Francisco C. Pereira. 2007. Enrichment of automatically generated texts using metaphor. In *MICAI 2007: Advances in Artificial Intelligence, 6th Mexican International Conference on Artificial Intelligence, Aguascalientes, Mexico, November 4-10, 2007, Proceedings*, volume 4827 of *Lecture Notes in Computer Science*, pages 944–954. Springer.

Michael Israel, Jennifer Riddle Harding, and Vera Tobin. 2004. On simile. *Language, culture, and mind*, 100.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer

Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ArXiv preprint*, abs/2107.13586.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *ArXiv preprint*, abs/2103.10385.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proc. Natl. Acad. Sci. USA*, 117(48):30046–30054.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Andrew Ortony. 1979. Beyond literal similarity. *Psychological review*, 86(3):161.

Anthony M. Paul. 1970. Figurative language. *Philosophy & Rhetoric*, 3(4):225–248.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ashequl Qadir, Ellen Riloff, and Marilyn A. Walker. 2016. Automatically inferring implicit properties in similes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1223–1232, San Diego, California. Association for Computational Linguistics.

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

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Online submission*.

Vassiliki Rentoumi, George A. Vouros, Vangelis Karkaletsis, and Amalia Moser. 2012. Investigating metaphorical language in sentiment analysis: A sense-to-sentiment perspective. *ACM Trans. Speech Lang. Process.*, 9(3):6:1–6:31.

Carlos Roncero and Roberto G de Almeida. 2015. Semantic properties, aptness, familiarity, conventionality, and interpretive diversity scores for 84 metaphors and similes. *Behavior research methods*, 47(3):800–812.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Ekaterina Shutova. 2010. Automatic metaphor interpretation as a paraphrasing task. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1029–1037, Los Angeles, California. Association for Computational Linguistics.

Wei Song, Jingjin Guo, Ruiji Fu, Ting Liu, and Lizhen Liu. 2021. A knowledge graph embedding approach for metaphor processing. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:406–420.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. 2021. Metaphor generation with conceptual mappings. In

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6724–6736, Online. Association for Computational Linguistics.

Chang Su, Shuman Huang, and Yijiang Chen. 2017. Automatic detection and interpretation of nominal metaphor based on the theory of meaning. *Neurocomputing*, 219:300–311.

Chang Su, Jia Tian, and Yijiang Chen. 2016. Latent semantic similarity based interpretation of chinese metaphors. *Eng. Appl. Artif. Intell.*, 48:188–203.

Wilson L. Taylor. 1953. “cloze procedure”: A new tool for measuring readability. *Journalism & Mass Communication Quarterly*, 30:415 – 433.

Xiaoyu Tong, Ekaterina Shutova, and Martha Lewis. 2021. Recent advances in neural metaphor processing: A linguistic, cognitive and social perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4673–4686, Online. Association for Computational Linguistics.

Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov. 2016. *Metaphor: A Computational Perspective*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Jiaan Wang, Fandong Meng, Ziyao Lu, Duo Zheng, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022a. Clidsum: A benchmark dataset for cross-lingual dialogue summarization. *ArXiv*, abs/2202.05599.

Jiaan Wang, Beiqi Zou, Zhixu Li, Jianfeng Qu, Pengpeng Zhao, An Liu, and Lei Zhao. 2022b. Incorporating commonsense knowledge into story ending generation via heterogeneous graph networks. *ArXiv*, abs/2201.12538.

Ping Xiao, Khalid Al-Najjar, Mark Granroth-Wilding, Kathleen Agres, and Hannu Toivonen. 2016. Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In *Proceedings of the Seventh International Conference on Computational Creativity, ICCC 2016, UPMC, Paris, France, June 27 - July 1, 2016*, pages 230–237. Sony CSL Paris, France.

Zhiwei Yu and Xiaojun Wan. 2019. How to avoid sentences spelling boring? towards a neural approach to unsupervised metaphor generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 861–871, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiali Zeng, Linfeng Song, Jinsong Su, Jun Xie, Wei Song, and Jiebo Luo. 2020. Neural simile recognition with cyclic multitask learning and local attention. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9515–9522. AAAI Press.

Jiayi Zhang, Zhi Cui, Xiaoqiang Xia, Yalong Guo, Yanran Li, Chen Wei, and Jianwei Cui. 2021. Writing polishment with simile: Task, dataset and A neural approach. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14383–14392. AAAI Press.

Danning Zheng, Ruihua Song, Tianran Hu, Hao Fu, and Jin Zhou. 2020. "love is as complex as math": Metaphor generation system for social chatbot. *ArXiv preprint*, abs/2001.00733.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

A More results of Pattern Search

The more results of Pattern Search are shown in Table 7.

B More Prediction

Some results are shown in Table 8 and Table 9.

Task	Subset of Patterns	MRR	R@5	R@10	R@15	R@25	R@50
SI	$\{p_1, p_5\}$	0.100	0.126	0.184	0.233	0.281	0.375
	$\{p_1, p_2, p_3, p_4, p_5, p_9\}$	0.095	0.107	0.171	0.203	0.268	0.377
	$\{p_1, p_2, p_3, p_4, p_5, p_7, p_8\}$	0.095	0.099	0.163	0.206	0.274	0.373
	$\{p_1, p_4, p_5\}$	0.094	0.094	0.163	0.203	0.261	0.366
	$\{p_1, p_2, p_4, p_5, p_9\}$	0.094	0.111	0.165	0.214	0.265	0.373
	$\{p_1, p_4, p_5, p_6, p_8\}$	0.093	0.109	0.171	0.212	0.283	0.368
	$\{p_1, p_4, p_5, p_6, p_7, p_8, p_9\}$	0.093	0.090	0.152	0.205	0.263	0.338
	$\{p_1, p_2, p_4, p_5\}$	0.093	0.113	0.167	0.210	0.280	0.371
	$\{p_1, p_2, p_4, p_6, p_8\}$	0.093	0.111	0.178	0.218	0.272	0.371
	$\{p_1, p_2, p_4, p_5, p_6, p_8\}$	0.093	0.105	0.173	0.216	0.283	0.370
	$\{p_1, p_2, p_4, p_5, p_6, p_7, p_8, p_9\}$	0.093	0.096	0.156	0.210	0.261	0.347
	$\{p_1, p_3, p_4, p_5, p_8, p_9\}$	0.093	0.098	0.159	0.223	0.265	0.368
	$\{p_1, p_5, p_6, p_7\}$	0.092	0.101	0.163	0.203	0.274	0.366
	$\{p_1, p_2, p_5, p_9\}$	0.092	0.099	0.171	0.225	0.285	0.362
	$\{p_1, p_2, p_4, p_5, p_8\}$	0.092	0.105	0.173	0.218	0.280	0.360
	$\{p_1, p_2, p_4, p_5, p_8, p_9\}$	0.092	0.099	0.158	0.216	0.274	0.360
	$\{p_1, p_2, p_4, p_5, p_6, p_8, p_9\}$	0.092	0.094	0.159	0.210	0.270	0.355
	$\{p_1, p_3, p_5\}$	0.092	0.105	0.169	0.216	0.280	0.381
	$\{p_1, p_3, p_4, p_5, p_8\}$	0.092	0.096	0.173	0.220	0.276	0.368
	$\{p_1, p_2, p_3, p_5, p_9\}$	0.092	0.107	0.165	0.208	0.281	0.371
	$\{p_1, p_5, p_6\}$	0.091	0.116	0.180	0.220	0.283	0.385
	$\{p_1, p_5, p_6, p_8\}$	0.091	0.111	0.174	0.229	0.278	0.364
	$\{p_1, p_4, p_5, p_8\}$	0.091	0.103	0.176	0.216	0.291	0.366
	$\{p_1, p_4, p_5, p_8, p_9\}$	0.091	0.099	0.165	0.205	0.270	0.358
SG	$\{p_3, p_4\}$	0.056	0.068	0.105	0.135	0.159	0.223
	$\{p_1, p_4\}$	0.056	0.071	0.092	0.120	0.154	0.225
	$\{p_1, p_3, p_4\}$	0.052	0.060	0.105	0.128	0.163	0.218
	$\{p_1, p_2, p_4\}$	0.052	0.058	0.096	0.116	0.137	0.197
	$\{p_1, p_4, p_5\}$	0.052	0.064	0.094	0.114	0.137	0.203
	$\{p_3, p_4, p_{11}\}$	0.050	0.058	0.079	0.099	0.141	0.186
	$\{p_1, p_4, p_6\}$	0.049	0.051	0.086	0.105	0.131	0.197
	$\{p_3, p_4, p_5\}$	0.048	0.058	0.096	0.114	0.144	0.208
	$\{p_3, p_4, p_6\}$	0.048	0.051	0.094	0.109	0.135	0.199
	$\{p_1, p_3, p_4, p_5\}$	0.048	0.049	0.092	0.120	0.148	0.208
	$\{p_1, p_3, p_4, p_6\}$	0.048	0.054	0.090	0.111	0.137	0.214
	$\{p_1, p_3, p_4, p_{11}\}$	0.048	0.062	0.088	0.105	0.128	0.188
	$\{p_2, p_3, p_4\}$	0.047	0.062	0.090	0.105	0.133	0.197
	$\{p_1, p_2, p_4, p_6\}$	0.047	0.051	0.084	0.113	0.146	0.184
	$\{p_1, p_2, p_4, p_5\}$	0.047	0.054	0.083	0.113	0.141	0.188
	$\{p_1, p_2, p_3, p_4\}$	0.046	0.058	0.088	0.109	0.133	0.206
	$\{p_1, p_2, p_3, p_4, p_5\}$	0.046	0.054	0.083	0.096	0.131	0.188
	$\{p_4, p_{11}\}$	0.046	0.053	0.081	0.099	0.122	0.171
	$\{p_1, p_3, p_4, p_5, p_{12}\}$	0.046	0.058	0.079	0.094	0.114	0.171
	$\{p_1, p_4, p_{11}\}$	0.045	0.053	0.084	0.101	0.139	0.208
	$\{p_1, p_2, p_3, p_4, p_{11}\}$	0.045	0.060	0.084	0.099	0.118	0.169
	$\{p_1, p_3, p_4, p_5, p_6\}$	0.045	0.047	0.083	0.116	0.137	0.184
	$\{p_1, p_4, p_5, p_6\}$	0.045	0.049	0.079	0.101	0.133	0.189
	$\{p_1, p_4, p_5, p_{11}\}$	0.045	0.045	0.077	0.096	0.133	0.186

Table 7: The top 25 best performing pattern subsets for SI and SG tasks, sorted according to MRR. See Table 1 for which class the pattern p_i belongs to.

Triple	Score
(anger, burning, fire)	2.00
(cities, humid, jungles)	2.00
(clouds, fluffy, cotton)	2.00
(deserts, hot, ovens)	2.00
(exams, tough, hurdles)	2.00
(families, powerful, fortresses)	2.00
(fingerprints, accurate, portraits)	2.00
(highways, crooked, snakes)	2.00
(love, pure, flower)	2.00
(anger, blazing, fire)	1.67
(love, romantic, melody)	1.67
(money, valuable, oxygen)	1.67
(obligations, binding, shackles)	1.67
(teachers, creative, sculptors)	1.67
(time, important, money)	1.67
(tv, addicted, drug)	1.67
(wisdom, infinite, ocean)	1.67
(desks, messy, junkyards)	1.33
(eyelids, close, curtains)	1.33
(god, benevolent, parent)	1.33
(music, soothing, medicine)	1.33
(skating, relaxing, flying)	1.33
(friendship, lovely, rainbow)	1.00
(life, challenging, journey)	1.00
(love, sweet, flower)	1.00
(love, fragile, rose)	1.00
(pets, annoying, kids)	1.00
(television, attractive, candy)	1.00
(women, quiet, cats)	1.00
(trust, secure, glue)	0.67
(tv, harmful, drug)	0.67
(tree trunks, weak, straws)	0.67
(trees, sturdy, umbrellas)	0.67
(winter, long, death)	0.33
(tongues, spicy, fire)	0.33
(typewriters, obsolete, dinosaurs)	0.00
(time, quick, snail)	0.00
(trees, long, umbrellas)	0.00
(tv, ineffective, drug)	0.00
(tv, unreliable, drug)	0.00

Table 8: Some results of simile interpretation. Score is the average score of human evaluation.

Triple	Score
(clouds, white, cream)	2.00
(friendship, colorful, jewelry)	2.00
(love, colorful, coral)	2.00
(love, shiny, pearl)	2.00
(skating, relaxing, noon)	2.00
(tv, addictive, drug)	2.00
(dreams, clear, crystal)	1.67
(friendship, colorful, sunrise)	1.67
(love, addictive, coke)	1.67
(love, colorful, sunrise)	1.67
(music, cure, lullaby)	1.67
(clouds, white, pearl)	1.33
(dreams, clear, glass)	1.33
(exams, challenging, boxing)	1.33
(friendship, colorful, pottery)	1.33
(knowledge, important, faith)	1.33
(love, addictive, alcohol)	1.33
(love, colorful, lavender)	1.33
(music, cure, art)	1.33
(clouds, white, dove)	1.00
(desks, messy, nightmare)	1.00
(desks, messy, storage)	1.00
(highways, long, march)	1.00
(knowledge, important, time)	1.00
(love, addictive, poison)	1.00
(love, colorful, perfume)	1.00
(love, colorful, silk)	1.00
(music, cure, time)	1.00
(skating, relaxing, outdoors)	1.00
(typewriters, ancient, legend)	1.00
(cities, crowded, blast)	0.67
(knowledge, important, intuition)	0.67
(love, colorful, neon)	0.67
(clouds, white, bone)	0.33
(friendship, colorful, lightning)	0.33
(love, addictive, spice)	0.33
(cities, crowded, hell)	0.00
(clouds, white, steel)	0.00
(dreams, clear, stone)	0.00
(exams, challenging, robotics)	0.00

Table 9: Some results of simile generation. Score is the average score of human evaluation.