A Self-verified Method for Exploring Simile Knowledge from Pre-trained Language Models

Longxuan Ma,Changxin Ke,Shuhan Zhou,Churui Sun,Weinan Zhang*,Ting Liu
Research Center for Social Computing and Information Retrieval,
School of Computer Science, Harbin Institute of Technology
lxma,wnzhang,shzhou,tliu@ir.hit.edu.cn
cxke@stu.hit.edu.cn, sunchurui@hit.edu.cn

Abstract

Simile tasks are challenging in natural language processing (NLP) because models require adequate world knowledge to produce predictions. In recent years, pre-trained language models (PLMs) have succeeded in NLP since they learn generic knowledge from a large corpus. The knowledge embedded in PLMs can be used for different kinds of Simile tasks. However, previous work usually explored one type of simile knowledge for a specific simile task, how to fully utilize different types of knowledge embedded in the PLMs requires further exploration. This paper proposes a self-verified method for exploring simile knowledge from PLMs, which allows the PLMs to leverage one type of simile knowledge to self-validate another. To this end, we first enhance PLMs with a novel multi-level simile recognition (MLSR) task that trains PLMs to evaluate the quality of similes. Then the PLMs leverage this evaluation score to assist the simile interpretation and generation tasks. In this way, we connect different types of simile knowledge in PLMs and make better use of them. Experiments on different pre-trained models and multiple publicly available datasets show that our method works for different kinds of PLMs and can explore more accurate simile knowledge for PLMs. Our code/data will be released on GitHub.

Keywords: Simile, Pre-trained language models, Self-verified

1. Introduction

A simile is a common linguistic phenomenon in daily communication and plays an important role in human language to make utterances more vivid, interesting, and graspable (Niculae and Danescu-Niculescu-Mizil, 2014; Zhang et al., 2021). A simile compares two things from different categories (called the tenor and the vehicle) via shared properties (Paul, 1970). The tenors and the vehicles are usually connected with comparator words such as "like" or "as". Table 1 shows several examples of similes. For instance, the first sentence "The man is as strong as a bull." is a simile where the tenor is "The man", the vehicle is "a bull", the comparator is "as ... as" and the shared property is "strong". In contrast, the third sentence is not a simile since it compares two things in the same category. This kind of linguistic phenomenon is usually named literal in simile research (Aghazadeh et al., 2022; Maudslay and Teufel, 2022).

There are a variety kinds of simile tasks in NLP. Different kinds of simile tasks require models to possess different types of simile knowledge. For example, the simile recognition (SR) task is a binary classification task (Tsvetkov et al., 2014; Mohler et al., 2016; Steen, 2010; Li et al., 2022) which requires models to judge whether a text sequence (e.g. a triplet or a sentence) is simile or literal. The simile interpretation (SI) task requires models to generate an interpretation text for a simile (Bizzoni and Lappin, 2018) or infer the shared properties of the tenor and the vehicle (He et al., 2022; Chen et al., 2022). The simile generation (SG) task requires models to generate the missing vehicle (Song et al., 2021; Chen et al., 2022; Yang et al., 2023) or a simile sentence (Li et al., 2022; Chakrabarty et al., 2020; Stowe et al., 2021; Zhang et al., 2021).

	Examples	**Simile?**
1	The man is as strong as *a bull*.	Yes
2	Tom runs so fast, like *a rabbit*.	Yes
3	The girl looks like her mother.	No
4	The girl looks like *an angel*.	Yes
5	A: Arguing with parents is not wise. B: It is like *throwing an egg at a rock*.	Yes

Table 1: Simile and literal examples. Underlined font means tenors and italic font means *vehicles*.

In recent years, pre-trained language models (PLMs) have achieved great success in NLP since they learn generic knowledge from a large corpus (Devlin et al., 2019; Radford et al., 2019). In simile study, considerable efforts have also been made to explore simile knowledge from PLMs to solve SR/SI/SG tasks (Chen et al., 2022; He et al., 2022). For example, a widely adopted method is fine-tuning PLMs for a specific simile task (Chakrabarty et al., 2020; Chen et al., 2022; Li et al., 2022). However, previous work usually explored one type of simile knowledge to address a specific simile task,

*Corresponding author

LREC-COLING 2024, pages 1563–1576
20-25 May, 2024.

which did not fully leverage the ability of PLMs since the previous research had also shown that PLMs possessed multiple types of simile knowledge (Song et al., 2021; He et al., 2022, 2023).

In this paper, we study to leverage multiple types of simile knowledge for a simile task. Specifically, when using PLMs to perform a simile task (SI/SG), besides the simile knowledge that is directly related to this task, we expect the PLMs can also leverage other indirect simile knowledge (SR). The indirect simile knowledge can serve as a self-verification perspective to improve the performance on the target task, so as to fully leverage the multiple types of simile knowledge in PLMs. However, there is a gap between different types of simile knowledge. For example, the SR task requires the PLMs to distinguish simile and literal, and the SI/SG task requires the PLMs to generate missing simile components. The SR and SI/SG tasks are different in task format and training objects. It is difficult to directly leverage simile recognition knowledge to assist SI/SG tasks. To address this problem, we propose a novel task named multi-level simile recognition (MLSR) to align different types of simile knowledge. The MLSR task expanded the SR task from binary classification to multiple classification. After training with MLSR, the PLMs not only can distinguish simile and literal but also can assign a quality score for a given simile. Then this quality score can serve as a self-verification mechanism to evaluate the generation results of the SI/SG task and re-organize the results according to the evaluation. In this way, we integrate different types of simile knowledge in PLMs and make better utilization of them. Experiments on multiple simile datasets show that our method achieves new state-of-the-art performance. We also test our method with different PLMs, the results show that the self-verified method works for different kinds of PLMs and can be applied to language models with much larger sizes. To sum up, our contributions are:

- We propose a novel self-verified method to explore simile knowledge from PLMs.

- We propose a novel multi-level simile recognition (MLSR) task that helps the PLMs to evaluate the simile quality. The MLSR score serves as a self-verification mechanism to align simile knowledge used in different kinds of simile tasks.

- Experiments on multiple simile datasets and different PLMs show the effectiveness and scalability of our method. Our code and data will be released on GitHub[1].

Metaphor	**Example**	**Simile?**
N. phrase	The judge is like *an angel*.	Yes
Adjective	The boy has a warm heart.	No
Verbal	He kills the seeds of peace.	No
A.-verb	The child speaks France fluidly.	No
V. phrase	Taking care of little pets is like *raising children*.	Yes
Sentence	The man walks into the crowd like *a fish swims into the ocean*.	Yes

Table 2: Metaphor and simile categories. "N."/"A."/"V." means "Noun"/"Adverb"/"Verbal", respectively. Underlined font means tenors and italic font means *vehicles*.

2. Related Work

We introduce previous work related to this paper.

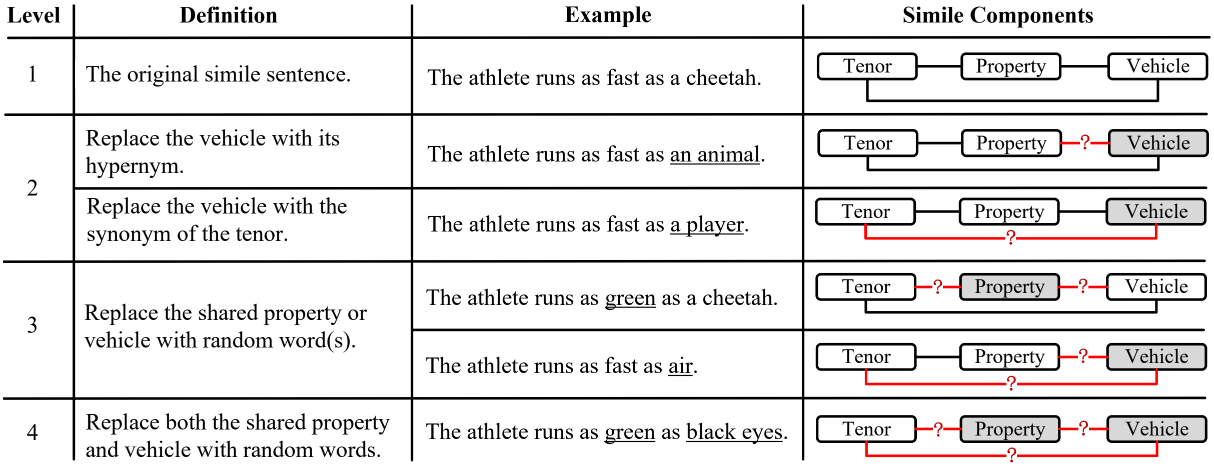
2.1. Simile and Metaphor

Metaphor is a figurative language that allows people to understand abstract concepts through concrete and familiar ones (Aghazadeh et al., 2022; Feng and Ma, 2022; He et al., 2023). Bizzoni and Lappin (2018) categorized metaphor into Noun phrases, Adjectives, Verbs, and Multi-word. Li et al. (2022) defined metaphor as Nominal, Verbal (Subject-Verb-Object), Adjective-Noun, and Adverb-Verb. Table 2 shows examples of these categories. The Noun phrase metaphor with comparator "like" or "as" is usually defined as a simile (Li et al., 2022; He et al., 2022; Chen et al., 2022). When simile happens in a more complex scenario such as human dialogue (Ma et al., 2023), verbal phrases/sentences can also function as the tenor or vehicle, such as the last two examples in Table 2. The commonly studied metaphor/simile tasks include recognition (Birke and Sarkar, 2006; Liu et al., 2018; Li et al., 2023a; Badathala et al., 2023; Zhang and Liu, 2023), interpretation (Su et al., 2016; Song et al., 2021), and generation (Li et al., 2022). In this paper, we propose a novel simile task named multi-level simile recognition (MLSR), which aims to train a simile model to evaluate the simile quality. The quality score can be used to align different types of simile knowledge and the shared semantic feature across different simile tasks.

2.2. Exploring Simile Knowledge in PLMs

Comprehending similes is not only essential to appreciate the inner connection between different concepts but also useful for other natural language processing (NLP) tasks such as sentiment analysis (Rentoumi et al., 2012), question answering (Zheng et al., 2019) and writing polishment (Zhang et al., 2021; He et al., 2023).

[1] https://github.com/malongxuan/MLSR



Level	Definition	Example	Simile Components
1	The original simile sentence.	The athlete runs as fast as a cheetah.	Tenor Property Vehicle
2	Replace the vehicle with its hypernym.	The athlete runs as fast as an animal.	Tenor Property ? Vehicle
	Replace the vehicle with the synonym of the tenor.	The athlete runs as fast as a player.	Tenor Property Vehicle ?
3	Replace the shared property or vehicle with random word(s).	The athlete runs as green as a cheetah.	Tenor ? Property ? Vehicle
		The athlete runs as fast as air.	Tenor Property ? Vehicle ?
4	Replace both the shared property and vehicle with random words.	The athlete runs as green as black eyes.	Tenor ? Property ? Vehicle ?

Figure 1: Different simile levels. The replaced words are underlined in the third column. The replaced component is shown in a gray box in the fourth column. A question mark indicates that the simile relationship between two components may be broken.

In recent years, PLMs-based approaches have become the de facto standard in NLP since they learn generic knowledge from a large corpus (Chen et al., 2022; Wachowiak and Gromann, 2023). Considerable attention has been paid to exploring simile knowledge from PLMs for resolving simile tasks. Song et al. (2021) fine-tune BERT (Devlin et al., 2019) for simile recognition and simile component (tenor, shared property, and vehicle) extraction. Chakrabarty et al. (2020) fine-tune BART (Lewis et al., 2020) to generate novel similes when giving a literal sentence. He et al. (2022) use a simile property probing task to infer the shared properties with the help of PLMs. Chen et al. (2022) propose an Adjective-Noun mask Training method to explore simile knowledge from BERT for simile interpretation and generation tasks. Li et al. (2022) fine-tune a GPT-2 (Radford et al., 2019) model for simile generation. Unlike previous work that fine-tuned PLMs on a specific simile task, which usually explored one type of simile knowledge, we propose a self-verified method to integrate simile knowledge from different training modes and better use the simile knowledge in PLMs.

3. Our Proposed Method

In this section, we introduce the MLSR task and our self-verified method in detail.

3.1. Multi-level Simile Recognition

Previous works (Li et al., 2022; Wang et al., 2022) defined SR as a binary classification task where the SR model needed to distinguish whether an input sequence was simile or literal. The only common feature between simile data and literal data is that they both contain the comparator words "like" or "as" (Liu et al., 2018). For example, the sentence "the athlete runs like a deer." is a simile and the sentence "the boy looks like his father." is a literal. When training PLMs as the SR models, the input sequence is first encoded into a vector as the sequence representation, then the classification score is calculated with the vector (Song et al., 2021). The output of the SR model is a binary label: True for simile and False for literal.

However, the traditional SR models are only trained to distinguish simile and literal, they can not assign a reasonable score to reflect the quality of a simile and serve as a self-verified mechanism for SI/SG tasks. To this end, we design a multi-level simile recognition (MLSR) task. We first construct simile data with multiple simile levels, then train the PLMs to assign quality scores for each level. Next, we will introduce how we obtain the multi-level simile data.

3.1.1. Multi-level Simile Data

A recent research (Chen et al., 2022) categorized the relations between simile elements into 4 classes and showed that data in different classes has different quality when fine-tuning PLMs. This inspires us that breaking the relations between simile elements can entail different simile qualities. In Figure 1, we define four simile levels with different simile qualities. The first level is the correct similes, which have three health simile relations among the simile components (as in the fourth column). The rest levels are constructed by replacing the simile components and cause varying scales of damage to the simile relations.

The second level has two replacement strategies. 1) Replace the vehicle with its hypernym. For example, "The athlete runs as fast as a cheetah." is changed to "The athlete runs as fast as an animal"

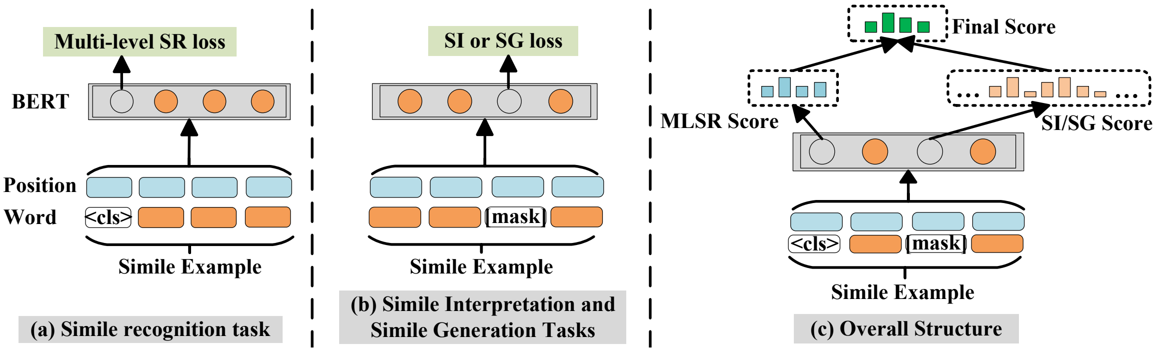


Figure 2: Demonstration of the training and testing with BERT.

in this level. The resulting sentence is still a simile that compares two different things with shared property. However, it is not as precise as the original one since not all animals "run fast". Although we understand this sentence is meant to compare "The athlete" to "an animal that can run fast", "a cheetah" is more concrete and makes the simile more vivid and graspable than "an animal". As shown in the fourth column of Figure 1, the relation between the shared property and the vehicle may be corrupted by this change. 2) Replace the vehicle with the synonym of the tenor. For instance, "The athlete runs as fast as a cheetah." is changed to "The athlete runs as fast as a player." in this level. Similes require comparing two things in different categories. "The athlete" and "a player" are in the same category, and the simile relation between the tenor and the vehicle is broken.

The third level replaces the vehicle or the shared property with a **Replacing rule** that will be introduced in the last part of this section. In Figure 1, "The athlete runs as fast as a cheetah." is changed to "The athlete runs as fast as air"/"The athlete runs as green as a cheetah." when replacing the vehicle/the shared property, respectively. When replacing the vehicle, the shared property ("fast") still belongs to the tenor ("The athlete") except it is unrelated to the vehicle ("air"); when replacing the shared property, the simile sentence is still comparing two different things ("The athlete" and "a cheetah") except the shared property ("green") is wrong. As shown in the fourth column of Figure 1, two out of three simile relations may be corrupted in the third level.

The fourth level replaces both the shared property and the vehicle with the **Replacing rule**. In Figure 1, "The athlete runs as fast as a cheetah." is changed to "The athlete runs as green as black eyes". Not only "green" is not a shared property between "The athlete" and "black eyes", but also "The athlete" and "black eyes" have no comparable properties. All three simile relations may be corrupted at this level.

Notice that in Figure 1, we keep the tenor and only replace property/vehicle for simply the demonstration. In practice, we can additionally keep the vehicle and change the tenor/property to obtain more training samples. For example, we can change "The athlete runs as fast as a cheetah." to "A leopard runs as fast as a cheetah." by replacing the tenor "The athlete" with the synonym of the vehicle "A leopard". This replacement breaks the simile relation between tenor and vehicle. The resulting sentence is still in the second level. We use the four simile levels in Figure 1 to obtain the multi-level simile data. Then we can train the MLSR model to evaluate simile quality.

Replacing rule. The words used for replacement follow certain rules. Firstly, the replacements use the words with the same part of speech. The shared properties are replaced with Adjectives and the tenor/vehicles are replaced with Nouns/phrases/sentences that can serve as the subject/object of a verb. Secondly, we leverage the relations in ConceptNet (Speer et al., 2017) to choose more proper words. In the second level, we use the "IsA" relation in ConceptNet to find the hypernym to the tenor/vehicle. In the fourth level, when replacing a vehicle, we choose Nouns/phrases that do not share properties with this vehicle; when replacing a property, we choose an Adjective that does not have a relation to the tenor and the vehicle. When the simile component is more than one word, we use the first Noun or Adjective in this component for replacement. When the simile components are not available in the ConceptNet or we cannot find a proper replacement, we only follow the first rule and randomly select words from ConceptNet.

3.1.2. Multi-level Simile Recognition Training

We adopt a multi-level contrastive learning method (Ye et al., 2021; Ma et al., 2022) to train the PLM. For each simile level in Figure 1, we assign a reference score between 0 and 1 (1.000, 0.667, 0.333, 0.000) to reflect the quality of the simile. During training, the MLSR model learns to predict a score for the input sequence. The training goal is to minimize the distance between the predicted score and the reference score. Notice that we hypothesize a linear relationship between the reference quality score and the simile level, which is correlated to the corruption of relations between simile elements as in Figure 1. This setting is convenient for model training but the scores do not accurately reflect simile qualities. A more reasonable way is to have the model automatically learn the quality scores at different levels. We leave the automatic learning for future work.

Figure 2 (a) shows the model structure of MLSR when using BERT as the backbone. The first output position (a special token <cls> before the input sequence) is used as the sequence representation to calculate the Multi-level SR loss. When using our method on other PLMs such as those with decoder structure (Raffel et al., 2020), the first decoding vector is used as the input representation to calculate the MLSR loss.

We denote the input sequence as **D** and the representation for **D** as h_D. The h_D is passed through a Multi-layer Perceptron (MLP) to get the predicted quality score $S^{\mathbf{D}}$ for **D**:

$$S^{\mathbf{D}} = \sigma(W_2 \cdot \mu(W_1 \cdot h_D + b_1) + b_2), \qquad (1)$$

where $W_{1,2}$ and $b_{1,2}$ are training parameters; σ/μ is the sigmoid/tanh function, respectively. We denote $\mathbf{D}_i$ ($i \in \{1, 2, 3, 4\}$) for each simile level in Figure 1. Then we use a separation loss and a compactness loss to train the MLSR model.

The separation loss separates different simile levels by distinguishing their quality scores. For each simile level i, we first calculate a centroid score $S^{\mathbf{D}_i} = \frac{1}{K_i} \sum_{k=1}^{K_i} S^{\mathbf{D}_i^k}$ where $S^{\mathbf{D}_i^k}$ is the quality score for this level, K_i is the example number of $\mathbf{D}_i$ in a training batch. The separation loss between different simile levels is:

$$l^{sep} = \sum_{j=1}^{3} \sum_{l=j+1}^{4} \max(0, \omega^{\star}\lambda + S^{\mathbf{D}_j} - S^{\mathbf{D}_l}), \qquad (2)$$

where λ=0.333 is the lower bound for the distance between two centroid scores, $\omega = l - j$ is the weight used for amplifying the lower bound according to the simile quality gap.

The compactness loss compacts the examples within the same simile level, which served as a regularization role to avoid outlier exceptions for each level. Specifically, the simile quality score $S^{\mathbf{D}_i^k}$ for $k \in \{1, 2, ..., K_i\}$ is forced to be closer to the corresponding centroid $S^{\mathbf{D}_i}$ as follows:

$$l^{com} = \sum_{i=1}^{4} \sum_{k=1}^{K_i} \max(0, |S^{\mathbf{D}_i} - S^{\mathbf{D}_i^k}| - \gamma), \qquad (3)$$

where γ is the upper bound for the distance between the centroid of a certain replacing level and the score within this level[2]. γ allows the model to automatically adjust the quality score of each level within a certain range. The final loss for MLSR is:

$$L^{mlsr} = - \frac{1}{N} \sum^{N} (l^{seq} + l^{com}), \qquad (4)$$

where N is the number of training examples. After training, the MLSR model can assign a simile quality score for an input sequence. Next, we will introduce the SI/SG tasks and how we leverage the quality score given by MLSR to assist the SI/SG tasks in a single PLM.

3.2. Simile Interpretation (SI) and Simile Generation (SG) Training

As shown in Figure 2 (b), when using BERT as the back-bone, SI and SG tasks are the same as the masked language model task where the BERT learns to recover the masked property or vehicle (Song et al., 2021; He et al., 2022). When using other PLMs with decoder structure (Radford et al., 2019; Raffel et al., 2020) for SI/SG tasks, we give the PLMs a task instruction and the masked sequence[3] and ask the PLMs to generate the missing component. The loss is:

$$\mathcal{L}^{si/sg} = - \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{r} (logP(R_i^t)), \qquad (5)$$

where R_i^t is the t-th word of the masked property/vehicle (total r words, $r \geq 1$).

3.3. Training and Testing

We train a single PLM with all three simile tasks, i.e., MLSR, SI, and SG. The loss is $L^{mlsr} + \mathcal{L}^{si/sg}$. During SI/SG testing, following previous work (Song et al., 2021; He et al., 2022), the PLMs choose the missing property or vehicle from a candidate pool. Two examples are shown in Table 3. In SI, the masked word is the shared property. In SG, the masked word is the vehicle.

As shown in Figure 2 (c), when using BERT for experiments, we use the masked-word-prediction heads of BERT to compute the probability for each candidate. The candidate with the highest probability will be chosen as the final choice. When using other large PLMs with decoder structures (Raffel

[2] For example, if γ=0.1, the expecting quality score ranges are $[0.9, 1]$, $[0.567, 0.767]$, $[0.233, 0.433]$, and $[0, 0.1]$.

[3] For example, the input to T5 is "i run as () as a rabbit. [choice] fast, wrong, bad, slow".

Task	Example and Candidates
SI	My client is as [MASK] as a newborn lamb. **A**. innocent. **B**. delicious. **C**. legal. **D**. guilty.
SG	The participant swims like a [MASK]. **A**. plait. **B**. dolphin. **C**. depiction. **D**. pod.

Table 3: Test examples for SI/SG tasks, the correct answer is innocent/dolphin, respectively. During the test, the model needs to select one answer from the candidates.

et al., 2020) for SI/SG tasks, we give an instruction and ask the PLMs to generate the missing component. The probability for each candidate is the generation probability.

To leverage the MLSR results for self-verification, we concatenate each of the candidates to the input sequence (to the masked position) as a new input. Then the model can calculate the MLSR score for this candidate. The final score S_{final} for each candidate is the weighted sum of the MLSR score S_{mlsr} and SI/SG score $S_{si/sg}$:

$$S_{final}\text{= } \beta\text{*}S_{mlsr} + (1 \text{ - } \beta)\text{*}S_{si/sg}, \qquad (6)$$

where β is a hyper-parameter.

4. Experimental Setup

4.1. Datasets

The multi-choice probing dataset (MCP) (He et al., 2022) is proposed for the SI task. There are two test sets named General Corpus and Quizzes. The multilingual simile dialogue dataset (MSD) (Ma et al., 2023) is proposed for studying similes in dialogue. We only use the English data in MSD. The SI/SG test sets are both multi-choice tasks and both have 450 examples. MCP and MSD both annotate all simile components. Hence, they are suitable for constructing multi-level simile data. The difference between MCP and MSD: 1) similes in MSD exist in 3-turn dialogues and the dialogue length is much longer than the sentence in MCP; 2) tenors and vehicles in MSD contain multiple formats such as verbal phrases and sentences. Table 4 shows the statistics of the datasets. For the MLSR task, we construct an equal number of examples (equal to the training set) for each simile level and randomly split all data into 9:1 as training/validation sets.

Dataset	Train / Dev / Test	Len.	Format
MCP	4,510 / - / 775+858	12.2	sentence
MSD	3126 (4570) / - / 450	40.1	dialogue

Table 4: Statistics of datasets. "Len." means average length per example. 775+858 are numbers of General Corpus + Quizzes. 3576 (4570) are numbers of similes (literals). Notice that the training set of MSD was originally proposed for simile recognition tasks. There are a total of 3576 examples. We remove the 450 examples that appear in the MSD SI/SG test sets, and only use the rest 3126 examples for MLSR and SR training.

4.2. Baselines

The baselines include: 1) **BERT-base**, **BERT-large** and **T5-3B** (Raffel et al., 2020). They are not fine-tuned with any simile data; 2) **BERT-ANT** (Chen et al., 2022) is trained with masked word prediction with a number of metaphor data. It is based on a BERT-large model and can solve the SI/ SG tasks in a unified framework of simile triple completion. For example, when giving tenor=athlete and vehicle=cheetah, BERT-ANT can generate a list of words including the shared property like "fast" or "agile". When performing our SI/SG tasks, we match the candidates of each example with the output list of BERT-ANT. An example is counted correct if the ground truth answer is listed before the other three distractors; 3) **BERT-large(mlm)** (He et al., 2022) is based on BERT-large that fine-tuned with masked-language model (MLM) training on MCP data; 4) **BERT-large(mlm+ke)** (He et al., 2022) is based on BERT-large that fine-tuned with both MLM training and knowledge embedding (KE) method (Bordes et al., 2013) on MCP data. The BERT-large(mlm) and BERT-large(mlm+ke) were originally trained for the SI task on MCP. We further fine-tuned them for SI/SG tasks on the MSD training set and reported their results on the test sets.

Besides applying our method on BERT-base, BERT-large, and T5-3B, we also provide different settings for our models. (- MLSR) means we remove the multi-level simile recognition training in the unified training process. Similarly, (- SI) and (- SG) mean we remove the SI and SG training, respectively. (- MLSR, +SR) means we replace MLSR training with traditional simile recognition (SR) training, where the model is trained to distinguish simile and literal. Notice that the MSD provides SR training data while the MCP does not. To train (- MLSR, +SR) on MCP, we obtain the literal data of MCP by replacing the vehicle with the synonym of the tenor as the second simile level in Figure 1. We consider the replacing results literal because they not only have the comparator words but also have a valid meaning that is not against the commonsense knowledge.

4.3. Evaluation Metrics

Following previous SI/SG work (Chen et al., 2022; He et al., 2022), we use Hit@1 for measuring multi-

Model	MCP					MSD				
	level-1	level-2	level-3	level-4	total	level-1	level-2	level-3	level-4	total
random	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500
BERT-base	0.7598	0.6797	0.6779	0.8203	0.7344	0.5723	0.5345	0.5389	0.6188	0.5661
(Joint Train)	0.7535	0.6755	0.6746	0.8200	0.7309	0.5523	0.5307	0.5344	0.5994	0.5524
BERT-large	0.7712	0.7025	0.7018	0.8317	0.7518	0.6123	0.5845	0.5889	0.6588	0.6111
(Joint Train)	0.7723	0.7028	0.7114	0.8310	0.7543	0.6137	0.5856	0.5910	0.6547	0.6112
T5-3B	0.8594	**0.7740**	0.7722	0.8790	**0.8212**	**0.6505**	0.6302	0.6314	0.6990	0.6528
(Joint Train)	**0.8598**	0.7675	**0.7731**	**0.8791**	0.8199	0.6498	**0.6311**	**0.6325**	**0.7005**	**0.6534**

Table 5: Multi-level simile recognition results. "(Joint Train)" means after joint training.

choice accuracy. For MLSR, we report accuracy for each simile level.

4.4. Implementation Details

The pre-trained models are all based on the public Pytorch implementation[4]. We use a single Tesla v100s GPU with 32GB memory for experiments. The batch size is all set to 24. The model is optimized using the Adam optimizer with a learning rate of 5e-6. The learning rate is scheduled by a warm-up and linear decay. The gradient clipping threshold is set as 10.0. During MLSR training, γ is set to 0.1 and early stopping on the corresponding validation data is adopted as a regularization strategy. The training epochs are 5.

5. Results and Analysis

We aim to answer the following questions: **1)** Can MLSR training help to assign a reasonable score for different simile levels? (section 5.1) **2)** Does the join training reduce the MLSR accuracy? (section 5.1) **3)** Does our method outperform the state-of-the-art models when exploring simile knowledge from PLMs? (section 5.2.1) **4)** Is our method useful for PLMs in different sizes or structures? (section 5.2.1) **5)** Where do the gains come from? (section 5.2.2) **6)** What is the effect when replacing MLSR with SR in our method? (section 5.2.2) **7)** How are the hyper-parameters decided? (section 5.3)

5.1. Multi-level Simile Recognition

As introduced in Section 4.1, we construct the multi-level simile data with the MCP/MSD training sets and split the data into 9:1 for MLSR training/validation. In Table 5, we report the performance of different PLMs on the validation set. This experiment tests whether the PLMs assign reasonable scores for different simile levels. Firstly, the PLMs have a much better performance than random results. It means the PLMs learn how to distinguish different simile levels through the MLSR training. Level-2 and level-3 have lower accuracy than level-1 and level-4. This is reasonable since level-2 and level-3 only corrupt parts of the simile relations as in Figure 1. They are harder to distinguish than the correct ones (level-1) or the fully corrupt ones (level-4). The performance has consistent performance across different PLMs and the performance increases when the model size becomes larger. BERT-large is better than BERT-base, and T5-3B is more accurate than BERT-large. The performance is very close before or after the joint training, which means the PLMs can maintain the MLSR capability in the joint training. BERT-large and T5-3B even have a small improvement in MCP and MSD after the joint training, respectively.

Model	SI(MCP)	SI(MSD)	SG(MSD)
BERT-base	0.6948	0.5333	0.2800
BERT-large	0.7755	0.5600	0.2977
BERT-ANT	0.7620	0.4622	0.3333
BERT-large(mlm)	0.7924	0.6867	0.5511
BERT-large(mlm+ke)	0.8005	0.6889	0.5556
BERT-base(ours)	0.8201*	0.7233*	0.5778*
BERT-large(ours)	**0.8464***	**0.7467***	**0.6155***
(- MLSR)	0.8145*	0.6978*	0.5578*
(- SI)	0.7876*	0.6476*	0.6000*
(- SG)	0.8332*	0.7413*	0.5689*
(- MLSR, +SR)	0.8089*	0.7069*	0.5578*
T5-3B	0.5754	0.6733	0.3111
T5-3B(ours)	**0.8158***	**0.8111***	**0.7756***
(- MLSR)	0.7012*	0.7222*	0.6711*
(- SI)	0.7043*	0.7111*	0.7178*
(- SG)	0.7861*	0.7778*	0.6689*
(- MLSR, +SR)	0.6916*	0.7244*	0.6733*

Table 6: SI and SG results (Hit@1) on MCP and MSD. The BERT-large(mlm) is the base model to do the significant test for our models (* means statistically significant with p<0.01). There are two sub-sets in MCP test data, we report the results of each sub-set in section 7.

To sum up, the MLSR training helps the PLMs to assign a reasonable quality score for different simile levels we defined and the performance does not drop after the joint training.

[4] https://github.com/huggingface/transformers

5.2. Simile Interpretation and Generation

Table 6 shows the SI (both MCP and MSD) and SG (MSD) results. We categorize the model into two groups: BERT-based and T5-based models.

5.2.1. Comparing with Baselines

In the first group, BERT-base and BERT-large are not fine-tuned. BERT-large outperforms BERT-base, especially on the MCP dataset. BERT-ANT is based on BERT-large and trained with a large corpus through Adjective-Noun mask Training. BERT-ANT outperforms the BERT-large on SG tasks. However, its performance on SI tasks is worse than BERT-large. This is because the training data of BERT-ANT is metaphor data, a large portion of the metaphor data does not have shared properties. Hence, BERT-ANT is better at predicting vehicles but worse at predicting shared properties. The BERT-large(mlm) is also based on a BERT-large model and fine-tuned with the corresponding training sets. Benefiting from the fine-tuning, BERT-large(mlm) largely outperforms all previous baselines on both SI/SG tasks. BERT-large(mlm+ke) is better than BERT-large(mlm) because the knowledge embedding method provides additional semantic features to align the simile components.

On the other hand, our self-verified method helps the BERT-base and BERT-large model to yield higher performance improvements. The BERT-base (ours) surpasses the strong BERT-large (mlm+ke) on both SI/SG. Our method also helps the BERT-large to obtain the highest performance among all models. BERT-large (ours) outperforms BERT-large (mlm+ke) by 4.59%/5.78%/5.99% on SI(MCP)/SI(MSD)/SG(MSD), respectively. The MCP SI test set has a much higher accuracy than the MSD SI test set, showing the simile in dialogue (MSD) task is more difficult than the simile in a single sentence (MCP). In the second group, a similar trend to the first group is observed on T5-3B. When using our self-verified method, T5-3B has consistent improvements on all three test sets. The results show that our method explores more accurate simile knowledge from PLMs.

5.2.2. Ablation Study on SI/SG

We also report the ablation study in Table 6. We can see that on both the MCP test set and MSD test sets, removing the training component of our model will cause declines.

For the BERT-large(ours), we can see that (-MLSR) causes 4.19%/6.00%/5.77% declines on SI(MCP)/SI(MSD)/SG(MSD), respectively. The results show that the MLSR score helps BERT to generate more accurate simile knowledge.

γ value	β is fixed with 0.8				
	0.0	0.05	0.1	0.15	0.20
SI results	75.01	81.17	**84.64**	84.43	84.31
γ value	β is fixed with 0.9				
	0.0	0.05	0.1	0.15	0.20
SI results	76.23	81.20	84.32	84.30	84.25
β value	γ is fixed with 0.1				
	0.1	0.2	0.3	0.4	0.5
SI results	84.15	84.17	84.17	84.27	84.32
β value	γ is fixed with 0.1				
	0.6	0.7	0.8	0.9	1
SI results	84.45	84.54	**84.64**	84.60	84.46

Table 7: The SI results (Hit@1) on MCP dataset with different values of γ/β. The back-bone model is BERT-large.

On the MCP test set, (- SI) causes 6.88% declines. On the MSD SI test set, (- SI) has an 11.02% decline. The results are reasonable since the MSD test set is a more difficult scenario which is harder for the PLM so it is more sensitive to the fine-tuning process. A similar trend can be observed with the (-MLSR) model, where SI(MSD) has more declines than SI(MCP). Meanwhile, (- SI) causes 1.55% declines in SG(MSD), which is much smaller than the declines in SI(MCP) and SI(MSD). It is also reasonable since SI and SG focus on different simile components. A similar trend happens with (-SG), which causes 4.66% declines on SG(MSD), but only entails 2.32%/1.65% declines on SI(MCP)/SI(MSD), respectively. However, removing SI does entail a performance drop on the SG task, which indicates the PLMs can leverage the simile knowledge learned from SI training to help the SG task, and vice versa.

On all test sets, (- MLSR, +SR) is very close to (-MLSR). The results indicate that the SR score contributes little to SI/SG tasks. This is because the SR score can only reflect whether an input contains a simile, it could not reflect the quality of the simile as the MLSR. On the other hand, whether SI/SG training can contribute to the SR task is beyond the scope of this paper.

In the second group, a similar trend to the first group is observed on T5-3B. When removing the training process, T5-3B has consistent declines on all three test sets.

5.2.3. Sum up of the SI/SG experiments

To sum up, experimental results show that 1) our self-verified method can explore more accurate simile knowledge from PLMs and largely increase the performance on SI/SG tasks; 2) our method can be applied to PLMs with different sizes and structures; 3) each fine-tuning task contributes to the performance; 4) the SR training is not as useful as the MLSR training.

Model	General Corpus	Quizzes
BERT-base	0.6413	0.7436
BERT-large	0.7239	0.8322
BERT-ANT	0.7410	0.8364
BERT-large(mlm)	0.7385	0.8458
BERT-large(mlm+ke)	0.7407	0.8594
BERT-base(ours)	0.7554*	0.8473*
BERT-large(ours)	**0.7962***	**0.8898***
T5-3B	0.5991	0.5432
T5-3B(ours)	**0.7443***	**0.8862***

Table 8: SI and SG results (Hit@1) on MCP. The BERT-large(mlm) is the base model to do the significant test for our models (* means statistically significant with p<0.01).

5.3. Hyper-parameters

Hyper-parameter γ in Section 3.1.2 and β in Section 3.3 are decided by experiments. For γ/β, we test with steps 0.05/0.1, respectively. Results on the SI(MCP) task are shown in Table 7. We finally choose γ=0.1 and β=0.8 according to the results.

6. Case Study

We randomly choose one SI example in the MSD test set for the case study. The dialogue is: "*Speaker A: "What's a good tip you hope you never have to use ?" Speaker B: "four pounds of weight can pull off a human ear, your pinky finger is as [MASK] as a baby carrot."*" The four choices are: "easy to bite or snap off, painless to poison or noise smoke, difficult to grip or thumb irritation, uneasy to champ or sound on". This is a difficult example since the shared property has six words. The T5-3B model in Table 6 assign a higher probability for "difficult to grip or thumb irritation". The wrong choice may be caused by the semantic connection between finger and thumb. After using our method, T5-3B model choose the correct answer "easy to bite or snap off". It benefits from the MLSR score which considers "your pinky finger is as easy to bite or snap off as a baby carrot" as a first level simile and assigns a higher score for "easy to bite or snap off".

7. More Experimental Results

There are two test sets in the MCP dataset, General Corpus and Quizzes. In Table 6, we combine them as a single test set for saving space. In Table 8, we report the experimental results on each of them separately. The trends are similar to Table 6.

8. Conclusion

This paper proposes a self-verified method to explore simile knowledge from PLMs. A multi-level simile recognition (MLSR) task is designed to train the PLMs to evaluate simile qualities. The MLSR score aligns simile recognition knowledge with simile generation knowledge and is used to assist in SI/SG tasks. Experimental results show that our method **1)** can help explore more accurate simile knowledge from PLMs; **2)** can be used on different kinds of PLMs and more large-scale PLMs. Future works include but are not limited to **1)** designing a method to automatically learn the quality scores at different levels; **2)** testing the self-verified method on more languages and larger-size PLMs.

9. Acknowledgements

This work is supported by the National Key Research and Development Program (No. 2022YFF0902100) and National Natural Science Foundation of China (No. 62076081 and No. 61936010).

9.1. Ethical considerations and limitations

The method proposed by this paper is only verified in languages with limited morphology (English). The effectiveness of the method on other morphology needs further verification. We test our method on BERT-base, BERT-large, and T5-3B and verify the effectiveness. The datasets and models we used are all released by previous work and all publicly available. We consider that there are no ethical problems with data or models. However, when using our method for simile interpretation or generation tasks, some unpleasant similes may be produced by the pre-trained language models due to the knowledge stored in the parameters.

10. Bibliographical References

Ehsan Aghazadeh, Mohsen Fayyaz, and Yadollah Yaghoobzadeh. 2022. Metaphors in pre-trained language models: Probing and generalization across datasets and languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2037–2050. Association for Computational Linguistics.

Naveen Badathala, Abisek Rajakumar Kalarani, Tejpalsingh Siledar, and Pushpak Bhattacharyya. 2023. A match made in heaven: A multi-task framework for hyperbole and metaphor detection. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 388–401. Association for Computational Linguistics.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics.

Yuri Bizzoni and Shalom Lappin. 2018. Predicting human metaphor paraphrase judgments with deep neural networks. In *Proceedings of the Workshop on Figurative Language Processing, Fig-Lang@NAACL-HLT 2018, New Orleans, Louisiana, 6 June 2018*, pages 45–55. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4762–4779. Association for Computational Linguistics.

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 2020, Online, November 16-20, 2020*, pages 6455–6469. Association for Computational Linguistics.

Weijie Chen, Yongzhu Chang, Rongsheng Zhang, Jiashu Pu, Guandan Chen, Le Zhang, Yadong Xi, Yijiang Chen, and Chang Su. 2022. Probing simile knowledge from pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5875–5887. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *ICLR*. OpenReview.net.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Kory W. Mathewson, and Osmar R. Zaïane. 2018. Augmenting neural response generation with context-aware topical attention. *CoRR*, abs/1811.01063.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar R. Zaïane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5271–5285. Association for Computational Linguistics.

Huawen Feng and Qianli Ma. 2022. It's better to teach fishing than giving a fish: An auto-augmented structure-aware generative model for metaphor detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 656–667. Association for Computational Linguistics.

Dedre Gentner, Brian Bowdle, Phillip Wolff, and Consuelo Boronat. 2001. Metaphor is like analogy. In *The analogical mind: Perspectives from cognitive science*, pages 199–253. MIT Press.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 55–64. ACM.

Qianyu He and Sijie Cheng and Zhixu Li and Rui Xie and Yanghua Xiao. 2022. *Can Pre-trained Language Models Interpret Similes as Smart as Human?* Association for Computational Linguistics.

Qianyu He, Xintao Wang, Jiaqing Liang, and Yanghua Xiao. 2023. MAPS-KB: A million-scale probabilistic simile knowledge base. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 6398–6406. AAAI Press.

Jintaro Jimi and Kazutaka Shimada. 2022. Pseudo data acquisition using machine translation and simile identification. In *12th International Congress on Advanced Applied Informatics, IIAI-AAI 2022, Kanazawa, Japan, July 2-8, 2022*, pages 391–396. IEEE.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *WMT@ACL*, pages 228–231. Association for Computational Linguistics.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Bin Li, Haibo Kuang, Yingjie Zhang, Jiajun Chen, and Xuri Tang. 2012. Using similes to extract basic sentiments across languages. In *Web Information Systems and Mining - International Conference, WISM 2012, Chengdu, China, October 26-28, 2012. Proceedings*, volume 7529 of *Lecture Notes in Computer Science*, pages 536–542. Springer.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*, pages 110–119. The Association for Computational Linguistics.

Yucheng Li, Chenghua Lin, and Frank Guerin. 2022. Cm-gen: A neural framework for chinese metaphor generation with explicit context modelling. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 6468–6479. International Committee on Computational Linguistics.

Yucheng Li, Shun Wang, Chenghua Lin, and Frank Guerin. 2023a. Metaphor detection via explicit basic meanings modelling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 91–100. Association for Computational Linguistics.

Yucheng Li, Shun Wang, Chenghua Lin, Frank Guerin, and Loïc Barrault. 2023b. Framebert: Conceptual metaphor detection with frame embedding learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 1550–1555. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Lizhen Liu, Xiao Hu, Wei Song, Ruiji Fu, Ting Liu, and Guoping Hu. 2018. Neural multitask learning for simile recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1543–1553. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6723–6737. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pre-training approach. *CoRR*, abs/1907.11692.

Longxuan Ma and Weinan Zhang and Shuhan Zhou and Churui Sun and Changxin Ke and Ting Liu. 2023. *I run as fast as a rabbit, can you? A Multilingual Simile Dialogues Datasets*. Association for Computational Linguistics.

Longxuan Ma, Ziyu Zhuang, Weinan Zhang, Mingda Li, and Ting Liu. 2022. Self-eval: Self-supervised fine-grained dialogue evaluation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 485–495. International Committee on Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and

David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics.

Rowan Hall Maudslay and Simone Teufel. 2022. Metaphorical polysemy detection: Conventional metaphor meets word sense disambiguation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 65–77. International Committee on Computational Linguistics.

Xu Ming. 2021. dialogbot: Dialogue model technology tool.

Xu Ming. 2022. textgen: Implementation of text generation models.

Saif M. Mohammad, Ekaterina Shutova, and Peter D. Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, *SEM@ACL 2016, Berlin, Germany, 11-12 August 2016*. The *SEM 2016 Organizing Committee.

Michael Mohler, Mary Brunson, Bryan Rink, and Marc T. Tomlinson. 2016. Introducing the LCC metaphor datasets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Vlad Niculae and Cristian Danescu-Niculescu-Mizil. 2014. Brighter than gold: Figurative language in user generated comparisons. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 2008–2018. ACL.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Anthony M Paul. 1970. Figurative language. In *Philosophy & Rhetoric*, page 225–248.

Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. GODEL: large-scale pre-training for goal-directed dialog. *CoRR*, abs/2206.11309.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Hongjin Qian, Xiaohe Li, Hanxun Zhong, Yu Guo, Yueyuan Ma, Yutao Zhu, Zhanliang Liu, Zhicheng Dou, and Ji-Rong Wen. 2021. Pchatbot: A large-scale dataset for personalized chatbot. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2470–2477. ACM.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

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.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational*

Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021, pages 3784–3803. 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.

Gerard Steen. 2010. A method for linguistic metaphor identification: From mip to mipvu. volume 14. John Benjamins Publishing.

Kevin Stowe, Nils Beck, and Iryna Gurevych. 2021. Exploring metaphoric paraphrase generation. In *Proceedings of the 25th Conference on Computational Natural Language Learning, CoNLL 2021, Online, November 10-11, 2021*, pages 323–336. Association for Computational Linguistics.

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

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 248–258. The Association for Computer Linguistics.

Lennart Wachowiak and Dagmar Gromann. 2023. Does GPT-3 grasp metaphors? identifying metaphor mappings with generative language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1018–1032. Association for Computational Linguistics.

Shun Wang, Yucheng Li, Chenghua Lin, Loïc Barrault, and Frank Guerin. 2023. Metaphor detection with effective context denoising. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 1396–1401. Association for Computational Linguistics.

Xiaoyue Wang, Linfeng Song, Xin Liu, Chulun Zhou, Hualin Zeng, and Jinsong Su. 2022. Getting the most out of simile recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3243–3252. Association for Computational Linguistics.

Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Yong Jiang, Xiaoyan Zhu, and Minlie Huang. 2020. A large-scale chinese short-text conversation dataset. In *Natural Language Processing and Chinese Computing - 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, October 14-18, 2020, Proceedings, Part I*, volume 12430 of *Lecture Notes in Computer Science*, pages 91–103. Springer.

Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2018. YEDDA: A lightweight collaborative text span annotation tool. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 31–36. Association for Computational Linguistics.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Xiangpeng Wei, Zhengyuan Liu, and Jun Xie. 2023. Fantastic expressions and where to find them: Chinese simile generation with multiple constraints. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 468–486. Association for Computational Linguistics.

Zheng Ye, Liucun Lu, Lishan Huang, Liang Lin, and Xiaodan Liang. 2021. Towards quantifiable dialogue coherence evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2718–2729. Association for Computational Linguistics.

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.

Shenglong Zhang and Ying Liu. 2023. Adversarial multi-task learning for end-to-end metaphor detection. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1483–1497. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT

: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.

Danning Zheng, Ruihua Song, Tianran Hu, Hao Fu, and Jin Zhou. 2019. "love is as complex as math": Metaphor generation system for social chatbot. In *Chinese Lexical Semantics - 20th Workshop, CLSW 2019, Beijing, China, June 28-30, 2019, Revised Selected Papers*, volume 11831 of *Lecture Notes in Computer Science*, pages 337–347. Springer.

11. Language Resource References

Robyn Speer and Joshua Chin and Catherine Havasi. 2017. *ConceptNet 5.5: An Open Multilingual Graph of General Knowledge*. AAAI Press. PID https://doi.org/10.1609/aaai.v31i1.11164.