

MAPS-KB: A Million-Scale Probabilistic Simile Knowledge Base

Qianyu He¹, Xintao Wang¹, Jiaqing Liang^{2*}, Yanghua Xiao^{1,3*}

¹Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University

²School of Data Science, Fudan University

³Fudan-Aishu Cognitive Intelligence Joint Research Center, Shanghai, China

{qyhe21, xtwang21}@m.fudan.edu.cn, liangjiaqing@fudan.edu.cn, shawyh@fudan.edu.cn

Abstract

The ability to understand and generate similes is an imperative step to realize human-level AI. However, there is still a considerable gap between machine intelligence and human cognition in similes, since deep models based on statistical distribution tend to favour high-frequency similes. Hence, a large-scale symbolic knowledge base of similes is required, as it contributes to the modeling of diverse yet unpopular similes while facilitating additional evaluation and reasoning. To bridge the gap, we propose a novel framework for large-scale simile knowledge base construction, as well as two probabilistic metrics which enable an improved understanding of simile phenomena in natural language. Overall, we construct MAPS-KB, a million-scale probabilistic simile knowledge base, covering 4.3 million triplets over 0.4 million terms from 70 GB corpora. We conduct sufficient experiments to justify the effectiveness of methods of our framework. We also apply MAPS-KB on three downstream tasks to achieve state-of-the-art performance, further demonstrating the value of MAPS-KB. Resources of MAPS-KB are publicly available at <https://github.com/Abbey4799/MAPS-KB>.

Introduction

As a figurative language, metaphors allow people to understand abstract concepts through concrete and familiar ones (Lakoff 1993; Lakoff and Johnson 2008). Metaphor understanding has become one of the most fundamental tasks to evaluate cognition intelligence (Liu et al. 2022; Chen et al. 2022). A simile is a special type of metaphor, which compares two fundamentally different terms via shared properties (Paul 1970; Tversky 1977). For example, in simile “*Her hair felt like silk.*”, “*hair*” (a.k.a, topic) is compared with “*silk*” (a.k.a., vehicle) with the implicit property “*soft*”, where *topic*, *vehicle* and *property* are three main components of similes (Hanks 2013). Since similes can make the literal expression more imaginative and graspable, they have been widely used in literature (Fishelov 2007) and daily conversations (Niculae and Danescu-Niculescu-Mizil 2014).

Nowadays, large pre-trained language models (PLMs) have become an important foundation of human-level machine intelligence. PLMs have achieved state-of-the-art performance in various language processing tasks. However,

PLMs still lag behind human cognition. For example, while PLMs exhibit a certain ability to interpret similes after fine-tuning (Chakrabarty, Choi, and Shwartz 2022; He et al. 2022), they can hardly understand similes that require common-sense knowledge (Liu et al. 2022). Although PLMs can do polish writing by generating coherent similes (Zhang et al. 2021; Chakrabarty, Muresan, and Peng 2020), the generated similes are far less diverse than human-written ones, since statistical models tend to favor high-frequency similes.

A large-scale symbolic knowledge base of similes in general is needed to further improve the simile understanding ability of PLMs. Symbolic knowledge bases have been widely introduced to bridge the gap between machines and human intelligence (Liu et al. 2020). Firstly, PLMs and other statistical models themselves suffer from a lack of knowledge that rarely appears in the corpora. Besides, *explicit structured* knowledge is indispensable for many applications that require explanation and interpretability, while PLMs only contain *implicit statistical* knowledge that is hard to understand. Furthermore, a structured knowledge base can provide additional analytical conveniences compared with purely statistical models, such as taxonomy-based reasoning and path-based reasoning. Hence, a large-scale simile knowledge base can help machines understand the diverse yet rarely expressed similes.

Many efforts have been devoted to the construction of large-scale symbolic knowledge bases, (KBs, or knowledge graphs) such as entity-oriented KBs (Vrandečić and Krötzsch 2014), concept-oriented KBs (Wu et al. 2012), and word-oriented KBs (Miller 1998). Recently, there has also been an increasing number of simile knowledge resources (Roncero and de Almeida 2015; Liu et al. 2022; Li, Zhu, and Wang 2013). However, they are still unsatisfactory, concerning both their coverage and expressiveness. On one hand, most of them are only thousand-scale (Roncero and de Almeida 2015; Liu et al. 2022), while million-scale knowledge bases are needed in a wide range of downstream applications, such as online recommendation systems (Wang et al. 2019) and open-domain question answering (Cui et al. 2017). On the other hand, existing simile knowledge bases skip acquisition of *property* (Li, Zhu, and Wang 2013), an important component of similes, since *properties* are usually implicit and can not be extracted from texts directly, such as the implicit property “*soft*” in “*Her hair felt like silk.*”. How-

*Corresponding author.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

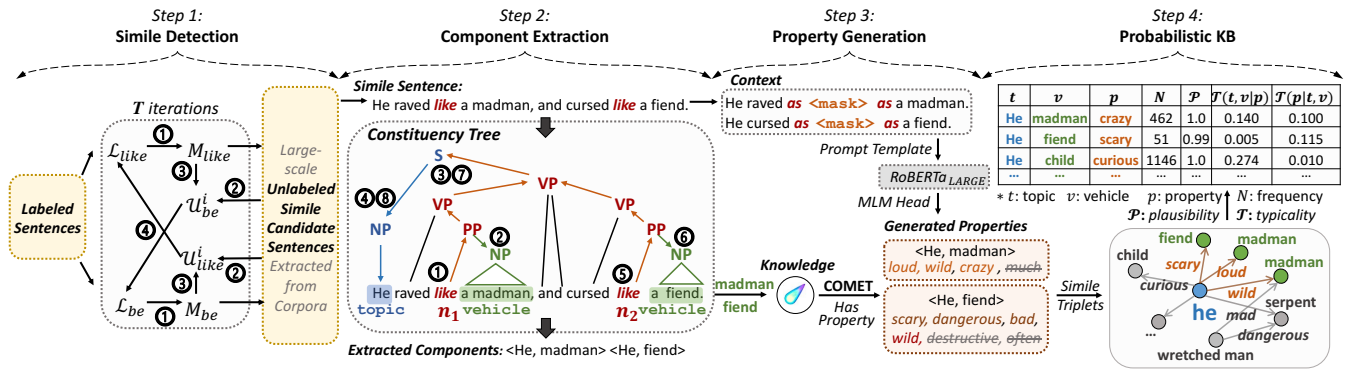


Figure 1: Our framework for large-scale simile knowledge base construction. In Step 1, with large-scale corpora and a small set of labeled sentences as input, we detect millions of simile sentences via *co-training*. In Step 2, given each simile sentence, we extract the important component *topic* and *vehicle* by designing rules based on constituency parsing. In Step 3, we generate the important component *property* considering both *context* information and external *knowledge* sources, and finally complete millions of simile triplets (t, p, v) . In Step 4, we design two probabilistic metrics (*plausibility* and *typicality*) for each simile triplet, which can facilitate simile-related inference.

ever, the shared *property* is the foundation for comparisons between the *topic* and *vehicle*, and is thus essential in downstream tasks (Chakrabarty, Muresan, and Peng 2020).

In this paper, we construct **MAPS-KB**, a Million-scAle Probabilistic Simile Knowledge Base. The construction framework is shown in Figure 1. In the first step, we extract simile candidate sentences via two syntactic patterns, then distinguish simile sentences from literal sentences through *co-training*, leveraging large-scale unlabeled sentences collected from both patterns. In the second step, we extract the simile components (*topic* and *vehicle*) by designing rules based on constituency parsing, addressing two challenges: *simile component positioning* with long distance dependencies and *simile component span identification*. In the third step, we generate the simile component *property* for each simile sentence, considering both surrounding *context* information and external *knowledge* sources, since properties can hardly be extracted from texts directly. Furthermore, we intentionally retain each simile triplet with its frequency information and design two probabilistic metrics for simile triplets in the fourth step, which allow machines to better understand similes in natural language and facilitate related applications in the real world.

To summarize, our contributions are mainly three-fold: (1) To the best of our knowledge, MAPS-KB is the first million-scale probabilistic simile knowledge base, covering 4.3 million triplets over 0.4 million terms from 70 GB corpora. The probabilistic metrics *plausibility* and *typicality* in MAPS-KB endow machines with an improved understanding of simile phenomena and facilitate a wide range of real applications. (2) We propose a novel framework for large-scale probabilistic simile knowledge base construction. This framework can be extended to other figurative languages like sarcasm and humor which can be represented by several major components. (3) We conduct extensive experiments to evaluate the effectiveness and necessity of methods in our framework, and further use MAPS-KB to achieve state-of-the-art performance in three popular simile-related tasks.

Related Work

Simile Knowledge Acquisition Early studies mainly acquire simile knowledge via linguists (Lakoff and Johnson 2008; Roncero and de Almeida 2015). However, the acquisition process is human-laborious and inefficient. With the rise of web documents, data-driven methods are developed to acquire simile knowledge automatically (Li, Zhu, and Wang 2013). (Li, Zhu, and Wang 2013) first collect simile sentences via syntactic patterns, and then retain the topic-vehicle pairs without is-a relations to construct a million-scale simile knowledge base. Recently, many studies probe simile knowledge from PLMs via designed prompt template (Chen et al. 2022; He et al. 2022). Different from these works, we automatically construct a million-scale probabilistic simile knowledge base in the form of simile triplets.

Simile Processing Tasks Earlier studies mainly focus on discriminating similes from literal sentences and extracting simile components from similes (Liu et al. 2018; Zeng et al. 2020). Recently, many studies transfer literal sentences to similes via finetuning PLMs (Chakrabarty, Muresan, and Peng 2020; Zhang et al. 2021). There is also a bulk of works designing specific tasks to investigate whether PLMs have the ability to interpret similes, such as simile property probing task (He et al. 2022), Winograd-style simile understanding (Liu et al. 2022) and simile continuation selection and generation (Chakrabarty, Choi, and Shwartz 2022). Different from these works, we perform simile processing tasks based on a large-scale probabilistic simile knowledge base.

MAPS-KB Construction

We start by collecting sentences from several corpora (Table 1). Then, we design two syntactic patterns to extract sentences. According to (Paul 1970), similes compare two different terms, typically using the comparator “like”, such as “Her hair felt like silk”. Additionally, after removing the comparator “like”, similes become implicit comparisons (Tversky 1977), such as “Her hair is silk.”. Considering these linguistic theories, our extraction patterns are de-

Name	Size	# Pattern _{like}	# Pattern _{be}
Openwebtext	38GB	1.5M	21M
Gutenberg	26GB	0.5M	8M
Bookcorpus	6GB	0.5M	2M
overall	70GB	2M	31M

Table 1: Statistics of used corpora and extracted simile candidates. G and M mean Gigabytes and millions.

defined as follows: (1) the *like* pattern: $Noun_1 \dots BE/VB \text{ like } \dots Noun_2$; (2) the *be* pattern: $Noun_1 \dots BE \dots Noun_2$. Finally, the extracted sentences are *simile candidates*, which can either be true similes or literal comparisons.

Simile Detection Next, we distinguish simile sentences from literal sentences. There are two syntactic patterns of data in the collected unlabeled sentences, namely the *like* pattern and the *be* pattern. Since expressions of two patterns for the same topic-vehicle pair are semantically consistent, we adopt *co-training* for simile detection. The sentences extracted by the two patterns are seen as instances from two views, namely *like* view and *be* view.

In a bootstrapping process of semi-supervised co-training, classifiers from multiple views are trained in a collaborative manner. Compared with the self-training procedure which only utilizes data from one view, the co-training procedure makes models from *like* view and *be* view complement each other. Compared with other supervised methods (Zeng et al. 2020), co-training is semi-supervised and effectively leverages massive unlabeled sentences.

Our detailed simile detection procedure with co-training is illustrated below. At each iteration i , two separate classifiers M_{like} and M_{be} are trained by the data from *like* view and *be* view respectively. Here, we use two separate BERT_{BASE} model as our classifiers. Then, M_{like} and M_{be} annotate the unlabeled subset of the other’s view U_{be}^i and U_{like}^i . Afterwards, the pseudo-labeled subsets U_{be}^i and U_{like}^i are integrated into the labeled set \mathcal{L}_{be} , \mathcal{L}_{like} . The process is performed iteratively, and there are T iterations in total. After co-training, the fully trained classifiers M_{like} and M_{be} are applied to U_{be} and U_{like} again respectively. Positively labeled samples in final round are denoted as \mathcal{L}'_{like} and \mathcal{L}'_{be} .

To improve the quality of pseudo-labels, only sentences with predicted confidence score greater than certain thresholds (θ_{like} and θ_{be}) would be labeled as simile sentences. Since there are much fewer positive samples than negative ones, pseudo-labeled negative samples are sampled to an equal number as positive ones to avoid data imbalance. A small set of data from *like* view are labeled to initialize the two classifiers at the first iteration.

Eventually, we extract 637,253 simile sentences from *like* view and 1,279,537 simile sentences from *be* view.

Components Extraction In this section, we extract the simile components *topic* t and *vehicle* v from simile sentences. There are two main challenges: *component positioning* and *component span identification*. Take the simile “*They were like kids in a candy store.*” as an example. Component positioning is to locate the object “*kid*” which the

topic “*They*” is compared to. Given the object “*kid*”, component span identification determines the optimal span for the semantics that the vehicle should carry. As a vehicle, “*kids in the candy store*” is more expressive than “*kids*” to represent how enthusiastic “*They*” were. Hence, the method not only needs to locate the components accurately, but also needs to detect suitable component spans.

To address these two challenges, we design rules based on *constituency parsing* of simile sentences. Previous studies (Liu et al. 2018; Zeng et al. 2020) ask annotators to label the training dataset and train the sequence labeling models. However, the models may perform worse for sentences with long-distance dependencies. Constituency parsing parses sentences into trees of sub-phrases. As illustrated in the second step in Figure 1, the leaf nodes are words, while the non-leaf nodes are the types of phrases in a constituency tree. The phrase-level parsing allows constituency parsing to address both *component positioning* with long distance dependencies and *component span identification*.

The pseudo-code of our component extraction is illustrated in Algorithm 1¹. First, we find all the leaf nodes n_i whose text is “*like*” as anchor nodes (line 10). This is because the subject and object of the comparator “*like*” are usually the *topic* and *vehicle*. According to our observation, the *vehicle* generally appears in the subtree rooted at n_i ’s sibling node (the green part) (line 12-13). Moreover, we iterate over the parent nodes of n_i from bottom to the top (the red part) until a node labeled as S or NP (line 15-20). The *topic* often appears in the subtree rooted at this node (the blue part). Second, we use the function GETCOMP for located subtrees to find the component. The function GETCOMP first performs pre-order traversal over the subtree, then returns the first node whose child nodes are all labeled by NP or PP (line 2-8). Even if a node is labeled by NP, its text may be a complex clause. Hence, ignoring the labels of child nodes may extract improper components. Finally, we further filter simile sentences based on extracted (topic, vehicle) pairs, due to the noise in the simile detection process. Overall, we collect 524,055 simile sentences with extracted topics and vehicles from 637,253 *like*-view sentences.

Property Generation A simile compares the topic t to the vehicle v via a shared property p (Paul 1970). Hence, revealing the shared properties is crucial for promoting machines’ ability to understand and generate similes. Previous works skip properties acquisition since properties are usually implicit and can hardly be extracted from texts directly.

We generate properties for each simile sentence from two perspectives: *knowledge* and *context*. Since *vehicle* is the most valuable component when inferring the shared properties (Tversky 1977), we utilize the *knowledge* from

¹The proposed method is based on similes from the “*like*” pattern. Extracting components from similes of the “*be*” pattern is left for future work. Because our main focus is to construct a million-scale simile KB and prioritize its quality. As for the coverage, (Niculae and Danescu-Niculescu-Mizil 2014) found that 82% similes contain the comparator “*like*” among the collected datasets. As for the precision, the “*like*” pattern similes can be better extracted with less noise, due to explicit structure.

Algorithm 1: Component Extraction Based on Constituency Parsing.

Input: Parsed constituency tree \mathcal{K} of sentence S .
Output: Extracted component topic c_t and vehicle c_v .

```
1 Function GETCOMP (Tree Node  $n$ ):  
2   while  $n \neq \text{NULL}$  do  
3      $\mathcal{L} = \text{label set of } n \rightarrow \text{children}$ .  
4     if  $\mathcal{L}$  are all in  $\{NP, PP\}$  and  $NP$  in  $\mathcal{L}$  then  
5       return  $n \rightarrow \text{text}$ .  
6     for  $c \in n \rightarrow \text{children}$  do  
7       if GETCOMP ( $c$ )  $\neq \text{NULL}$  then  
8         return GETCOMP ( $c$ ).  
9   return NULL.  
10  $\mathcal{T} = \{n_0, n_1, \dots, n_m\}$  is a subset of  $\mathcal{K}$ .  $n_i$  is the leaf node  
    where  $n \rightarrow \text{text} == \text{"like"}$ .  
11 for  $i = 0 \rightarrow m$  do  
12    $n = n_i \rightarrow \text{parent}$ .  
13    $c_v = \text{GETCOMP}(n)$ .  
14    $c_t = \text{NULL}$ .  
15   while  $n \neq \text{NULL}$  and  $c_t == \text{NULL}$  do  
16     if special rules for similes are triggered then  
17        $c_t = \text{GETCOMP}(\text{corresponding node})$ .  
18     else if  $n \rightarrow \text{label}$  in  $\{NP, S\}$  then  
19        $c_t = \text{GETCOMP}(n)$ .  
20    $n = n \rightarrow \text{parent}$ .  
21 return  $c_t, c_v$ .
```

vehicles to retrieve the properties. Specifically, given a vehicle, we generate properties via *HasProperty* relation from COMET (Bosselut et al. 2019). Besides, inferring the properties also requires an understanding of the *context* (He et al. 2022). Therefore, we rewrite collected simile sentence $s = (w_1, \dots, w_{i-1}, \text{like}, w_{i+1}, \dots, w_N)$ to $s' = (w_1, \dots, w_{i-1}, \text{as}, [\text{MASK}], \text{as}, w_{i+1}, \dots, w_N)$ (He et al. 2022). Then, we use RoBERTa_{LARGE} with pre-trained masked-word-prediction heads to generate properties.

We ensure the quality of the properties based on confidence scores. We design dynamically adjusted thresholds rather than fixed ones. For each simile sentence, we keep the top-10 predictions from each perspective, whose scores are normalized into $[0, 0.5]$. Only properties with a score greater than thresholds $\theta_{\text{knowledge}}, \theta_{\text{context}}$ are retained.

Overall, we obtain 5,511,111 *simile instances* in the form of (s, t, p, v) . From them, we extract 4,347,111 simile triplets. For each (s, t, p, v) , the final score is the sum of normalized scores predicted by COMET and RoBERTa_{LARGE}, which ranges from 0 to 1. For each simile triplet, we calculate *frequency* to present how many instances support it.

A Probabilistic Knowledge Base Previous works (Li, Zhu, and Wang 2013) abandon extracted similes with low frequency. Unpopular as the abandoned similes are, they may still be plausible and expressive. Hence, instead of removing the uncertain similes, we model simile triplets with probabilistic information, which contains two metrics, *plausibility* and *typicality* (Li, Zhu, and Wang 2013). *Plausibility* evaluates the quality of simile triplets based on scores of

their supporting instances, and *typicality* measures how well a property suits a topic-vehicle pair, and how typical it is to compare the topic to the vehicle given the property.

Plausibility. Each simile triplet (t, p, v) is supported by multiple (s_i, t, p, v) instances. The quality of simile triplets depends on confidence scores of corresponding instances. Hence, we adopt the *noisy-or* model to measure *plausibility* of simile triplets. In the *noisy-or* model, the plausibility of triplet (t, p, v) tends to be zero when confidence scores of all (s_i, t, p, v) instances are close to zero. Formally, the plausibility of triplet (t, p, v) is defined as follows:

$$\mathcal{P}(t, p, v) = 1 - \prod_{i=1}^{\eta} (1 - S(s_i, t, p, v)),$$

where $S(s_i, t, p, v) = P(p|s_i, t, v)$ is the confidence score of each (s_i, t, p, v) instance in the previous step and η is the number of instances supporting triplet (t, p, v) .

Typicality. Intuitively, *curious* is a more typical property of $(\text{he}, \text{child})$ compared with *good*, and people will first think of $(\text{he}, \text{child})$ rather than (he, cat) given the property *curious*. Hence, the *typicality* is an important measure for simile triplets, and can facilitate simile processing tasks.

We design two metrics to measure the *typicality* of simile triplets: $\mathcal{T}(p|t, v)$ for simile understanding and $\mathcal{T}(t, v|p)$ for simile generation, which are formulated as follows:

$$\mathcal{T}(p|t, v) = \frac{N(t, p, v) \cdot \mathcal{P}(t, p, v)}{\sum_{(t', p', v) \in \mathcal{G}_{(t, v)}} N(t', p', v) \cdot \mathcal{P}(t', p', v)},$$

$$\mathcal{T}(t, v|p) = \frac{N(t, p, v) \cdot \mathcal{P}(t, p, v)}{\sum_{(t', p, v') \in \mathcal{G}_p} N(t', p, v') \cdot \mathcal{P}(t', p, v')},$$

where \mathcal{G}_p or $\mathcal{G}_{(t, v)}$ denotes triplets containing p or (t, v) pair. $N(t, p, v)$ denotes the *frequency* of (t, p, v) , i.e., the number of supporting instances (s_i, t, p, v) . $\mathcal{P}(\cdot)$ are the plausibility.

MAPS-KB Statistics

In this section, we compare the statistics of MAPS-KB with existing simile-related resources, as shown in Table 2. To begin with, MAPS-KB is million-scale, while most of the others are hundred or thousand-scale. Although there is one million-scale knowledge base MetaKB (Li, Zhu, and Wang 2013), it lacks the important component *property*, which plays an imperative role in downstream tasks. Moreover, MAPS-KB is the only probabilistic simile knowledge base.

To further compare with the only existing million-scale knowledge base MetaKB, we show the frequency distribution of (t, v) pairs in Figure 2 (left). Although both MAPS-KB and MetaKB* exhibit long-tail distributions, the tail of MetaKB*'s distribution is much heavier than that of MAPS-KB's distribution. This indicates that MAPS-KB contains more reliable (t, v) pairs which appear multiple times.

According to the conceptual metaphor theory (Lakoff 1993), metaphor can be viewed as a mapping between terms in two domains, which reflects people's understanding of the world. Hence, we study the domain mapping patterns of similes by distinguishing topic and vehicle into ten domains via WordNet (Miller 1998) hypernyms. First, we manually select ten popular and general hypernyms as the domains that

Resource	Size	# <i>t</i>	# <i>v</i>	# <i>p</i>	<i>Prob</i>
Linguistics	679	65	75	379	N
SPP	1633	721	667	333	N
Fig-QA(M)	1458	441	1198	646	N
MetaKB	2.6M	-	-	-	N
MetaKB*	0.9M	383k	434k	-	N
MAPS-KB	4.3M	178k	227k	9k	Y

Table 2: The statistics of MAPS-KB and existing simile-related resources, the Linguistics (Roncero and de Almeida 2015), SPP (He et al. 2022), Fig-QA(M) (Liu et al. 2022) and MetaKB (Li, Zhu, and Wang 2013). MetaKB* denotes the published subset of MetaKB. Fig-QA(M) denotes the medium training set which reports the statistics. *Prob* indicates whether probabilistic information is contained.

(c_t, c_v) Pair	Example Simile Triplet	%
(artifact, artifact)	(road, decorative, ribbon)	10.61
(person, person)	(man, funny, fool)	10.05
(person, artifact)	(man, silent, statue)	06.24

Table 3: Examples and percentages of the top 3 domain mappings between topics and vehicles. For each domain mapping, % denotes the percentage of corresponding simile triplets among all triplets.

cover most of the terms: $\{person, animal, body\ part, food, natural\ object, natural\ phenomenon, feeling, artifact, location, action\}$. Then, to assign domains to terms, we select nouns in terms via NLTK². For a given noun, we traverse its synsets until finding a domain in the hypernym path of the synset, defined as the domain of the term. We show the examples and percentages of the top 3 domain mappings between topics and vehicles in Table 3.

Intrinsic Evaluation

In this section, we conduct experiments to evaluate the effectiveness and necessity of methods in our framework as well as the quality of simile triplets in MAPS-KB.

Simile Detection We evaluate the effectiveness of our simile detection method. We collect 1,197 sentences of *like* view from Wikipedia and ask two annotators to label them. As a result, 401 similes and 796 literal sentences are labeled. We also consider existing labeled dataset UGC (Niculae and Danescu-Niculescu-Mizil 2014), which contains 368 similes and 405 literal sentences of *like* view. The sentences are split into 8:1:1 as the train/dev/test splits.

We compare the performance of BERT_{BASE} classifier(s) in three settings: (1) Supervision: The model is trained entirely on labeled data. (2) Self-supervision: Besides labeled data, we also utilize unlabeled data from *like* view. (3) Co-training: We leverage unlabeled data from *like* view and *be* view, as is introduced in the previous section.

The results are shown in Table 4. The results of all our experiments are averaged over three random seeds. According

²<https://www.nltk.org/>

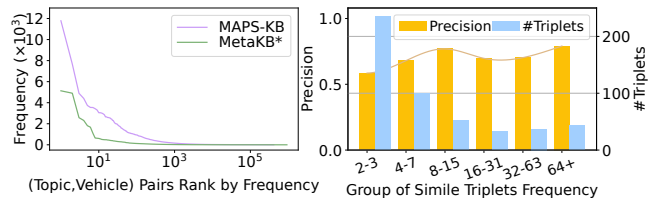


Figure 2: The frequency distribution of the (t, v) pairs in MAPS-KB and MetaKB* (left), and human evaluation results of similes in MAPS-KB grouped by frequency (right).

Dataset	Wikipedia			UGC		
	P	R	F1	P	R	F1
Supv.	71.18	78.33	74.58	81.73	77.16	79.50
Self-Supv.	73.38	91.67	81.49	83.20	78.33	80.66
Co.	74.50	90.00	81.52	84.67	77.50	80.88

Table 4: Results of different training methods in simile detection, with Supv., Self-Supv. and Co. representing Supervision, Self-supervision, and Co-training, respectively.

to the results, we find that co-training achieves the best precision and F1 score with comparable recall, which demonstrates that co-training can detect most of the simile sentences while ensuring better quality of the predicted ones. As we concern more about precision in this step, co-training is a better solution than other methods.

Component Extraction We also evaluate the quality of our component extraction method. Two annotators are asked to label the components of similes in the previous section. Annotated components should cover rich semantics as much as possible. For example, in “*He is like a kid in the candy store.*”, it is better for the vehicle to be “*kid in the candy store*” rather than simply “*kid*”. We finally get 401 similes with labeled components. Existing labeled datasets (Niculae and Danescu-Niculescu-Mizil 2014) are not concerned, since the labeled components are only one word.

We compare components extracted by the rules of MetaKB and ours. MetaKB designs rules solely based on regex, without considering syntactic structure. For evaluation, we consider two settings with different difficulties. In *hard* setting, a prediction is considered correct only if it matches the label exactly, while in *easy* setting, a prediction is simply expected to have the same core noun as the label. From the results in Table 5, we have the following analyses. First of all, our rules outperform those of MetaKB significantly under both settings. MetaKB rules are only comparable to ours for vehicles in the *easy* setting. This indicates that syntactic structure plays an imperative role in component extraction. Second, extracting topics is more difficult than extracting vehicles. This is because topics tend to have longer distance from the comparator “*like*” than vehicles. Additionally, with regards to vehicles, the performance gap between rules of ours and MetaKB is much more significant in the *hard* setting than *easy*. This indicates that though regex-based rules can locate vehicles correctly, it is hard for them to identify the integral vehicles with full semantics.

Eval	Comp	Method	P	R	F1
Hard	Topic	MetaKB	53.28	51.13	52.19
		MAPS-KB	64.50	59.95	62.14
	Vehicle	MetaKB	67.10	64.41	65.73
		MAPS-KB	81.79	77.69	79.69
Easy	Topic	MetaKB	71.39	68.51	69.92
		MAPS-KB	81.30	76.92	79.05
	Vehicle	MetaKB	92.43	88.72	90.54
		MAPS-KB	92.35	88.38	90.32

Table 5: Results of rules of MetaKB and ours in the component extraction task. *Hard* and *Easy* represent different levels of difficulty.

Human Evaluation

We conduct human evaluation to evaluate the precision of simile triplets in MAPS-KB. We randomly sample 500 triplets with frequency greater than one, and ask two annotators to label whether they are correct or not. The average precision is 0.71, and the Fleiss’ Kappa score (Fleiss 1971) is 0.75. We further study the precision of triplets grouped by frequency. According to the results shown in Figure 2 (right), we find that simile triplets with higher frequency tend to be more precise, which is a reasonable result.

Extrinsic Evaluation

In this section, we apply MAPS-KB knowledge to three downstream applications to demonstrate its effectiveness.

MAPS-KB for Simile Processing Tasks

There are mainly two simile processing tasks: simile interpretation (SI) and simile generation (SG). SI aims to infer shared properties for given (t, v) pairs, while SG aims to generate suitable vehicles for given (t, p) pairs.

Inference Rules We define scores $S_{(t,v)}(p)$ and $S_{(t,p)}(v)$ based on *typicality*, *frequency* and *plausibility* for the inference of SI and SG respectively. The candidate properties and vehicles are ranked by corresponding scores, and the ones with higher scores are regarded as possible answers. These scores are formulated as follows:

$$S_{(t,v)}(p) = \sum_{(t',p,v) \in \mathcal{G}_{(p,v)}} \mathcal{T}(p|t',v) \cdot N(t',p,v) \cdot \mathcal{P}(t',p,v),$$

$$S_{(t,p)}(v) = \sum_{(t',p,v) \in \mathcal{G}_{(p,v)}} \mathcal{T}(t',v|p) \cdot N(t',p,v) \cdot \mathcal{P}(t',p,v).$$

Specially, we ignore the original topic t and sum over all topics t' of $(t',p,v) \in \mathcal{G}_{(p,v)}$ that serve as topics of (p,v) in MAPS-KB. The reasons are twofold: (1) Theoretically, properties and vehicles are more salient to each other compared with topics (Tversky 1977; Veale and Hao 2007). (2) Practically, MAPS-KB cannot cover the probabilistic information of all (t,p,v) triplets in downstream datasets, which hinders inference. Considering both reasons, we adopt the more available statistic information of (p,v) .

For both tasks, the scores are sums of products of *typicality* \mathcal{T} , *frequency* N and *plausibility* \mathcal{P} , which empirically yield the best results compared with the alternatives that concern only one or two of them.

Experiment Setup We evaluate our inference rules on three benchmark datasets: Linguistics (Roncero and de Almeida 2015), Quizzes and General Corpus (He et al. 2022). Following (Chen et al. 2022), we keep the similes in Linguistics with frequency larger than 4. We use Mean Reciprocal Rank (**MRR**) and Recall@k (**R@k**) as metrics.

We compare our inference rules with the following baselines: (1) **Meta4meaning** (Xiao et al. 2016): They adopt statistical associations to select optimal properties. (2) **GEM** (Bar, Dershowitz, and Dankin 2020): They rank properties via their co-occurrence and similarity with topics and vehicles. (3) **ConScore** (Zheng et al. 2019): They rank candidates via their distance from other components in embedding space. (4) **LMASKB** (Chen et al. 2022): They probe simile knowledge from PLMs via designed prompt templates. Note that ConScore and LMASKB utilize the train split, while Meta4meaning, GEM and our method do not.

Results The results are shown in Table 6, from which we draw the following conclusions. First, MAPS-KB with our inference rules achieves the best performance on most of the metrics for both tasks. Even though our method does not utilize the train set, it surpasses supervised methods ConScore and LMASKB which do. This validates that MAPS-KB contains abundant informative simile knowledge that well supports downstream simile processing tasks. Besides, our method outperforms LMASKB which directly probes simile knowledge from PLMs. This reveals that our explicit structure knowledge is more useful than the implicit statistical knowledge contained in PLMs when processing similes.

MAPS-KB for Writing Polishment

We propose three methods to polish writing with MAPS-KB: a PLM-based method, a rule-based method, and their combination. The PLM-based method first injects simile knowledge in MAPS-KB into a sequence-to-sequence PLM BART (Lewis et al. 2020) via finetuning. The finetuning dataset is automatically constructed from MAPS-KB, where a collected simile instance (s, t, p, v) is transformed into a sample, whose output decoder target is the simile sentence s itself and the input encoder source is s rewritten to drop v but include p . For example, given $s = \text{“Her hair felt like silk.”}$ and $(t, p, v) = (\text{hair, soft, silk})$, the encoder source would be $\text{“Her hair is soft.”}$. Afterward, the PLM can be directly applied to downstream datasets without further training.

The rule-based method infers vehicles from probabilistic information of MAPS-KB, and then rewrites sentences with rules. Given the adjective/adverb property p in the input, the score $S'_p(v)$ for vehicle v is defined as follows:

$$S'_p(v) = \sum_{(t',p,v) \in \mathcal{G}_{(p,v)}} \mathcal{T}(t',v|p) \cdot N(t',p,v) \cdot \mathcal{P}(t',p,v) \cdot e^{\gamma \cdot l(v)}.$$

Compared with $S_p(v)$ in simile processing, $S'_p(v)$ has an additional term $e^{\gamma \cdot l(v)}$, where $l(v)$ is the number of words

Dataset		Linguistics					Quizzes				General Corpus			
Task	Method	TS	MRR	R@5	R@10	R@15	MRR	R@5	R@10	R@15	MRR	R@5	R@10	R@15
SI	Meta4meaning	N	N/A	0.221	0.303	0.339	0.153	0.222	0.276	0.314	0.069	0.094	0.156	0.195
	GEM	N	<u>0.312</u>	0.198	0.254	0.278	0.075	0.055	0.116	0.154	0.030	0.023	0.055	0.075
	ConScore	Y	0.078	0.076	0.138	0.172	0.044	0.048	0.112	0.167	0.030	0.029	0.052	0.084
	LMASKB	Y	0.270	0.378	<u>0.490</u>	<u>0.524</u>	<u>0.342</u>	<u>0.440</u>	<u>0.549</u>	<u>0.628</u>	<u>0.196</u>	<u>0.271</u>	<u>0.352</u>	<u>0.419</u>
MAPS-KB	N	0.392	<u>0.367</u>	0.493	0.544	0.375	0.520	0.699	0.749	0.200	0.318	0.439	0.493	
SG	ConScore	Y	0.036	0.055	0.090	0.103	0.002	0.002	0.007	0.007	0.001	0.001	0.003	0.004
	LMASKB	Y	<u>0.095</u>	<u>0.124</u>	<u>0.145</u>	<u>0.159</u>	<u>0.042</u>	<u>0.059</u>	<u>0.094</u>	<u>0.121</u>	<u>0.022</u>	<u>0.026</u>	<u>0.037</u>	<u>0.046</u>
	MAPS-KB	N	0.105	0.140	0.162	0.179	0.132	0.201	0.272	0.336	0.067	0.092	0.115	0.144

Table 6: Evaluation results of different methods on simile interpretation(SI) task and simile generation(SG) task. Bold numbers are the best results. The second best results are marked by “ ”. TS indicates whether the train set is used. On the Linguistics dataset, results of ConScore and LMASKB are taken from (Chen et al. 2022), other results are taken from their original papers.

in v and γ is a hyperparameter. By introducing $e^{\gamma \cdot l(v)}$, we encourage longer vehicles which tend to be more expressive. Then, we replace the adjective/adverb p in the input with a comparator “like” and the vehicle v of highest $S'_p(v)$ score.

Finally, we propose a combination of the PLM-based and rule-based methods. We find that PLM-generated similes may be incoherent if (1) they do not contain “like” or (2) their vehicles contain commas or are longer than seven words. The combined method outputs similes generated by rules in this case, and similes generated by PLMs otherwise.

Experiment Setup We evaluate our methods on the test set proposed by (Chakrabarty, Muresan, and Peng 2020). In this testset, the literal inputs always end with adjective/adverb, while the simile outputs end with “like” and vehicles. E.g., the literal input “Love is rare” is expected to be polished into the simile output “Love is like a unicorn”.

For evaluation, we only retain the generated vehicles after the word “like”, following (Chakrabarty, Muresan, and Peng 2020). We adopt three automatic evaluation metrics: (1) **BLEU** (Papineni et al. 2002), one of the most popular metrics for assessing generation tasks. (2) **BERTScore** (Zhang et al. 2019), which measures semantic similarity with BERT. (3) **Average Length** of generated vehicles, which is important because longer vehicles are generally more expressive and contain more semantics.

We compare our methods with following baselines: (1) **RTRVL** (Chakrabarty, Muresan, and Peng 2020): They adopt the HasProperty relation from ConceptNet (Speer, Chin, and Havasi 2017) to choose the optimal vehicle. (2) **SCOPE** (Chakrabarty, Muresan, and Peng 2020): They fine-tune BART via automatically constructed parallel datasets. (3) **META** (Stowe, Ribeiro, and Gurevych 2020): They fine-tune BART via parallel data containing masked literal sentence and similes. (4) **LMASKB** (Chen et al. 2022): They generate vehicle by prompting via designed templates. (5) **GPT-3** (Brown et al. 2020): We prompt GPT3-Davinci-002 to generate similes given 4 random demonstrative examples.

Results Results shown in Table 7 suggest that our methods significantly outperform prior methods in this task, even though we do not use the train set. This not only validates the effectiveness and quality of simile knowledge in

Method	TS	BLEU1	BLEU2	BERT-S	Length
RTRVL	N	00.00	00.00	12.94	01.59
LMASKB	N	00.53	00.00	14.05	01.00
META	Y	03.73	00.96	15.14	01.58
SCOPE	Y	08.03	03.59	<u>18.04</u>	01.87
GPT-3	N	<u>12.40</u>	02.87	16.96	03.20
MK _{PLM}	N	11.99	03.13	16.65	<u>02.95</u>
MK _{Rule}	N	04.80	03.27	10.47	02.03
MK _{PLM+Rule}	N	13.11	<u>03.44</u>	18.94	02.71

Table 7: Results of metrics BLEU-1, BLEU-2, BERTScore and Average Length in the writing polishment task. MK represents our methods using MAPS-KB. TS indicates whether the train set from (Chakrabarty, Muresan, and Peng 2020) is used. Results of RTRVL, META and SCOPE are taken from (Chakrabarty, Muresan, and Peng 2020).

MAPS-KB, but also indicates that our methods with MAPS-KB can rewrite literal sentences to similes that are more plausible (higher BLEU1, BLEU2 and BERT-S) and contain more semantics (higher Average Length). Besides, the combined method MAPS-KB_{PLM+Rule} performs better than methods based solely on PLMs or rules. The improvement of MAPS-KB_{PLM+Rule} over MAPS-KB_{PLM} shows the practical value of MAPS-KB’s probabilistic information in this task. Furthermore, MAPS-KB_{Rule} surpasses the retrieval method RTRVL based on ConceptNet, which implies that MAPS-KB is better than existing commonsense knowledge bases in terms of polishing writing with similes.

Conclusions

In this work, we introduce MAPS-KB, the first million-scale probabilistic simile knowledge base. Specifically, MAPS-KB covers 4.3 million triplets with probabilistic information over 0.4 million terms from 70 GB corpora. The proposed construction framework consists of four steps and can be extended to other figurative languages. We further conduct intrinsic and extrinsic evaluation to verify the effectiveness of our KB and framework, and achieve state-of-the-art performance on three downstream tasks.

Acknowledgements

This work is supported by Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103), Science and Technology Commission of Shanghai Municipality Grant (No. 22511105902), National Natural Science Foundation of China (No.62102095). Yanghua Xiao is also a member of Research Group of Computational and AI Communication at Institute for Global Communications and Integrated Media, Fudan University.

References

- Bar, K.; Dershowitz, N.; and Dankin, L. 2020. Automatic Metaphor Interpretation Using Word Embeddings. *arXiv preprint arXiv:2010.02665*.
- Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Celikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4762–4779.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.
- Chakrabarty, T.; Choi, Y.; and Shwartz, V. 2022. It’s not rocket science: Interpreting figurative language in narratives. *Transactions of the Association for Computational Linguistics*, 10: 589–606.
- Chakrabarty, T.; Muresan, S.; and Peng, N. 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)*, 6455–6469.
- Chen, W.; Chang, Y.; Zhang, R.; Pu, J.; Chen, G.; Zhang, L.; Xi, Y.; Chen, Y.; and Su, C. 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)*, 5875–5887.
- Cui, W.; Xiao, Y.; Wang, H.; Song, Y.; Hwang, S.-w.; and Wang, W. 2017. KBQA: Learning Question Answering over QA Corpora and Knowledge Bases. *Proceedings of the VLDB Endowment*, 10(5).
- Fishelov, D. 2007. Shall I compare thee? Simile understanding and semantic categories. *Journal of literary semantics*, 36(1): 71–87.
- Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5): 378.
- Hanks, P. 2013. *Lexical analysis: Norms and exploitations*. Mit Press.
- He, Q.; Cheng, S.; Li, Z.; Xie, R.; and Xiao, Y. 2022. Can Pre-trained Language Models Interpret Similes as Smart as Human? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7875–7887.
- Lakoff, G. 1993. The contemporary theory of metaphor. *Metaphor and Thought*.
- Lakoff, G.; and Johnson, M. 2008. *Metaphors we live by*. University of Chicago press.
- Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 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*, 7871–7880.
- Li, H.; Zhu, K. Q.; and Wang, H. 2013. Data-driven metaphor recognition and explanation. *Transactions of the Association for Computational Linguistics*, 1: 379–390.
- Liu, E.; Cui, C.; Zheng, K.; and Neubig, G. 2022. Testing the Ability of Language Models to Interpret Figurative Language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4437–4452.
- Liu, L.; Hu, X.; Song, W.; Fu, R.; Liu, T.; and Hu, G. 2018. Neural multitask learning for simile recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1543–1553.
- Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; and Wang, P. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2901–2908.
- Miller, G. A. 1998. *WordNet: An electronic lexical database*. MIT press.
- Niculae, V.; and Danescu-Niculescu-Mizil, C. 2014. Brighter than gold: Figurative language in user generated comparisons. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2008–2018.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.
- Paul, A. M. 1970. Figurative language. *Philosophy & Rhetoric*, 225–248.
- Roncero, C.; and de Almeida, R. G. 2015. Semantic properties, aptness, familiarity, conventionality, and interpretive diversity scores for 84 metaphors and similes. *Behavior research methods*, 47(3): 800–812.
- Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.
- Stowe, K.; Ribeiro, L.; and Gurevych, I. 2020. Metaphoric paraphrase generation. *arXiv preprint arXiv:2002.12854*.
- Tversky, A. 1977. Features of similarity. *Psychological review*, 84(4): 327.
- Veale, T.; and Hao, Y. 2007. Learning to understand figurative language: From similes to metaphors to irony. In *Proceedings of the annual meeting of the cognitive science society*, volume 29.

- Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.
- Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 950–958.
- Wu, W.; Li, H.; Wang, H.; and Zhu, K. Q. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*, 481–492.
- Xiao, P.; Alnajjar, K.; Granroth-Wilding, M.; Agres, K.; Toivonen, H.; et al. 2016. Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In *Proceedings of the Seventh International Conference on Computational Creativity*. Sony CSL Paris.
- Zeng, J.; Song, L.; Su, J.; Xie, J.; Song, W.; and Luo, J. 2020. Neural simile recognition with cyclic multitask learning and local attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9515–9522.
- Zhang, J.; Cui, Z.; Xia, X.; Guo, Y.; Li, Y.; Wei, C.; and Cui, J. 2021. Writing polishment with simile: Task, dataset and a neural approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14383–14392.
- Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Zheng, D.; Song, R.; Hu, T.; Fu, H.; and Zhou, J. 2019. “Love Is as Complex as Math”: Metaphor Generation System for Social Chatbot. In *Workshop on Chinese Lexical Semantics*, 337–347. Springer.