A Knowledge Graph Embedding Approach for Metaphor Processing

Wei Song, Jingjin Guo, Ruiji Fu, Ting Liu, and Lizhen Liu

***Abstract*—Metaphor is a figure of speech that describes one thing (a target) by mentioning another thing (a source) in a way that is not literally true. Metaphor understanding is an interesting but challenging problem in natural language processing. This paper presents a novel method for metaphor processing based on knowledge graph (KG) embedding. Conceptually, we abstract the structure of a metaphor as an attribute-dependent relation between the target and the source. Each specific metaphor can be represented as a metaphor triple $(target, attribute, source)$. Therefore, we can model metaphor triples just like modeling fact triples in a KG and exploit KG embedding techniques to learn better representations of concepts, attributes and concept relations. In this way, metaphor interpretation and generation could be seen as KG completion, while metaphor detection could be viewed as a representation learning enhanced concept pair classification problem. Technically, we build a Chinese metaphor KG in the form of metaphor triples based on simile recognition, and also extract concept-attribute collocations to help describe concepts and measure concept relations. We extend the translation-based and the rotation-based KG embedding models to jointly optimize metaphor KG embedding and concept-attribute collocation embedding.**

Experimental results demonstrate the effectiveness of our method. Simile recognition is feasible for building the metaphor triple resource. The proposed models improve the performance on metaphor interpretation and generation, and the learned representations also benefit nominal metaphor detection compared with strong baselines.

***Index Terms*—Metaphor processing, Knowledge graph embedding, Metaphor generation, Metaphor interpretation, Metaphor detection.**

Manuscript received October 12, 2019; revised April 5, 2020, July 18, 2020, and September 25, 2020; accepted November 8, 2020. Date of publication December 11, 2020; date of current version December 21, 2020. This work was supported in part by the Beijing Natural Science Foundation under Grant 4 192 017, in part by the National Natural Science Foundation of China under Grants 61 876 113 and 61 876 112, and in part by the Project of High-level Teachers in Beijing Municipal Universities in the Period of 13th Five-year Plan (CIT&TCD20170322) and Capital Building for Sci-Tech Innovation-Fundamental Scientific Research Funds. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kai Yu.
(Corresponding author: Lizhen Liu.)

Wei Song, Jingjin Guo, and Lizhen Liu are with the College of Information Engineering, and Academy for Multidisciplinary Studies, Capital Normal University, Beijing 100048, China (e-mail: wsong@cnu.edu.cn; jjguo@cnu.edu.cn; lizliu_cnu@163.com).

Ruiji Fu is with the State Key Laboratory of Cognitive Intelligence, iFLYTEK Research & iFLYTEK AI Research (Hebei), Langfang 230088, China (e-mail: rjfu@iflytek.com).

Ting Liu is with the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin 150001, China (e-mail: tliu@ir.hit.edu.cn).

Digital Object Identifier 10.1109/TASLP.2020.3040507

I. Introduction

Metaphor is a figure of speech that is ubiquitous in our daily language [1]. It provides a mechanism for reasoning by analogy [2]. In recent years, considerable attention has been paid to metaphor computation, since understanding metaphors is essential to understanding the implications of texts.

Conceptual metaphor is a well accepted theory for metaphor understanding [1]. The theory claims that a metaphor involves two concepts or conceptual domains: target and source. The target is described by or identified with the source, when two things have similar attributes in some way, although they are not alike in most other ways [3]. In the metaphor "*argument is war.*," the target *argument* is described by the source *war*, because they might be both *intense*, although the shared attribute is not explicitly shown. Therefore, to understand a metaphor, two kinds of representations are important: the concept domains and the cross-domain mappings. Especially, we should pay attention to the attributes shared by the source and the target.

This paper proposes a novel framework to learn representations of concepts, attributes and concept mappings. We abstract the structure of metaphors as an attribute-dependent concept mapping and represent a specific metaphor as a triple

$$(target, attribute, source).$$

We can assume that all metaphors are represented as such metaphor triples. Known metaphors are visible triples, while others are completely or partially unobservable triples. These triples have the same form as entity-relation triples in fact KGs [4]. From this view, a metaphor can be formulated as a concept relation between the target and the source, which is specified through the shared attribute. These observations motivate us to exploit KG embedding techniques for modeling metaphors.

Fig. 1 illustrates our main framework. We start by building a metaphor triple set as a metaphor KG based on simile recognition. Simile is a figurative language with an explicit comparator such as *like* or *as*. We first manually extract metaphor triples from simile sentences, and then train a simile recognizer that can automatically extract more metaphor triples to expand the initial ones. Moreover, we extract concept-attribute collocations through dependency parsing patterns. The resource can complement the metaphor triples to better describe concept domains.

Given the above resources, we propose to learn representations of concepts, attributes and metaphor structures by jointly

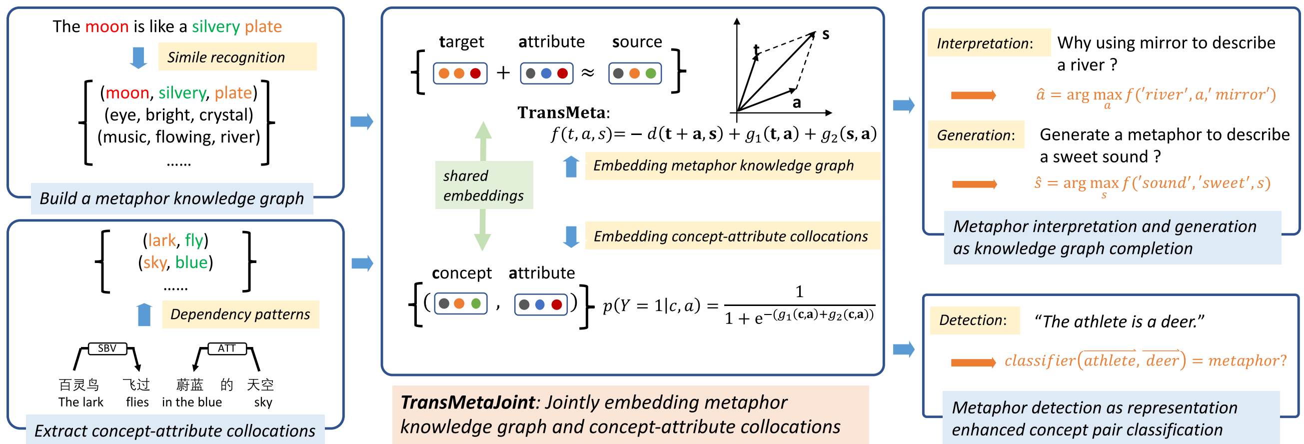


Fig. 1. Framework of modeling attribute-dependent concept mappings for metaphor processing. Simile recognition is used to build a metaphor triple set as a metaphor KG, while dependency parsing is applied to extract concept-attribute collocations. **TransMetaJoint** model is proposed by jointly embedding the metaphor KG and concept-attribute collocations. Three metaphor processing tasks could be handled within the same framework. ($\mathbf{t}$, $\mathbf{a}$ and $\mathbf{s}$ are the representations of the target, attribute and source; d is a dissimilarity function and g_1/g_2 are used to measure relatedness between a target/source and an attribute).

embedding the metaphor KG and the concept-attribute collocations.

To model the metaphor triples, we expect that the representations of the target and the source of a metaphor should be close enough when considering the attribute they share, i.e.,

$$composition(\overrightarrow{target}, \overrightarrow{attribute}) \simeq \overrightarrow{source},$$

where $composition$ is some semantic composition function. This intuition is the same as previous translation-based [5]–[8] and rotation-based [9] KG embedding models, where *addition* and *rotation* operations can be used as the *composition* function. However, the attribute should have a close relation to the source or the target, which is essential for representing metaphors but is not explicitly modeled by previous models. As a result, we propose a new formulation for scoring a metaphor triple by adding target-attribute and source-attribute relatedness measures. Taking the translation-based models as an example, the middle part of Fig. 1 shows a proposed scoring function **TransMeta** for a triple (t, a, s). In addition to the dissimilarity function d, two more functions g_1 and g_2 are proposed to measure the relatedness between a target/source and an attribute.

To better incorporate the associations between concepts and attributes, we further propose the joint model (e.g., the **TransMetaJoint** model in Fig. 1) for jointly optimizing metaphor KG embedding and concept-attribute collocation embedding, which are connected through the target-attribute and source-attribute relatedness measures g_1 and g_2.

Finally, we demonstrate that main metaphor processing tasks could be conducted within the same framework. Metaphor interpretation and generation tasks are processed as the KG completion problem and metaphor detection task can be viewed as representation enhanced concept pair classification.

We conduct comprehensive experiments for evaluating the proposed approach. We find that a two-stage simile recognizer works well on simile sentence classification and component extraction. With the expanded triples as extra training data, our approach obtains large improvements on metaphor interpretation and generation. The proposed metaphor triple scoring function and the joint learning framework achieve significant improvements on metaphor interpretation and generation compared with previous translation-based and rotation-based models in automatic evaluation and also outperform state-of-the-art systems in human evaluation. Qualitative analysis shows that our method has a clear interpretability, can distinguish targets and sources, and capture the abstractedness of concepts. We also show that combining ours and pre-trained generic word embeddings can achieve the best performance on nominal metaphor detection. To summarize, this paper has the following contributions:

- We are the first to formulate a metaphor as an attribute-dependent concept mapping and manipulate metaphor processing based on KG embedding.
- We propose to jointly optimize metaphor KG embedding and concept-attribute collocation embedding to learn the representations of concepts, attributes, concept relations simultaneously, and demonstrate the effectiveness on three metaphor processing tasks.
- We construct a Chinese metaphor KG and extract concept-attribute collocations to describe concepts through their attributes. These resources and the datasets used in experiments will be released.

II. Related Work

A. Simile Processing for Building Metaphor Resources

Simile is a simpler form of figurative language with explicit comparators such as *like* or *as*. Learning from explicit similes has been thought to be useful for comprehending and generating non-explicit metaphors [10].

One line of work is to extract concepts' attributes (or properties) from similes based on heuristic patterns [10]–[13]. Such resources are useful for metaphor understanding. However, the pattern based methods are difficult to deal with sentences with complex structures, leading to low coverage. Moreover, sentences with a comparator or a pattern do not always have a figurative meaning.

Instead of using patterns, simile recognition task learns to recognize simile sentences and components based on annotated examples. Niculae and Danescu-Niculescu-Mizil [14] proposed to classify a comparator into figurative or literal meaning in product reviews, but they assumed simile components are identified already. Neural multi-task learning approaches have been proposed to jointly optimize simile sentence classification and simile component extraction [15], [16], outperforming feature-based methods [14], [17].

Our work builds a metaphor KG by analyzing similes, which is motivated by previous work. Different from previous metaphor knowledge bases, such as Berkeley metaphor list [18] and MetaBank [19], the metaphor KG we build is in the form of metaphor triples, which imply an attribute-dependent concept mapping, i.e., a new view of metaphors. Although this procedure of extracting metaphor relations is similar to open information extraction [20], the focus of this paper is representation learning on the metaphor KG, which is not well studied in previous work.

B. Metaphor Processing

From the theoretical perspective, the concept mapping theory [1] is widely used as a fundamental theory for metaphor processing, which explains a metaphor as a mapping between two conceptual domains. The main metaphor processing tasks include metaphor detection, interpretation and generation.

1) Metaphor Detection: Metaphor detection is to distinguish expressions as metaphorical or literal, which is typically viewed as a supervised classification problem. The key is to represent semantic domains and learn domain mappings through annotated metaphor examples.

The feature-based methods are based on linguistic motivated manual features to describe concept domains and capture semantic relations. Shutova *et al.* [21] constructed domains by clustering nouns and verbs. Turney *et al.* [22] considered the abstractness of contexts. Shutova and Sun [23] incorporated the distinction between abstract and concrete concepts. Mohler *et al.* [24] exploited Wikipedia and WordNet to build domain signatures. Neuman *et al.* [25] incorporated selectional preferences [26]. Other semantic features include verbal arguments [27], discourse [28] and visual features [29]. Tsvetkov *et al.* [30] combined abstractness, imageability, supersenses, word vectors of concepts and cross-lingual features. However, these surface level features heavily rely on linguistic resources and are difficult to represent and compute meanings.

Recently, distributed representations are exploited for metaphor detection. Distributed word embeddings were used as features [30] or to measure semantic relatedness [31]. However, general purpose word embeddings may not embed metaphor structures.

Rei *et al.* [32] proposed a similarity network to capture interactions between words to directly optimize metaphor detection. Token level metaphor detection [33] considers contextual representations [34], [35]. These methods only depend on annotated examples, and ignore attributes of concepts that are not annotated. Bulat *et al.* [36] used the attribute-based concept representations to detect metaphors.

Previous works separates concept representation and concept mapping learning, while our method learns attribute based concept representations and concept mappings in a unified representation learning framework.

2) Metaphor Interpretation and Generation: The early work depends on hand-coded knowledge [37]. Shutova *et al.* [38] cast metaphor interpretation as a paraphrasing task and derived literal paraphrases for metaphorical expressions.

Metaphor interpretation and generation often depend on measuring word associations. Abe *et al.* [39] transformed adjective modified nouns into *A_Like_B* style metaphors based on the probabilistic relationship between words. Xiao *et al.* [40] measured word associations from a large corpus based on the statistical significance of their co-occurrence. Qadir *et al.* [41] used syntactic structures, dictionary definitions, and word embeddings to infer implicit properties in similes. Su *et al.* [42] computed the relatedness of the concept-pairs by the cosine of its distributed representations. Salience imbalance theory [3] is often used together for candidate selection [43], [44]. However, most of these methods do not consider attribute-dependent concept mappings.

Although the previous solutions of metaphor processing are based on similar theories, they are usually solved with different frameworks. This paper proposes a novel view of metaphor processing as metaphor KG embedding. We propose a framework to jointly model concepts, their attributes and concept mapping relations and the main metaphor processing tasks could be handled with the same framework.

C. Knowledge Graph Embedding

KG embedding is to embed components of a KG such as entities and relations into continuous vector spaces so that the inherent structure of the KG can be preserved, while the KG manipulation is simplified [4].

In this paper, we represent attribute-dependent concept mappings as metaphor triples, which have a similar form to entity-relation triples in fact KGs. We extend the translation-based models [5]–[8] and the recently proposed RotatE model [9] for embedding the metaphor KG by considering metaphor specific characteristics and jointly embedding the concept-attribute collocations.

III. Overview

Our main idea is considering a metaphor as an attribute-dependent mapping, which could be represented as a triple (target, attribute, source), where the *target* and the *source* are concepts from different domains and the *attribute* plays a role as a relation. Our approach has the following parts.

Resource Construction: Our method is based on two types of resources. First, we build a metaphor KG by analyzing similes to capture attribute-dependent concept mappings. We manually extract metaphor triples from simile sentences to build an initial metaphor KG. Due to the limited amount of manually annotated triples, we train a simile recognizer to automatically expand the initial metaphor KG. Second, since concept-attribute collocations are important for representing the sense of concepts [45] and understanding metaphors [10], [46], we use dependency parsing patterns to extract concept-attribute collocations to complement the metaphor KG.

Jointly Embedding Metaphor KG and Concept-Attribute Collocations: We propose to utilize KG embedding for representation learning to preserve metaphor structures. For embedding the metaphor KG, we extend the translation-based and rotation-based models by proposing a new formulation for scoring metaphor triples. Moreover, we propose to jointly optimize metaphor KG embedding and concept-attribute collocation embedding to learn better representations of concepts, attributes and concept-attribute relatedness measures.

Knowledge Graph Embedding based Metaphor Processing: We demonstrate that main metaphor processing tasks can be conducted as inference over knowledge graph: metaphor interpretation and generation can be seen as knowledge graph completion, which can be converted as a ranking problem based on the proposed models, while metaphor detection can be seen as a representation enhanced concept pair classification problem.

IV. Metaphor Resource Construction

A. Building a Metaphor KG Manually

Simile: is a figurative language with an explicit comparator such as *like* or *as*. Some similes contain attributes as well in addition to the source and the target, e.g.,

$$The\ moon\ is\ like\ a\ silvery\ plate,$$

where the target is *moon*, the source is *plate* and one of their shared attribute is *silvery*. Due to the existence of explicit comparators, similes are easier to be located compared with metaphors, which do not have an explicit comparator. However, not all sentences containing a comparator have a figurative meaning. For example, the sentences such as *the boy looks like his father.* and *we love the weather like today.* just express literal meanings.

We sampled more than 10 000 sentences, which contain the most frequently used comparator in Chinese–像 *(like)*, from an essays corpus. The essays were crawled from a website LeleKetang.[1] The essays were written by junior and senior high school Chinese students, covering diverse genres and topics.

We manually labeled each sentence as a simile sentence or not. From the simile sentences (∼5000), we further annotated simile components: *source*, *target* and *attribute*. Currently, we restrict that source and target concepts should be single nouns and attributes should be verbs or adjectives.

The annotation was conducted by two students from the college of literal arts. The inter-annotator agreement on labeling simile sentences on a sample of 200 sentences is 91%. For labeling simile components, two annotators labeled 200 simile sentences and 94% of their extracted triples are agreed.

[1] [Online]. Available: http://www.leleketang.com/

TABLE I
Statistics of Constructed Metaphor Triples From Simile Sentences

Metaphor triples	Number
# triple	2530
# distinct concepts	1,329
# distinct sources	830
# distinct targets	729
# distinct attributes	1,158

Some statistics of this initial metaphor KG are listed in Table I.

B. Metaphor KG Expansion Through Simile Recognition

In order to expand the initial metaphor KG, we study the simile recognition task based on the manually annotated simile sentences. We divide simile recognition into two subtasks:

Simile Sentence Classification: For a sentence containing a comparator, determine whether the sentence is a simile sentence.

Simile Component Extraction: From a (predicted) simile sentence, extract simile components including the target, source and attribute to build metaphor triples.

1) Simile Sentence Classification: We use BERT [47] for simile sentence classification. We fine-tune BERT on a dataset of sentences containing the comparator 像, including 5000 simile sentences and 5000 non-simile sentences. For each sentence, we take the final layer hidden state of the special token [CLS] as the sentence representation. This representation is mapped to a softmax layer through a fully connected forward network to get a classification probability. All of the parameters of BERT and the forward network are fine-tuned simultaneously to maximize the log-probability of the correct labels.

2) Simile Component Extraction: We regard the task as a sequence labeling problem at Chinese character level. We convert the annotated dataset to IOBES scheme (indicating Inside, Outside, Beginning, Ending, Single) and use different prefixes to distinguish different components. We use BERT to encode an input sentence and add a Conditional Random Field (CRF) layer [48] on the top of BERT to enhance interactions among predictions.

C. Evaluating Simile Recognition

1) Settings: The simile dataset is randomly divided into 5 folds, 4 of which are used as training set and validation set (80% for training, 20% for validation), and the rest one fold is used as test set. All models were trained on the training set. The best hyper-parameters were gained based on the results on the validation set. We fine-tune the pre-trained BERT-Chinese model with Adam [49] for optimization with a learning rate of 0.001.

2) Evaluating Simile Sentence Classification: We compare with the following baselines.

ManualFeature [17]: With manually designed features, we build a Random Forest classifier to determine whether a sentence contains a simile. The features include: (1) the tokens and POS tags of the words around the comparator within a fixed window (set to 5 in experiments); (2) the tokens, POS tags and

TABLE II
EXPERIMENTAL RESULTS ON SIMILE SENTENCE CLASSIFICATION

Model	**Simile Classification**		
	P	R	F1
ManualFeature [17]	0.77	0.78	0.77
NeuralMTL [15]	0.81	0.92	0.86
BERT	0.89	0.89	0.89

dependency relation tags of the words that have dependency relations with the comparator.

NeuralMLT [15]: It exploits a neural multitask learning framework, jointly optimizing three tasks: simile sentence classification, simile component extraction and language modeling.

Results: Table II shows that the fine-tuned BERT performs best. It outperforms NeuralMLT by 3% and outperforms ManualFeature by 12% in F1 score. The best F1 score can reach to 0.89.

3) Evaluating Simile Component Extraction: We compare the following systems.

CRF: The features include the tokens, their POS tags, the comparator feature (whether the token is a comparator), the position feature (the distance between the token and the comparator) and the dependency parsing based features. We consider these token features within a fixed context window (set to 5 in experiments).

LSTM_CRF: Instead of using the manually features, we use the bidirectional LSTM [50] to learn contextual distributed token representations and add a CRF layer on top for sequence labeling.

BERT: We also compare the performance of token-level fine-tuned BERT model. We directly feed the final hidden representation of each token into a classification layer over the label set.

BERT_CRF: This is the simile component extraction model described in section IV-B2.

Metrics: We report the performance on both component level and sentence level. The component level evaluation is similar to evaluating named entity recognition. We use precision, recall and F1 score as metrics. At sentence level, a sentence is correctly recognized only if all its components are correctly identified.

Results: We conduct experiments on two datasets. The first dataset is a subset of the whole test set, which consists of all manually labeled simile sentences. The second dataset is the whole test set, which has both simile and non-simile sentences.

Table III presents the results. The first 4 rows show the performance of directly applying the component extractors on two datasets.

We first compare the performance on two datasets. On gold simile sentence dataset, the systems have higher scores. On the whole test dataset, all systems' performance decreases due to the noises from non-simile sentences. This indicates that simile sentence classification should be an important pre-processing for simile component extraction.

On both datasets, the performance of neural network based models greatly outperforms the feature-based CRF model. BERT based models outperform LSTM based model, which verifies the power of pre-trained language encoders. Adding a CRF layer on the top of BERT further improves the performance on the whole test dataset.

Based on the above observations, we further evaluate a **pipeline** approach. We first classify a sentence as simile or not with the BERT simile sentence classifier, then use the BERT_CRF model to extract components from simile sentences. The last row in Table III presents the performance of the pipeline on the whole test. We can see that a further improvement is gained. The performance on component level is close to 0.80 in F1 score, but the performance on sentence level is relatively lower.

4) Error Analysis: An important issue is whether the outputs of the simile component extractor are good enough for metaphor KG expansion. Therefore, we analyze the main error types produced by the strongest pipeline system.

In the whole test set, the components of 38% simile sentences are perfectly extracted. The errors in the remaining sentences can be categorized into two categories: **acceptable errors** and **unacceptable errors**.

Acceptable errors would not pollute the metaphor KG. This category of error types include *incomplete triples* that are not completely extracted (about 19%) and *extracting additional triples* that the system extracts triples that were not annotated by human labelers but are still viewed as reasonable after rechecking (about 20.5%). Most such cases involve pronouns. The human labelers prefer to annotate entities while the system labels references sometimes. Incomplete triples would be automatically filtered after extraction so that they would not be added to the initial metaphor KG. Additional triples actually bring in new knowledge.

Unacceptable errors relate to the extracted triples that are complete but do not have a figurative meaning. This type of errors accounts for 22.5% sentences. Among these, about 11% errors are caused by simile classification errors, where simile components are extracted from false positive simile sentences. The others are component extraction errors, including identifying wrong targets, sources or attributes. In summary, although noises exist, most extracted metaphor triples are reasonable and safe to expand the metaphor KG.

5) Metaphor Triple Extraction: We use the pipeline approach to automatically extract more metaphor triples from about 90 000 sentences with a comparator from the LELEKETANG essay corpus. We only keep triples that all three types of simile components could be extracted from the same sentence. Moreover, we constrain that all the concepts and attributes should be in a Chinese synonymy thesaurus to further reduce noises [51].

Finally, we extracted 7630 metaphor triples. The statistics of the automatically extracted metaphor triples and the combination of manually and automatically built metaphor triples are shown in Table IV.

D. Extracting Concept-Attribute Collocations

Collocation is defined as the co-occurrence of two words in some defined relationship [45], which is widely used to shape the sense of words. We extracted adjective-noun and noun-verb

TABLE III
EXPERIMENTAL RESULTS ON SIMILE COMPONENT EXTRACTION

Model	Gold Simile Sentences						Whole Test Sentences					
	Component			Sentence			Component			Sentence		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CRF	0.79	0.64	0.70	0.32	0.32	0.32	0.32	0.68	0.53	0.18	0.21	0.20
Bilstm_CRF	0.79	0.79	0.79	0.45	0.45	0.45	0.68	0.69	0.68	0.30	0.37	0.33
BERT	0.80	0.83	0.81	0.48	0.48	0.48	**0.74**	0.69	0.73	0.35	0.40	0.38
BERT_CRF	0.81	0.83	0.82	0.49	0.49	0.49	0.72	0.77	0.75	0.41	0.44	0.43
Pipeline	–						**0.74**	**0.81**	**0.78**	**0.43**	**0.47**	**0.45**

TABLE IV
STATISTICS OF AUTOMATICALLY EXTRACTED METAPHOR TRIPLES AND THE COMBINATION OF MANUALLY AND AUTOMATICALLY BUILT METAPHOR TRIPLES

	Metaphor Triples	Number
Expanded triples	# triples	7,630
	# distinct concepts	2,446
	# distinct sources	1,492
	# distinct targets	1,544
	# distinct attributes	1922
Manual+ expanded triples	# triples	10,160
	# distinct concepts	2,857
	# distinct sources	1,756
	# distinct targets	1,820
	# distinct attributes	2,303

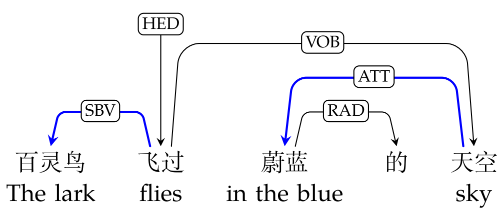


Fig. 2. Dependency parsing tree for a Chinese sentence. The blue arcs indicate two of the patterns used to extract concept-attribute collocations, e.g., (百灵鸟/lark, 飞过/fly) and (天空/sky, 蔚蓝/blue).

TABLE V
DEPENDENCY PARSING PATTERNS THAT ARE USED TO EXTRACT CONCEPT-ATTRIBUTE COLLOCATIONS. SBV AND VOB INDICATE THE SUBJECT-VERB AND VERB-OBJECT RELATIONS, AND NOUN, VERB AND ADJECTIVE ARE PART-OF-SPEECH TAGS OF THE WORDS

Dependency Parsing Patterns
$noun \leftarrow SBJ - verb$
$noun \leftarrow SBV - adj$
$noun \leftarrow SBV -$ 是 (be) $- VOB \rightarrow adj$
$adj \leftarrow ATT - noun$

collocations from the LELEKETANG essay corpus based on dependency parsing patterns.[2] Fig. 2 shows the dependency parsing tree for a Chinese sentence. We view adjectives and verbs as attributes, nouns as concepts, and call all the extracted collocations as **concept-attribute collocations**. The patterns that are used to extract concept-attribute collocations are listed in Table V.

In this way, we extract concept-attribute collocations in the form of $(concept, attribute, count)$, e.g., $(sky, blue, 90)$ and $(lark, fly, 25)$, where $count$ is the number of a collocation in the extracted data.

There are attributes that are commonly used by many concepts, such as *become* and *good*. They can't imply properties of specific concepts. Therefore, we remove the attributes that modify more than 15% of the concepts in our collected data. The collocations with a frequency less than 2 are removed to reduce noises. Again, we use the Chinese synonymy thesaurus to filter collocations with out-of-vocabulary words. Finally, we have 84 000 concept-attribute collocations including 4284 distinct concepts and 2919 distinct attributes.

[2]We use the dependency parser from the HIT-LTP toolkit.

V. JOINTLY EMBEDDING METAPHOR KG AND CONCEPT-ATTRIBUTE COLLOCATIONS

Given the metaphor KG in the form of metaphor triples, we can apply existing KG embedding models for representation learning. However, metaphor relations have special properties that are different from entity relations. For example, the attributes of concepts are important for describing concept domains and establishing metaphor relations. But such information is not explicitly or sufficiently modeled previously. As a result, we propose a novel method by jointly embedding the metaphor KG and concept-attribute collocations to better capture the inherent structures of metaphors.

We adopt the representative translation-based models as the main base models and show that our idea is easy to be implemented with the recently proposed rotation-based RotatE model as well.

A. Notations

Definition 1: Let $\mathcal{C}$ and $\mathcal{A}$ be the concept set and the attribute set.

Definition 2: Let $\mathcal{K} = \{(t, a, s)\}$, where $t \in \mathcal{C}, a \in \mathcal{A}, s \in \mathcal{C}$, be the metaphor KG, which has a set of attribute-dependent concept mappings {(target, attribute, source)}.

DEFINITION 3. Let $\mathcal{CA} = \{(c, a)\}$, where $c \in \mathcal{C}, a \in \mathcal{A}$, be the concept-attribute collocations corresponding to the set of collocations $\{(concept, attribute)\}$.

Our task is to learn the embeddings of the concepts and attributes that could benefit representation learning based metaphor processing, i.e.,

$$\mathcal{K}, \mathcal{CA} \xrightarrow{embedding} \mathbf{C}, \mathbf{A}, f$$

where $\mathbf{C} \in \mathbb{R}^{|\mathcal{C}| \times n}$ and $\mathbf{A} \in \mathbb{R}^{|\mathcal{A}| \times m}$ represent the embeddings of all concepts and attributes with n, m being the dimensions of concept embeddings and attribute embeddings, and f represents a function for inferring metaphor relation.

B. Translation-Based Models

The translation-based models learn representations for fact triples $\{$(head, relation, tail)$\}$, where $head$ is the head entity and $tail$ is the tail entity and a *relation* between them holds, e.g., (*tokyo, is_capital, japan*). In our task, these models can be naturally applied to learn the representation $(\mathbf{t}, \mathbf{a}, \mathbf{s})$ for each attribute-dependent concept mapping $(t, a, s) \in \mathcal{K}$, where $\mathbf{t}, \mathbf{s} \in \mathbf{C}, \mathbf{a} \in \mathbf{A}$.

TransE: [5] is the first translation-based model. It proposes to force the representations to satisfy $\mathbf{t} + \mathbf{a} \simeq \mathbf{s}$ for a triple (t, a, s) in $\mathcal{K}$. Several successive models extend TransE.

TransH: [6] enables a concept to have different representations for different attributes. The core idea is that the attribute vector $\mathbf{a}$ is placed in the attribute specific hyperplane $\mathbf{w}_a$ and the translation operation is done in this hyperplane.

TransR: [7] transforms concepts' representations to an attribute dependent space with a projection matrix and conducts the translation operation in a distinct space for each attribute. **TransD** [8] further extends **TransR** to use two vectors to represent a concept or an attribute. The first one represents the meaning of a concept or an attribute, while the other one is used to construct a mapping matrix dynamically.

In this paper, we use TransE, TransH and TransD as basic models. Without loss of generality, we use $\mathbf{t}_\perp$ and $\mathbf{s}_\perp$ to represent the projected vectors (including the identical mapping), the score of a triple is

$$f(t, a, s) = -d(\mathbf{t}_\perp + \mathbf{a}, \mathbf{s}_\perp),$$

where d is a dissimilarity measure, which could be either the L_1 or the L_2-norm.

C. TransMeta: Modeling Attribute-Dependent Concept Mappings

The previous translation-based models capture the fact that two concepts have certain relation. However, for metaphors, the attribute not only indicates two concepts have a relation, but also should be closely related to the source and/or the target.

To explicitly capture such properties, we propose the TransMeta model, which scores a triple as

$$f(t, a, s) = -d(\mathbf{t}_\perp + \mathbf{a}, \mathbf{s}_\perp) + g_1(\mathbf{t}_\perp, \mathbf{a}) + g_2(\mathbf{s}_\perp, \mathbf{a}), \quad (1)$$

where g_1 and g_2 are functions to measure the relatedness between a target and an attribute and between a source and an attribute, respectively, i.e.,

$$g_1(\mathbf{t}, \mathbf{a}) = \mathbf{t}_\perp \mathbf{M}_t \mathbf{a}^\top, g_2(\mathbf{s}, \mathbf{a}) = \mathbf{s}_\perp \mathbf{M}_s \mathbf{a}^\top. \quad (2)$$

We use different matrixes $\mathbf{M}_t, \mathbf{M}_s \in \mathbb{R}^{n \times m}$ for targets and sources, because some concepts may be more proper to be used as a target than as a source or vice versa. For example, abstract concepts are less used as a source.

Training: We follow the negative sampling strategy used by the translation-based models. For each correct triple, the target, the attribute and the source are replaced to form a corrupted triple.

Sampling negative examples: To reduce false negative examples, we selectively sample negative attributes, targets or sources. For generating a negative attribute, we randomly sample attributes from $\overline{\mathcal{A}_s \cup \mathcal{A}_t}$, where $\mathcal{A}_t, \mathcal{A}_s$ are the attribute sets of target t and source s, which are obtained from $\mathcal{K}$ and $\mathcal{CA}$. For generating a negative target or source, we randomly sample concepts, each of which, e.g., c', has the attribute set $\mathcal{A}_{c'} \cap \{a\} = \emptyset$. We replace source concepts two more times than target concepts, because the associations between attributes and sources are more important.

Formally, the loss function is

$$\mathcal{L}_1 = \sum_{(t,a,s) \in D^+} \sum_{(t,'a,'s') \in D^-_{(t,a,s)}} cost((t, a, s), (t,' a,' s')), \quad (3)$$

where D^+ is the set of correct triples in $\mathcal{K}$ and $D^-_{(t,a,s)}$ is a set of corrupted triples to a correct triple,

$$D^-_{(t,a,s)} = \{(t, a,' s) | a' \in \overline{\mathcal{A}_s \cup \mathcal{A}_t}\}$$

$$\cup \{(t, a, s') | \mathcal{A}_{s'} \cap \{a\} = \emptyset\} \cup \{(t,' a, s) | \mathcal{A}_{t'} \cap \{a\} = \emptyset\}.$$

The $cost$ function of a corrupted triple is

$$[b] cost((t, a, s), (t,' a,' s')) = \max(0, \gamma - f(t, a, s) + f(t,' a,' s')),$$

where $\gamma > 0$ is a margin hyper parameter.

D. TransMetaJoint: Jointly Learning Concept Attribute Collocation Embeddings and TransMeta

1) Embedding Concept Attribute Collocations: We further capture the associations between concepts and attributes in $\mathcal{CA}$. We define the probability that a word pair (c, a) is a true concept-attribute collocation as

$$p(Y = 1 | c, a) = \frac{1}{1 + \exp(-(g_1(\mathbf{c}, \mathbf{a}) + g_2(\mathbf{c}, \mathbf{a})))}, \quad (4)$$

where g_1 and g_2 are defined in Equation 8. In this way, we compute the probability based on the relatedness between a concept-attribute pair and establish a connection between the metaphor KG and concept-attribute collocations through the TransMeta scoring function.

We adopt a negative sampling strategy similar to the way in [52] for parameter learning. We aim to find the best parameters to maximize the probability of a pair of concept and attribute being in $\mathcal{CA}$ if it indeed is, and maximize the probability of a pair of concept and attribute not being in $\mathcal{CA}$ (i.e., in $\mathcal{CA}^-$) if it indeed is not, i.e.,

$$\prod_{(c,a) \in \mathcal{CA}} p(Y = 1 | c, a) \prod_{(c,'a') \in \mathcal{CA}^-} (1 - p(Y = 1 | c,' a')), \quad (5)$$

which is to minimize

$$\mathcal{L}_2 = -\sum_{(c,a)\in\mathcal{CA}} \log \frac{1}{1+\exp(-(g_1(\mathbf{c},\mathbf{a})+g_2(\mathbf{c},\mathbf{a})))} - \sum_{(c,'a')\in\mathcal{CA}^-} \log \frac{1}{1+\exp(g_1(\mathbf{c}',\mathbf{a}')+g_2(\mathbf{c}',\mathbf{a}'))}. \quad (6)$$

In practice, we construct $\mathcal{CA}^-$ by sampling k negative examples for each $(c,a)\in\mathcal{CA}$. Specifically, we sample a c' each time to replace c where $\mathcal{A}_{c'}\cap\{a\}=\emptyset$. The sampling is based on the frequencies of concepts in $\mathcal{CA}$.

2) TransMetaJoint: The metaphor triples reflect attribute-dependent concept relations and the concept-attribute allocations provide rich descriptions of concepts. We combine them together through joint learning that both tasks share the same concept and attribute embeddings, and the scoring functions g_1 and g_2. The final loss function is

$$\mathcal{L} = \mathcal{L}_1 + \beta\cdot\mathcal{L}_2, \quad (7)$$

where β is a non-negative parameter to control the weight of $\mathcal{L}_2$.

E. RotatEMeta and RotatEMetaJoint

We can also use other KG embedding methods to investigate our idea, e.g., the recently proposed RotatE model. RotatE defines each relation as a rotation from the target concept to the source concept in the complex vector space.

Formally, given a triple (t,a,s), $\mathbf{t},\mathbf{a},\mathbf{s}\in\mathbb{C}^n$, RotatE forces that $\mathbf{t}\circ\mathbf{a}\simeq\mathbf{s}$, where $\circ$ is the element-wise product and $|a_i|=1$, which means that a counterclockwise rotation defined by the attribute's representation $\mathbf{a}$ is applied to the embedding $\mathbf{t}$ to reach the embedding $\mathbf{s}$ in the complex vector space. So the score of a triple can be defined as

$$f(t,a,s) = -d(\mathbf{t}\circ\mathbf{a},\mathbf{s}).$$

1) RotatEMeta: Analogical to TransMeta, RotatEMeta uses the following two functions g_1 and g_2 to measure relatedness between the target/source and the attribute

$$g_1(\mathbf{t},\mathbf{a}) = -d(\mathbf{t}\circ\mathbf{r}_t,\mathbf{a}), g_2(\mathbf{s},\mathbf{a}) = -d(\mathbf{s}\circ\mathbf{r}_s,\mathbf{a}), \quad (8)$$

where $\mathbf{r}_t,\mathbf{r}_s\in\mathbb{C}^n$ are two parameter vectors in the complex vector space, which represent the $target\_attribute$ relation and the $source\_attribute$ relation, respectively.

2) RotatEMetaJoint: Given g_1 and g_2, RotatEMetaJoint is realized in the same way as TransMetaJoint by jointly modeling $\mathcal{K}$ and $\mathcal{CA}$.

VI. KG Embedding for Metaphor Processing Tasks

Main metaphor processing tasks include metaphor detection, interpretation and generation. This work focuses on dealing with nominal metaphors. All these tasks can be done within the proposed framework: metaphor interpretation and generation can be viewed as a KG completion problem and metaphor detection can be viewed as a representation enhanced concept pair classification problem.

The metaphor interpretation and metaphor generation can be seen as recovering explicit similes from implicit metaphors.

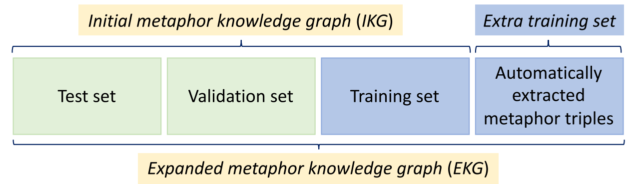


Fig. 3. Datasets used in our experiments.

Example 1: Explain the metaphor *Argument is war*.

Interpretation: We first infer the missing attribute in the incomplete triple *(argument, ?, war)*. With the inferred attribute (e.g.,*intense*), we can interpret the complete triple *(argument, intense, war)* as:

Argument is like war, because it is intense.

The resulted interpretation can be understood by both human and machines.

Example 2: Compose a metaphor to express *teachers' diligence*.

Generation: We view *teacher* as the target, *diligent* as an attribute. Metaphor generation is to infer the missing source concept to recover the incomplete triple *(teacher, diligent,?)*. After predicting the source (e.g., *gardener*), we can generate a metaphor like

Teachers are gardeners.

In our framework, we formulate metaphor interpretation and generation as the following two tasks.

Attribute Prediction: Given the target and the source $t,s\in\mathcal{C}$, recommend an attribute that can best explain the metaphorical meaning. This can be solved as a ranking problem to find $\hat{a}=\arg\max_a f(t,a,s)$.

Source Prediction: Given the target t and an attribute a, recommend a source concept that can be used to describe the attribute a of t. This can be solved as a ranking problem as well to find $\hat{s}=\arg\max_s f(t,a,s)$.

Moreover, metaphor detection could be formulated as:

Metaphor Detection: Given a pair of concepts (t, s), predict whether it has a meaningful metaphorical meaning with a classifier based on their representations $(\mathbf{t},\mathbf{s})$.

VII. Evaluating Metaphor Processing

A. Evaluating Metaphor Interpretation and Generation

We first conduct automatic evaluation of KG embedding models and then compare the proposed approach with previous methods.

1) Experimental Settings: We use the following settings for evaluation.

Dataset: We call the manually built initial metaphor KG IKG and the expanded metaphor KG EKG, as shown in Fig. 3. We randomly divided IKG into training set (70%), validation set (10%) and test set (20%). The expanded metaphor triples are used as extra training data. We ensure that there is no shared triple among these datasets.

Evaluation Procedure For each test instance, the attribute (or the source) to predict is masked. For prediction, we fill the masked field with each candidate (attribute or concept) in turn and compute the score of the completed triple as the score of the candidate. The candidates are then ranked to get a prediction list. For fairly comparison, we use all concepts or attributes in IKG as candidates.

Metrics: We use the following evaluation metrics:

Mean Rank(MR): the mean rank of the correct answers in corresponding prediction lists;

Mean Reciprocal Rank(MRR): the mean of the reciprocal ranks of the correct answers in corresponding prediction lists;

Hits @N: the proportion of the correct answers that are among the top N predictions. Hits@1 and Hits@10 would be reported.

We use the original attributes or sources in the metaphor triples in the test set as answers for automatic evaluation. Since a shared attribute could be described with different words. In addition to the **exact-match based evaluation**, we also report the **synonymy based evaluation** results. If a prediction and the corresponding answer have a synonymy relation in the Chinese synonymy thesaurus [51], the prediction is viewed as correct.

Model Variants: We use TransE, TransH, TransD and RotatE as the base models. The corresponding extensions with the new scoring functions are TransEMeta, TransHMeta, TransD-Meta and RotatEMeta. The corresponding final joint models are named as TransEMetaJoint, TransHMetaJoint, TransDMeta-Joint and RotatEMetaJoint, respectively. All the variants are implemented based on OpenKE [53].

Parameter Settings: The word embeddings for both concepts and attributes are randomly initialized and the dimension is set to 200. We use the Adadelta algorithm [54] for optimization with a learning rate of 1.0. The dissimilarity measure is set to L_1-norm. The margin hyper-parameter γ and the parameter β in Equation 7 are tuned on the validation set for each model. All systems iterate 1000 times. Each time, one negative example is built for every correct metaphor triple and every true concept-attribute collocation. During training, a model is saved every 50 epochs. The best one is chosen for testing based on the results on the validation set.

2) Automatic Evaluation Results: Table VI shows the results on IKG and EKG, respectively. The figures in brackets are the synonymy-based evaluation results. The trends of exact-match based and synonymy based evaluations are mostly consistent. We analyze the results from the following aspects.

The effect of EKG: By comparing the two sub-tables in Table VI, we can see that for all models, the performance on EKG, i.e., using the extra training data, is greatly improved on all metrics compared with using IKG only.

The comparison of the base models: For trans-series models, TransE and TransH achieve better performance than TransD on both attribute prediction and source prediction tasks. The results indicate that putting concept and attribute embeddings in the same space (e.g., TransE and TransH) is enough for metaphor processing. On both IKG and EKG, TransE has very competitive performance. RotatE achieves best results on several metrics on IKG, while its performance on EKG is slightly worse than TransE and TransH in average.

The performance of the joint models: On both IKG and EKG, the joint models gain large improvements compared with the base models. For example, when using EKG, the joint models can achieve a 33% Hits@10 relative improvement on attribute prediction, and a 18% Hits@10 relative improvement on source prediction in average using the exact-match based evaluation. The corresponding relative improvements using the synonymy based evaluation are 19% and 13%.

TransHMetaJoint performs best in general among the translation-based joint models, especially when the training data is large. Its performance is also consistent across tasks. The performance of RotatEMetaJoint is not so good as TransEMeta-Joint and TransHMetaJoint, except on source prediction on IKG. We plan to investigate more about measuring relatedness in the complex vector space in future.

Detailed results by attribute category: Following the way described in [5], we categorize the attributes into four types according to the cardinalities of their target and source arguments: 1-to-1, 1-to-many, many-to-1, many-to-many. Table VII shows the distribution of different attribute categories and the source prediction results on EKG according to the exact-match evaluation. We can see that the joint models gain improvements on almost all attribute categories, especially on m-to-1 type. TransHMetaJoint achieves consistently superior performance.

For a brief summary, the above results show that (1) the expanded metaphor triples are very useful for learning representations, indicating that the simile recognition based metaphor KG expansion is feasible and successfully brings in useful supervision; (2) For all model variants, the joint models outperform corresponding base models with a large margin. This means that joint modeling metaphor triples and concept-attribute collocations is effective, consistent and general enough; (3) among all model variants, TransHMetaJoint performs consistently well when the metaphor KG is large due to its ability to better deal with multi-relational metaphors.

3) Model Analysis: We take TransHMetaJoint as an example to analyze the contributions of the TransMeta scoring function and the joint embedding strategy. Figure 4 shows the exact-match based evaluation results of TransH, TransHMeta, TransHJoint and TransHMetaJoint on the attribute prediction and source prediction tasks. TransHJoint means jointly embedding metaphor triplets and concept-attribute collocations but without the g_1 and g_2 functions in Equation 1. The trend of the synonymy based evaluation results is similar.

TransHMeta outperforms TransH by a large margin on both Mean rank and Hits@10 on two tasks. This demonstrates that the TransHMeta scoring function that incorporates relatedness measures between concept-attribute pairs is effective due to proper description of the properties of metaphor relations.

The performance of TransHJoint is better than TransH, indicating the usefulness of bringing in concept-attribution collocations. But its performance is slightly lower than TransHMeta. In contrast, TransMetaHJoint achieves significant improvements compared with TransHJoint and TransHMeta. The results indicate the importance of the functions g_1 and g_2, which play a role as a bridge to connect metaphor KG embedding and concept-attribute collocation embedding. TransHMetaJoint gains large improvements on Mean rank on both tasks. This means that embedding concept-attribute collocations helps representation

TABLE VI
AUTOMATIC EVALUATION OF SYSTEMS ON ATTRIBUTE PREDICTION AND SOURCE PREDICTION. THE FIGURES IN BRACKETS ARE THE RESULTS OF SYNONYMY BASED EVALUATIONS. THE FIGURES IN BOLD MEAN THE BEST PERFORMANCE IN EACH COLUMN. (A) EXPERIMENTAL RESULTS ON THE INITIAL METAPHOR KG (IKG). (B) EXPERIMENTAL RESULTS ON THE EXPANDED METAPHOR KG (EKG)

(A)

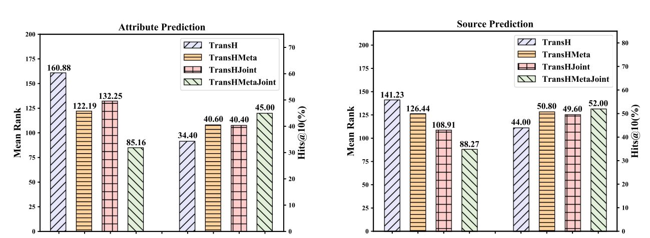
Model	Attribute Prediction			
	MR	MRR	Hits@1(%)	Hits@10(%)
TransE	366.74 (186.64)	0.077 (0.137)	2.40 (5.40)	19.80 (**30.40**)
TransH	365.45 (173.33)	0.072 (0.133)	1.80 (5.20)	17.80 (29.40)
TransD	409.61 (187.85)	0.080 (0.136)	**3.20** (**6.20**)	17.20 (26.80)
RotatE	**265.03** (**140.96**)	**0.087** (**0.138**)	3.00 (5.60)	**20.20** (29.00)
TransEMetaJoint	**154.71** (87.05)	0.147 (0.216)	7.20 (12.40)	29.80 (42.00)
TransHMetaJoint	171.49 (**87.02**)	**0.160** (**0.228**)	**8.00** (**13.00**)	**34.20** (**45.40**)
TransDMetaJoint	187.99 (90.74)	0.114 (0.193)	5.00 (9.80)	25.60 (39.60)
RotatEMetaJoint	239.28 (134.14)	0.111 (0.183)	5.40 (9.20)	22.80 (36.80)
Model	**Source Prediction**			
	MR	MRR	Hits@1(%)	Hits@10(%)
TransE	303.83 (205.03)	**0.149** (0.198)	**8.60** (12.20)	**27.80** (**34.20**)
TransH	338.22 (218.40)	0.142 (0.197)	8.20 (12.40)	24.20 (32.80)
TransD	359.86 (236.54)	0.132 (0.175)	8.00 (10.60)	23.60 (31.40)
RotatE	**262.93** (**174.68**)	0.147 (**0.205**)	8.40 (**13.60**)	27.00 (33.80)
TransEMetaJoint	**153.20** (120.80)	0.170 (**0.227**)	9.40 (**15.80**)	**33.80** (38.40)
TransHMetaJoint	166.11 (122.71)	0.161 (0.220)	8.60 (13.40)	32.40 (**39.00**)
TransDMetaJoint	180.30 (138.90)	0.155 (0.203)	8.40 (12.00)	30.40 (36.20)
RotatEMetaJoint	157.73 (**110.17**)	**0.173** (0.226)	**10.80** (**15.80**)	30.60 (36.80)

(B)

Model	Attribute Prediction			
	MR	MRR	Hits@1(%)	Hits@10(%)
TransE	166.69 (87.21)	**0.143** (**0.232**)	4.80 (**11.20**)	**35.00** (**49.00**)
TransH	**160.88** (**84.21**)	0.132 (0.208)	4.00 (8.60)	34.40 (47.80)
TransD	179.94 (96.90)	0.135 (0.215)	**5.00** (10.60)	31.80 (46.40)
RotatE	165.45 (98.74)	0.113 (0.178)	4.20 (8.20)	25.40 (37.60)
TransEMetaJoint	91.62 (54.28)	0.214 (**0.300**)	11.00 (**17.60**)	43.40 (54.28)
TransHMetaJoint	**85.16** (**47.93**)	**0.218** (0.293)	**11.60** (16.40)	**45.00** (**57.40**)
TransDMetaJoint	86.07 (49.12)	0.202(0.291)	9.60(16.80)	43.40 (55.80)
RotatEMetaJoint	126.51 (80.22)	0.182(0.259)	9.20(14.60)	36.60 (47.00)
Model	**Source Prediction**			
	MR	MRR	Hits@1(%)	Hits@10(%)
TransE	**140.46** (**92.84**)	**0.238** (**0.298**)	**13.20** (**19.40**)	**45.20** (**51.00**)
TransH	141.23 (97.23)	0.222 (0.282)	11.40 (17.40)	44.00 (50.40)
TransD	188.07 (126.02)	0.208 (0.266)	11.60 (16.80)	39.40 (46.20)
RotatE	141.35 (103.27)	0.194 (0.246)	10.80 (15.60)	37.40 (43.00)
TransEMetaJoint	87.54 (67.81)	0.264 (0.326)	15.00 (21.60)	50.60 (55.40)
TransHMetaJoint	88.27 (**66.51**)	**0.273** (**0.335**)	**16.00** (**22.20**)	**52.00** (**57.80**)
TransDMetaJoint	**86.63** (68.27)	0.249 (0.305)	14.20 (19.80)	46.60 (53.20)
RotatEMetaJoint	98.35 (79.60)	0.242 (0.310)	13.80 (22.40)	45.60 (49.80)

learning generally. The improvement on Hits@10 is more obvious for attribute prediction than for source prediction.

Fig. 5 shows the exact-match based evaluation results of TransHMetaJoint model on the validation set with different values of β in Equation 7. When $\beta = 0$, the model becomes the TransHMeta model. In most cases, jointly embedding concept-attribute collocations improves both attribute prediction and source prediction. When $\beta = 1$, the model obtains the best performance. When $\beta = 2$, the performance decreases. This indicates that both the metaphor triples, which reflect the metaphor structures, and the collocations, which enhance concept-attribute associations,

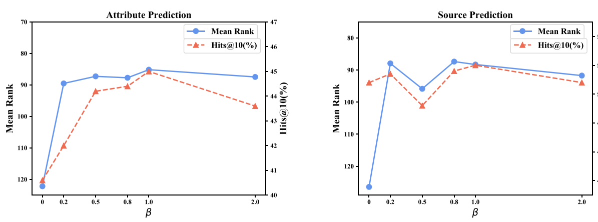


(a) Attribute prediction. (b) Source prediction.

Fig. 4. The comparison of TransH, TransHMeta, TransHJoint and TransHMetaJoint.

TABLE VII
EXACT-MATCH HITS@10(%) ON SOURCE PREDICTION BY ATTRIBUTE CATEGORY ON EKG. (M. STANDS FOR MANY)

	Source Prediction			
	1-to-1	1-to-m	m-to-1	m-to-m
Ratio	79%	5%	12%	4%
TransE	28.18	32.50	55.10	50.79
TransEMetaJoint	30.90	35.00	75.51	52.38
TransH	28.18	25.00	60.20	47.52
TransHMetaJoint	**31.82**	**40.00**	**78.57**	**52.38**
TransD	23.64	22.50	47.96	45.63
TransDMetaJoint	25.45	37.50	70.41	49.21
RotatE	24.54	30.00	44.90	41.27
RotatEMetaJoint	26.46	35.00	61.22	50.00



(a) Attribute prediction. (b) Source prediction.

Fig. 5. The Sensitivity of the parameter β in Equation 7.

are important for representation learning based metaphor processing.

4) Comparisons With Previous Methods: No previous work models metaphor interpretation and generation as KG completion. We conduct comparisons with approaches based on word association and salience measures. The baselines for attribute prediction include

- *Baseline1 [40]:* It ranks attributes using a metric based on the saliences of an attribute to the source and the target. We measure the salience using the concept-attribute collocation data.
- *Baseline2 [42]:* It exploits pre-trained distributed semantic embeddings of concepts and attributes to measure word associations for metaphor interpretation. We implemented it using the Tencent pre-trained word embeddings [55].

The baseline for source prediction is

- *Baseline3 [43]:* It is based on salience imbalance and depends on a dictionary consisting of entities and attributes. We implemented this method using the concept-attribute collocation data.

We use the TransHMetaJoint model and the TransH model trained on EKG for comparison.

Dataset: Notice that some baselines require the target and the (candidate) source should have common attributes. For fair comparison, we use a subset of the test dataset for evaluation. In this subset, for every correct metaphor triple (t, a, s), both (t, a) and (s, a) pairs could be covered by the concept-attribute collocation data. The subset consists of 256 triples.

Metrics: We conducted human evaluation, since metaphor interpretation and generation may have multiple possible answers. We evaluated the top 5 predictions of each system and use precision@5 (p@5) as a metric. All systems' predictions were anonymized and merged together. If one recovered triple is understandable to a labeler, it is viewed as a correct prediction. Two labelers evaluated all the predictions independently and the average performance of each system against two labelers would be reported.

Results: Table IX shows the results. The predictions of TransHMetaJoint are more accurate compared with TransH and the baselines. This is reasonable since our method not only exploits word association knowledge like the baselines but also incorporates supervision from known metaphors during representation learning. Moreover, our method does not require the target and source have common attributes since it depends on distributed representations. So it can produce more interesting and diverse predictions.

5) Qualitative Analysis: Table VIII shows some top predictions of the baselines and TransHMetaJoint.

Qualitative comparisons: We can see a baseline uses *various* and *practicable* to explain the metaphor relation between *knowledge* and *lamp*, because these attributes are shared but the interpretations are not so proper. Similarly, a baseline predicts sources that have a similar meaning to the target, e.g., *tornado* vs. *wind*, and uses *hand* as a source to *mother*, but people rarely create such metaphors. In contrast, TransMetaJoint can give more reasonable predictions.

Understanding the predictions with function scores: As shown in the last three columns in Table VIII, TransHMetaJoint combines three scores together to determine the predictions, integrating different types of evidence and explaining the predictions. We can see that the top predictions can focus on different scores. Notice that a target can borrow an attribute from different sources and a target-source pair can have multiple reasonable shared attributes. Combining multiple scores allow our method to handle such conditions better compared with previous translation-based models, because multiple reasonable sources or attributes are not inappropriately forced to have the same representations.

Can our model distinguish sources and targets?: Table X shows that the function scores change a lot if we swap the target and source in correct metaphor triples. Especially, the $-d(t, a, s)$ scores drop sharply. The same concept-attribute pair can have notable differences in g_1 and g_2 scores.

Table XI shows the average $g_1(c, a)$ and $g_2(c, a)$ scores over associated attributes for some abstract and concrete concepts. In general, the abstract concepts have larger values of g_1 and lower values of g_2, indicating that such concepts are often used as a target and rarely used as a source. In contrast, some concrete concepts may have a larger value of g_2 like *soldier*, which is often used as a source. Some concrete concepts have close values of g_1 and g_2 like *sun*, indicating that they can be used as either a target or a source.

These observations indicate that our method can distinguish targets and sources, and capture the abstractness of concepts.

TABLE VIII
EXAMPLES OF PREDICTIONS OF BASELINES AND TRANSHMETAJOINT ON ATTRIBUTE AND SOURCE PREDICTION. THE THREE SCORES IN EQUATION 1 ARE ALSO SHOWN

Target and Source	**Attribute Prediction**		**Scores in TransHMetaJoint**		
	Baseline [40]	TransHMetaJoint	$-d(t,a,s)$	$g_1(t,a)$	$g_2(s,a)$
< 夏天 (summer), ? , 孩子 (baby)>	可爱 (lovely)	喜怒无常 (temperamental)	-11.53	2.74	1.06
	快乐 (happy)	可爱 (lovely)	-17.06	2.02	5.59
	美丽 (beautiful)	淘气 (mischievous)	-16.77	0.31	6.68
< 知识 (knowledge), ? , 灯 (lamp)>	照亮 (light)	照亮 (light)	-14.68	3.32	6.99
	各种各样 (various)	指引 (guide)	-17.15	2.62	5.87
	实用 (practicable)	照耀 (enlighten)	-15.11	2.81	3.46
Target and Attribute	**Source Prediction**		**Scores in TransHMetaJoint**		
	Baseline [43]	TransHMetaJoint	$-d(t,a,s)$	$g_1(t,a)$	$g_2(s,a)$
< 母亲 (mother), 勤劳 (hardworking), ?>	双手 (hands)	花匠 (floriculturist)	-15.83	3.09	11.45
	手 (hand)	老黄牛 (old cattle)	-17.19	3.09	8.54
	蜜蜂 (bee)	蜜蜂 (bee)	-15.87	3.09	5.06
< 风 (wind), 刮 (blow), ?>	龙卷风 (tornado)	刀子 (knife)	-12.37	2.14	2.83
	寒风 (cold_wind)	铁 (iron)	-15.25	2.14	2.68
	刀子 (knife)	针 (needle)	-14.21	2.14	1.45

TABLE IX
HUMAN EVALUATIONS ON ATTRIBUTE PREDICTION AND SOURCE PREDICTION WITH PRECISION@5 (P@5) AS THE METRIC

Method	p@5(%)
Attribute Prediction	
Baseline1 [40]	75.21
Baseline2 [42]	63.54
TransH	80.67
TransHMetaJoint	**85.42**
Source Prediction	
Baseline3 [43]	57.81
TransH	76.85
TransHMetaJoint	**81.64**

TABLE X
SWAPPING THE TARGET AND SOURCE CHANGES THE FUNCTION SCORES. THE CORRECT TRIPLES ARE <母亲 (MOTHER), 勤劳 (HARDWORKING), 花匠 (FLORICULTURIST) > AND <知识 (KNOWLEDGE), 照亮(LIGHT), 灯 (LAMP)>

Triple <t, a, s>	$-d(t,a,s)$	$g_1(t,a)$	$g_2(s,a)$
< 母亲, 勤劳, 花匠 >	-15.83	3.09	11.45
< 花匠, 勤劳, 母亲 >	-20.97	3.25	-0.04
< 知识, 照亮, 灯 >	-14.68	3.32	6.99
< 灯, 照亮, 知识 >	-21.99	1.93	0.68

TABLE XI
AVERAGE g_1 AND g_2 SCORES FOR SOME ABSTRACT CONCEPTS AND CONCRETE CONCEPTS

Concept c	**Scores in TransHMetaJoint**	
	$avg(g_1(c,a))$	$avg(g_2(c,a))$
人 (people)	3.91	-0.32
梦想 (dream)	3.03	-1.19
爱 (love)	0.74	-5.87
战士 (soldier)	1.74	4.30
太阳 (sun)	1.61	1.33
花朵 (flower)	1.45	2.66

B. Evaluating Metaphor Detection

We focus on detecting A_is_B style metaphors and view the task as a concept pair classification problem.

1) Experimental Settings: The experimental settings are as follows:

Dataset: Our dataset consists word pairs for identifying metaphorical ones. We sampled word pairs from the human labeled metaphor triples data to construct positive pairs. The negative pairs include two types. The first type of pairs were collected from the sentences that have a comparator but do not have a metaphorical meaning. This type of word pairs have literal meanings. The second type of pairs just have randomly sampled words to form word pairs. Involving this type of pairs in training data could benefit evaluating the robustness of metaphor detection systems. We converted these word pairs into A_is_B form and let the two labelers re-check these expressions. Finally, the dataset for evaluation include 1937 metaphorical pairs and 1936 non-metaphorical pairs. The dataset is divided into training set (60%), validation set (20%) and test set (20%).

Baselines and comparisons: Our purpose is to verify the effectiveness of learned representations for metaphor detection. We compare with two categories of methods.

The first category is relatedness based methods: [31], [42], which use embeddings for concept representation and identify metaphors by measuring concept relatedness. We use the following method as a baseline.

- *SimRel [42]:* This method determines a nominal metaphor, only if the target and source share little information between them, i.e., a relatedness score is less than a threshold and the concept-pair is not a hyponym/hypernym relation. The relatedness is computed based on cosine similarity between word embeddings. The general purpose Tencent pre-trained word embeddings [55] are used.

To compare with this line of work, we use the TransMeta scoring function for metaphor detection.

- *TransMeta:* The method follows similar procedure to SimRel but measures relatedness between concepts based on KG embeddings. For each candidate word pair (t, s), we use the top 1 predicted attribute a by TransHMetaJoint to form a triple (t, a, s), and we use the score computed by the TransMeta function (Equation 1) to measure the possibility

TABLE XII
EVALUATION RESULTS ON METAPHOR DETECTION. [TENCENT,OURS] INDICATES THE CONCATENATION OF TENCENT AND OUR EMBEDDINGS

Model	Embeddings	P	R	F_1
SimRel [42]	Tencent	0.55	0.68	0.61
TransMeta	Ours	0.73	0.72	0.72
AttribSemantic [36]	Distributional	0.71	0.70	0.70
MLP	Tencent	0.75	0.79	0.77
	Ours	0.76	0.72	0.74
	[Tencent, Ours]	**0.79**	**0.81**	**0.80**
SSN [32]	Tencent	0.74	0.76	0.75
	Ours	0.76	0.70	0.73
	[Tencent, Ours]	0.79	0.79	0.79

that the two concepts can form a metaphor. The higher the score is, the more likely t and s form a metaphor.

Both SimRel and TransMeta need a threshold. A predicted score y by TransMeta is normalized to [0,1] based on $\frac{y-\min}{\max-\min}$, where max and min are the maximum and minimum triple scores on the training set. For both SimRel and TransMeta, we use 0.01 as the step length for tuning the threshold and choose the best one based on the F1 score on the validation data.

The second category of methods are based on supervised classifiers: The keys are the model architecture and feature representation. We adopt the following architectures in our experiments.

- *SSN:* Semantic Similarity Network (SSN) [32] is a state-of-the-art neural model for metaphor detection.
- *MLP:* We use a simple two-layer multi-layer perceptron (MLP) as the classifier. The two hidden layers of the MLP have 256 and 128 neurons respectively.

The classifier input is the concatenation of the embeddings of a target-source candidate pair. For comparison, we use the Tencent embeddings and our embeddings for feature representation, respectively. We use the Adadelta algorithm [54] as the optimizer. All of the word embeddings are fixed during classifier training.

We also compare with the following method.

- *AttribSemantic:* [36]: This method constructs distributional concept representations in the attribute space, which depends on an English property norm dataset. To implement this method for Chinese metaphor detection, we use the concept-attribute collocation data to simulate the property norm dataset to learn concept representations and use SVM to perform the classification.

2) Results: The results are shown in Table XII. For relatedness based methods, TransMeta outperforms SimRel with a large margin. This is reasonable, because TransMeta exploits attribute-dependent concept relations, which are embedded in our embeddings. TransMeta even achieves similar performance to some supervised models.

For supervised classifiers, MLP and SSN with Tencent and our embeddings achieve better performance than the distributional representation based baseline AttribSemantic. Because Tencent word embeddings are learned on very large text corpora, it is easy to explain its superior performance compared with the baseline. Our method and AttribSemantic both exploit the concept-attribute collocation data, while our method also learns from known metaphors through metaphor KG embedding, which leads to the improvement.

Using our embeddings performs slightly worse than using Tencent embeddings in both architectures. We observe that the large pre-trained embeddings are good at capturing semantic relationship between concepts that often co-occur. In other words, using general purpose embeddings helps identifying concept pairs with literal meanings. In contrast, our embeddings embed metaphor relation information and lead to a higher precision. These two types of embeddings should complement each other. We combine our embeddings with Tencent embeddings by concatenating them together for classification, and the combination can lead to the best performance.

VIII. CONCLUSION

This paper has presented a novel method for metaphor processing based on KG embedding. We view a metaphor as an attribute-dependent concept mapping and propose a new function for scoring metaphor relations and a novel joint model for simultaneously embedding metaphor triples and concept-attribute collocations. The main metaphor processing tasks including metaphor detection, interpretation and generation could be conducted in the same framework.

Our research shows that we can collect the metaphor triple resource based on a simile recognizer. The proposed new scoring function and the joint embedding models can effectively improve the performance on metaphor interpretation and generation compared with previous KG embedding methods and strong baselines. The learned representations can also benefit nominal metaphor detection.

In future, we plan to further expand the metaphor KG for better representation learning. This work mainly deals with nominal metaphors. We should investigate methods to incorporate the learned embeddings for other types of metaphors such as token level contextual metaphor detection.

REFERENCES

[1] G. Lakoff and M. Johnson, *Metaphors We Live By*. Chicago, IL, USA: University of Chicago Press, 1980.
[2] D. Gentner, B. Bowdle, P. Wolff, and C. Boronat, "Metaphor is like analogy," *Analogical Mind: Perspectives Cogn. Sci.*, pp. 199–253, 2001.
[3] A. Ortony, "Beyond literal similarity," *Psychol. Rev.*, vol. 86, no. 3, pp. 161–180, 1979.
[4] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.
[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.
[6] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI*, 2014, pp. 1112–1119.
[7] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI*, 2015, pp. 2181–2187.
[8] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Nat. Lang. Process.*, 2015, pp. 687–696.

[9] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *Proc. 7th Int. Conf. Learn. Representations*, 2019.
[10] T. Veale and Y. Hao, "Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language," in *Proc. AAAI*, pp. 1471–1476, 2007.
[11] V. Niculae and V. Yaneva, "Computational considerations of comparisons and similes," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguist. Student Res. Workshop*, 2013, pp. 89–95.
[12] V. Niculae, "Comparison pattern matching and creative simile recognition," in *Proc. Joint Symp. Semantic Process. Textual Inference Struct. Corporation*, 2013, pp. 110–114.
[13] H. Li, K. Q. Zhu, and H. Wang, "Data-driven metaphor recognition and explanation," *Trans. Assoc. Comput. Linguist.*, vol. 1, pp. 379–390, 2013.
[14] V. Niculae and C. Danescu-Niculescu-Mizil, "Brighter than gold: Figurative language in user generated comparisons," in *Proc. Conf. Empirical Methods Nat. Lang. Process*. 2014, pp. 2008–2018.
[15] L. Liu, X. Hu, W. Song, R. Fu, T. Liu, and G. Hu, "Neural multitask learning for simile recognition," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2018, pp. 1543–1553.
[16] J. Zeng, L. Song, J. Su, J. Xie, W. Song, and J. Luo, "Neural simile recognition with cyclic multitask learning and local attention," in *Proc. AAAI*, 2020, pp. 9515–9522.
[17] B. Li, L.-l. Yu, M. Shi, and W.-g. Qu, "Computation of Chinese simile with "xiang"," *J. Chin. Inf. Process.*, vol. 22, no. 6, pp. 27–32, 2008.
[18] J. E. Lakoff, G., and A. Schwartz, "Master metaphor list second draft copy," Cogn. Linguist. Group, Univ. California Berkeley, Tech. Rep. CSRP-01-05. 1991.
[19] J. H. Martin, "Metabank: A knowledge-base of metaphoric language conventions," *Comput. Intell.*, vol. 10, no. 2, pp. 134–149, 1994.
[20] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proc. EMNLP*, 2011, pp. 1535–1545.
[21] E. Shutova, L. Sun, and A. Korhonen, "Metaphor identification using verb and noun clustering," in *Proc. 23rd Int. Conf. Comput. Linguist.*, 2010, pp. 1002–1010.
[22] P. D. Turney, Y. Neuman, D. Assaf, and Y. Cohen, "Literal and metaphorical sense identification through concrete and abstract context," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2011, pp. 680–690.
[23] E. Shutova and L. Sun, "Unsupervised metaphor identification using hierarchical graph factorization clustering," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, 2013, pp. 978–988.
[24] M. Mohler, D. Bracewell, M. Tomlinson, and D. Hinote, "Semantic signatures for example-based linguistic metaphor detection," in *Proc. 1st Workshop Metaphor NLP*, 2013, pp. 27–35.
[25] Y. Neuman *et al.*, "Metaphor identification in large texts corpora," *PLOS One*, vol. 8, no. 4, 2013, Art. no. e62343.
[26] D. Fass, "met*: A method for discriminating metonymy and metaphor by computer," *Comput. Linguist.*, vol. 17, no. 1, pp. 49–90, 1991.
[27] M. Gedigian, J. Bryant, S. Narayanan, and B. Ciric, "Catching metaphors," in *Proc. 3rd Workshop Scalable Nat. Lang. Understanding.*, 2006, pp. 41–48.
[28] H. Jang, S. Moon, Y. Jo, and C. Rose, "Metaphor detection in discourse," in *Proc. 16th Annu. Meeting Special Int. Group Discourse Dialogue*, 2015, pp. 384–392.
[29] E. Shutova, D. Kiela, and J. Maillard, "Black holes and white rabbits: Metaphor identification with visual features," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, 2016, pp. 160–170.
[30] Y. Tsvetkov, L. Boytsov, A. Gershman, E. Nyberg, and C. Dyer, "Metaphor detection with cross-lingual model transfer," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguist.*, 2014, vol. 1, pp. 248–258.
[31] R. Mao, C. Lin, and F. Guerin, "Word embedding and wordnet based metaphor identification and interpretation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, 2018, pp. 1222–1231.
[32] M. Rei, L. Bulat, D. Kiela, and E. Shutova, "Grasping the finer point: A supervised similarity network for metaphor detection," 2017, *arXiv:1709.00575*.
[33] G. Steen, *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*. Amsterdam, The Netherlands: John Benjamins Publishing Company, 2010, vol. 14.
[34] E.-L. Do Dinh and I. Gurevych, "Token-level metaphor detection using neural networks," in *Proc. 4th Workshop Metaphor NLP*, 2016, pp. 28–33.
[35] G. Gao, E. Choi, Y. Choi, and L. Zettlemoyer, "Neural metaphor detection in context," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2018, pp. 607–613.
[36] L. Bulat, S. Clark, and E. Shutova, "Modelling metaphor with attribute-based semantics," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist.*, 2017, pp. 523–528.
[37] J. H. Martin, *A Computational Model of Metaphor Interpretation*. New York, NY, USA: Academic Press Professional, Inc., 1990.
[38] E. Shutova, "Automatic metaphor interpretation as a paraphrasing task," in *Proc. Hum. Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2010, pp. 1029–1037.
[39] K. Abe, K. Sakamoto, and M. Nakagawa, "A computational model of the metaphor generation process," in *Proc. Annu. Meeting Cogn. Sci. Soc.*, 2006, pp. 937–942.
[40] P. Xiao, K. Alnajjar, M. Granroth-Wilding, K. Agres, and H. Toivonen, "Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations," in *Proc. 7th Int. Conf. Comput. Creativity.* Paris, France, 2016, pp. 230–237.
[41] A. Qadir, E. Riloff, and M. A. Walker, "Automatically inferring implicit properties in similes," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, 2016, pp. 1223–1232.
[42] C. Su, S. Huang, and Y. Chen, "Automatic detection and interpretation of nominal metaphor based on the theory of meaning," *Neurocomputing*, vol. 219, pp. 300–311, 2017.
[43] R. Rzepka, P. Dybala, K. Sayama, and K. Araki, "Semantic clues for novel metaphor generator," in *Proc. 2nd Int. Workshop Comput. Creativity, Concept Invention, Gen. Intell., C3GI*, 2013, pp. 937–942.
[44] N. Cila, P. Hekkert, and V. Visch, "Source selection in product metaphor generation: The effects of salience and relatedness," *Int. J. Des.*, vol. 8, no. 1, pp. 15–28, 2014.
[45] D. Yarowsky, "One sense per collocation," in *Proc. Workshop Hum. Lang. Technol.*, 1993, pp. 266–271.
[46] L. K. Tyler, H. Moss, M. Durrant-Peatfield, and J. Levy, "Conceptual structure and the structure of concepts: A distributed account of category-specific deficits," *Brain Lang.*, vol. 75, no. 2, pp. 195–231, 2000.
[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre- training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Computat. Linguistics, Human Lang. Technol.*, 2019, pp. 4171–4186.
[48] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[50] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[51] J. Mei,同义词词林 *(Synonymy Thesaurus of Chin. Words)*. 商務 印書館; 上海, 1984, vol. 1983.
[52] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[53] X. Han *et al.*, "Openke: An open toolkit for knowledge embedding," in *Proc. EMNLP*, 2018, pp. 139–144.
[54] M. D. Zeiler, "Adadelta: An adaptive learning rate method," 2012, *arXiv:1212.5701*.
[55] Y. Song, S. Shi, J. Li, and H. Zhang, "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol., Vol. 2 (Short Papers)*. New Orleans, Louisiana: Assoc. Comput. Linguist., Jun. 2018, pp. 175–180.

Wei Song received the Ph.D. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in 2013. He is now with the College of Information Engineering and Academy for Multidisciplinary Studies, Capital Normal University, Beijing, China.

He is currently an Associate Professor with Capital Normal University. His research interests include natural language processing and its use in education.

Jingjin Guo received the master's degree from the College of Information Engineering, Capital Normal University in 2020. She is currently working toward the Central University of Finance and Economics. She is currently with the Central University of Finance and Economics, and previously was at the College of Information Engineering, Capital Normal University, Beijing, China.

Her research focuses on natural language processing.

Ruiji Fu received the Ph.D. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in 2014. He is now with the State Key Laboratory of Cognitive Intelligence, iFLYTEK Research & iFLYTEK AI Research (Hebei), Langfang, China.

He is currently a Deputy Dean with iFLYTEK Research and a Vice-Director with the Joint Laboratory of HIT and iFLYTEK Research, in charge of cognitive intelligence technology research and development in education. His research interests include automated essay scoring, adaptive learning, text mining, and knowledge graphs.

Ting Liu received the Ph.D. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in 1998. He is now with the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin, China.

He is currently a Full Professor with the Harbin Institute of Technology. His research interests include information retrieval, natural language processing, and social media analysis.

Lizhen Liu received the Ph.D. degree in computer application from the Beijing Institute of Technology, Beijing, China. She is with the College of Information Engineering and Academy for Multidisciplinary Studies, Capital Normal University, Beijing, China.

She is currently a Full Professor with Capital Normal University. Her research interests include text mining, recommender systems, and natural language processing.