View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Southern Queensland ePrints

Noname manuscript No.
(will be inserted by the editor)

A Hybrid Representation Based Simile Component Extraction

Da Ren · Pengfei Zhang · Qing Li · Xiaohui Tao · Junying Chen · Yi Cai

Received: date / Accepted: date

Abstract Simile, a special type of metaphor, can help people to express their ideas more clearly. Simile component extraction is to extract tenors and vehicles from sentences. This task has a realistic significance since it is useful for building cognitive knowledge base. With the development of deep neural networks, researchers begin to apply neural models to component extraction. Simile components should be in cross-domain. According to our observations, words in cross-domain always have a different concept. Thus, concept is important when identifying whether two words are simile components or not. However, existing models do not integrate concept into their models. It is difficult for these models to identify the concept of a word. What's more, corpus about simile component extraction is limited. There are a number of rare words or unseen words and the representations of these words are always nor proper enough. Exiting models can hardly extract simile components accurately when there are low frequency words in sentences. To solve these problems, we propose a Hybrid Representation based Component Extraction (HRCE) model. Each word in HRCE is represented in three different levels: word level, concept level and character level. Concept representations (representations in concept level) can help HRCE to identify the words in cross-domain more accurately. Moreover, with the help of character representations (representations in character levels), HRCE can represent the meaning of a word more properly since words are consisted of characters and these characters can partly represent the meaning of words. We conduct experiments to compare the performance between HRCE and existing models. The experiment results show that HRCE significantly outperforms current models.

Keywords Simile component · Concept · Character

Da Ren
South China University of Technology
E-mail: edwardyam@outlook.com

Pengfei Zhang
South China University of Technology
E-mail: sezhangpengfei@mail.scut.edu.cn

Qing Li
The Hong Kong Polytechnic University
E-mail: csqli@comp.polyu.edu.hk

Xiaohui Tao
The University of Southern Queensland
E-mail: xtao@usq.edu.au

Junying Chen
South China University of Technology
E-mail: jychense@scut.edu.cn

✉ Yi Cai
South China University of Technology
E-mail: ycai@scut.edu.cn

1 Introduction

Metaphor is commonly used in human conversations and literatures. It is as a matter of cross-domain mappings in conceptual structure which are expressed in language [21]. Metaphor can help people to express their ideas more accurately. Moreover, people can understand the thought of other people more easily with the help of metaphor. Interpreting metaphors is an integral and inescapable process in human understanding of natural language [5]. Therefore, there are growing researches about metaphor analyses [15, 21, 30].

Researchers begin to analyze metaphor by recognizing simile [21]. Simile is a special type of metaphor. There are explicit markers (e.g. "as", "like") which are also known as comparators [12] in simile sentences. With the help of comparators, it is more clear for people to recognize whether a sentence is a simile sentence or not. Simile recognition is important for machines to understand natural language.

With the development of deep neural networks (DNNs), DNNs begin to be applied to simile recognition [21]. These

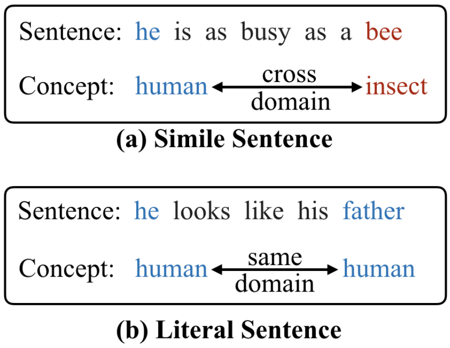

Fig. 1 Sentences with Comparators

experiment results show that DNNs can have a better performance than rule-based and feature-based methods [21]. There are two tasks in simile recognition: simile sentence classification and simile component extraction. Simile sentence classification is to identify whether a sentence is a simile sentence, while simile component extraction is to extract simile components from sentences. In this work, we focus on the simile component extraction task. Simile component extraction, which is potentially useful for building cognitive knowledge base, has a realistic significance [21].

Currently, it is still challenging to extract simile components from sentences accurately. A sentence with comparators may not be a simile sentence. It can be regarded as a simile sentence when there is a cross-domain mapping in the sentence. We show two examples in Fig. 1. Both of the sentences contain comparators ("as" and "like"). In Fig. 1 (a), the concept of "he" is "human" while the concept of "bee" is "insect". There is a cross-domain mapping in the concepts of these two words. Thus, "he" and "insect" are simile components in this sentence. However, in Fig. 1 (b), both the concept of "he" and "father" are "human". These two words are in the same concept domain. This sentence is a literal sentence (not a simile sentence). Thus, these two words are not simile components. The concept of a word is important when identifying whether two words are in the same domain. However, existing models [21] use word embeddings to represent words. These models can hardly capture the concept of a word. We consider that if a model can represent a word with its concept information, this model will extract simile components more accurately.

Existing corpus about simile component extraction is limited. A number of words seldom appear in the corpus. Moreover, models may even meet unseen words when testing. The word embeddings of rare words or unseen words can not be trained sufficiently. These word embeddings can hardly present their meanings. Therefore, it is difficult for existing models to extract simile components which are consisted of rare words or unseen words. A word is consisted of characters. These characters can partly indicate the meaning of the word. If a model can use character representations to enrich the representation of each word, these representations will capture the meaning of this word more accurately.

To solve these problems, we propose a Hybrid Representation based Component Extraction (HRCE) model. In HRCE, each word is represented in three different levels: word level, concept level and character level. The representations in concept level can help HRCE to identify whether two words are in the same domain more accurately. The representations in character level can help HRCE to capture the meaning of rare words or unseen words. In this way, HRCE can extract simile components more accurately.

Our contributions can be summarized as follows.

- We integrate concept representations (representations in concept level) to enrich the performance of word representations. We propose a Hybrid Representation based Component Extraction (HRCE) model to extract simile components in sentences based on the representations. In this way, HRCE can be aware of the concept of each word. Thus, it can identify whether two words are in the same domain more accurately.
- According to our observations, existing extraction models can hardly extract simile components which are consisted of rare words or unseen words correctly. Thus, we further integrate character representations (representations in character level) in the word representations. Character representations can help HRCE to represent rare words and unseen words more properly. In this way, HRCE can extract simile components more accurately.
- We conduct experiments to compare the performance between HRCE and current extraction models. Our experiment results show that HRCE significantly outperforms existing models.

This paper is structured as follows. In section 2, we introduce the related works of metaphor and simile. We elaborate the details of our models in section 3. We illustrate our experiments in section 4. Finally, we draw our conclusions in section 5.

2 Related Work

2.1 Metaphor Analysis

There are growing researches about metaphor analyses in recent years [21]. Metaphor analysis often contains three tasks [21]: metaphor recognition, metaphor explanation and metaphor generation [15, 30]. Simile is a special type of metaphor which contains comparators. Comparators make the metaphorical parts more easy to be located [21].

In the early stage, researchers use rule-based methods or feature-based methods to recognize simile sentences [18, 24, 35]. Niculae and Danescu-Niculescu-Mizil [24] use a series of linguistic cues as features to distinguish a comparison from figurative or literal in product reviews. Veale and Hao [35] describe how the category-defining knowledge required by metaphor can be acquired from exposure to explicit similes. They demonstrate that this knowledge offers a richer and more diagnostic picture of category structure than that acquired from alternate sources [35]. Li et al. [18] propose a feature-based method for Chinese simile recognition and evaluate their methods on a small dataset. Veale [34] build a lexical stereotype model from similes. They construct the stereotype-based lexicon in two stages [34]. For the first layer, a large collection of stereotypical descriptions is harvested from the web [34]. For the second layer, they link these common-sense qualities in a support graph [34]. Qadir et al. [26] use lexical features, semantic features and sentiment features to infer the affective polarity of simile in twitters to build classifiers. Qadir et al. [27] infer implicit properties by using syntactic structure, dictionary definitions, statistical co-occurrence and word embeddings.

Recently, researchers begin to use neural networks in simile recognition. Liu et al. [21] build a Chinese dataset of simile recognition which is consisted of sentences which containing a comparator. Moreover, inspired by the thought of multitask learning [4, 23], Liu et al. [21] further propose a neural multitask learning framework jointly optimizing three tasks: simile sentence classification, simile component extraction and language modeling. Their experimental results show that the neural network based approaches can outperform all rule-based and feature-based baselines [21]. What's more, both simile sentence classification and simile component extraction can benefit from multitask learning [21]. However, they do not integrate concept into their models to improve the performance of their models.

2.2 Sequence Labeling

In this article, we focus on simile component extraction. Simile component extraction is considered as a sequence labeling task [21]. In recent years, researchers propose a number of model which are successfully applied to various sequence labeling tasks [9, 10, 36] (e.g. chunking, part-of-speech (POS) tagging and named entity recognition (NER)).

Huang et al. [14] use BiLSTM for word-level representations and CRF for jointly label decoding. They combine their neural network model with handcrafted features to improve their performance. Therefore, their model is not an end-to-end system. Lample et al. [16] further introduce a hierarchy structure by incorporating BiLSTM-based character embeddings. Their models rely on two sources of information: character-based word representations learned from the supervised corpus and unsupervised word representations learned from unannotated corpora. Liu et al. [20] propose Knowledge Augmented Language Model (KALM). It is a language model with access to information available in a knowledge base. In addition to improving language modeling performance, KALM learns to recognize named entities in an entirely unsupervised way by using entity type information latent in the model. Inspired by the thought of multi-task learning [6], Rei [28] proposes a labeling framework with a secondary training objective: learning to predict surrounding words for every word in the dataset. This objective incentivises the system to learn general-purpose patterns of semantic and syntactic composition. Rei [28] considers that these patterns are useful for improving accuracy on sequence labeling tasks. Strzyz et al. [32] use sequence labeling for constituency [11] and dependency parsing [33] combined with multi-task learning to learn across syntactic representations. They show that adding a parsing paradigm as an auxiliary loss consistently improves the performance on the other paradigm. What's more, they further demonstrate that a single multi-task learning model following their strategy can robustly produce constituency and dependency trees. This model obtains a performance and speed comparable with previous sequence labeling models for constituency and dependency parsing [32]. Chen and Moschitti [7] propose an approach for transferring the knowledge of a neural model for sequence labeling, learned from the source domain, to a new model trained on a target domain, where new label categories appear. Wiseman and Stratos [37] show that they can perform accurate sequence labeling by explicitly copying labels from retrieved neighbors. What's more, they can achieve impressive performance in zero-shot sequence-labeling tasks since this copying is label-agnostic. Alzaidy et al. [3] address the keyphrase extraction problem as sequence labeling. They propose a model that jointly exploits the complementary strengths of Conditional Random Fields that capture label dependencies through a transition parameter matrix consisting of the transition probabilities from one label to the neighboring label, and Bidirectional Long Short Term Memory networks that capture hidden semantics in text through the long distance dependencies [3].

A number of works move away from the original "one word, one embedding" paradigm to investigate contextualized embedding models [1, 2, 25]. Akbik et al. [1] leverage the internal states of a trained character language model to produce a novel type of word embedding. These embeddings are contextualized by their surrounding text. Thus, the same word will have different embeddings depending on its contextual use. Peters et al. [25] consider that the recurrent network that operates on word-level representations to produce context sensitive representations is trained on relatively little labeled data. Thus, they demonstrate a general semi-supervised approach for adding pretrained context embeddings from bidirectional language models to NLP systems and apply it to sequence labeling tasks [25]. Akbik et al. [2] consider that purely character-based approaches struggle to produce meaningful embeddings if a rare string is used in a underspecified context. To solve this problem, they dynamically aggregate contextualized embeddings of each unique string that they encounter. Then, they use a pooling operation to distill a global word representation from all contextualized instances [2].

3 Model

Existing models use word embeddings to be the representations of each word. These representations will lead two problems. 1) Simile components should be in cross-domain. Concepts can help extraction models to identify whether two words are in the same domain or not. However, word embeddings do not represent the concept of each word. Thus, existing models can hardly identify extract simile components accurately since teh lost of concept information. 2) The embeddings of rare words or unseen words can not be trained sufficiently. These insufficiently-trained embeddings always fail to represent a word properly. Therefore, it is difficult for existing models to label these words incorrectly.

To solve these problems, we propose a Hybrid Representation based Component Extraction (HRCE) model. For the problem of missing concept, we integrate the concept of each word into HRCE. According to our observations, words in the same domain always have similar concepts. Therefore, HRCE can identify the words in cross-domain more accurately with the concept information.

For the insufficiently-trained embeddings of rare word or unseen words, we integrate character representations into HRCE. Words are consisted of character sequences. These sequences can partly indicate the meaning of corresponding words. With the help of character representations, HRCE can capture the meaning of a word more properly. It can help HRCE to extract simile components more accurately.

The general structure of HRCE is shown in Fig. 2. HRCE represents a word in three levels: **word level**, **concept level** and **character level**. The representations in word level are word embeddings which are denoted as $e_w(w_t)$ in Fig. 2 (w_t is the t-th word in the sentence). The representations in concept level and character level are denoted as $r_{con.}(w_t)$ and $r_{char}(w_t)$ respectively.

3.1 Task Description

Simile, a special type of metaphor, is as a matter of cross-domain mappings in conceptual structure [21]. It is a mapping between *target* and *resource*. The target, which is also known as *tenor* is the subject to which attributes are ascribed [21]. The resource, which is also known as *vehicle* is the object whose attributes are borrowed [21]. There are explicit markers (e.g. “as”, “like”) in simile sentences. These markers are always denoted as *comparators* [12]. The comparators make people more easy to locate the tenors and the vehicles. However, a sentence with comparators may not be a simile sentence, since there may be no cross-domain concept mappings in the sentences.

Simile component extraction is one of the tasks in simile recognition. Given a sentence $S = (w_1, w_2, ..., w_{|S|})$, w_t is the t-th word and $|S|$ is the size of S. If S is a simile sentence, an extraction model needs to extract the tenors and vehicles from S. It should be noted that both tenors and vehicles may be phrases (consisted of more than a word).

The identifying of simile components is same with sequence labeling tasks (e.g. NER, chunking). Therefore, following the previous work [21], we also consider this task as a sequence labeling problem. Our extraction model identifies simile components by giving a label to each word.

3.2 Proposed Model

In HRCE, each word is represented in three different levels: word level, concept level and character level. The representations in word level are word embeddings. Word embeddings are widely used in NLP tasks [17, 19, 29]. They are usually initialized randomly and adjust along with training.

We denote the representations in concept level as **concept representations**. We use CN-Probase[1] to extract the concept of each word. Words in CN-Probase are represented as concept collections. We denote the concept collection of a word w_t as $\hat{C}_t = \{(\hat{c}^t_1, \hat{a}^t_1), (\hat{c}^t_2, \hat{a}^t_2), ..., (\hat{c}^t_{|C|}, \hat{a}^t_{|C|})\}$. $\hat{c}^t_i$ is one of the concepts which w_t may indicate. $\hat{a}^t_i$ is the weight of c^t_i. The higher the weight is, the more likely the concept is indicated. We consider that a concept with higher weight should be considered more importantly. Thus, we use the weighted sum of the concept embeddings of a word to be its concept representation (as shown in Fig. 2). However, the concept weights given by CN-Probase always have a large differences. Therefore, we normalize these weights before using it. The concept representation of w_t is calculated according to Equation 1.

$$r_{con.}(w_t) = \sum_{i=1}^{|\hat{C}_t|} a^t_i e_{con.}(\hat{c}^t_i) \tag{1}$$

where $r_{con.}(w_t)$ is the concept representation of w_t, $e_{con.}(\hat{c}^t_i)$ is the embedding of the concept $\hat{c}^t_i$, a^t_i is the weight of each concept after normalizing and is calculated as follows.

$$a^t_i = \hat{a}^t_i / \sum_{j=1}^{|\hat{C}_t|} \hat{a}^t_j \tag{2}$$

[1] http://kw.fudan.edu.cn/cnprobase/search/

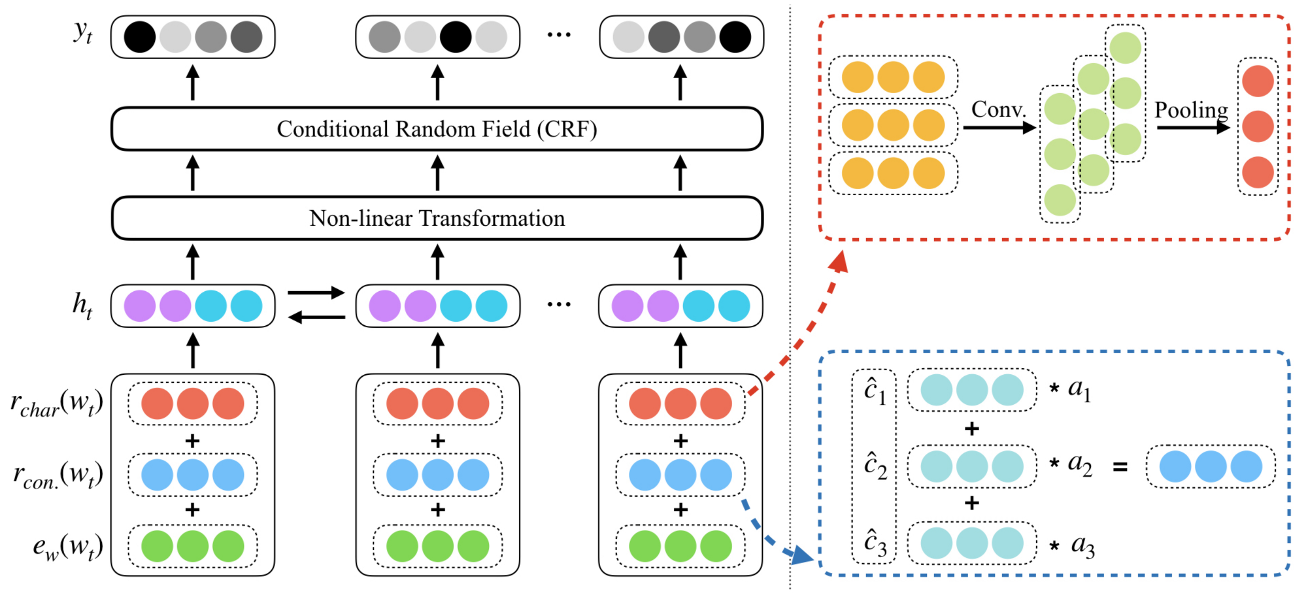

Fig. 2 A Hybrid Representation based Component Extraction (HRCE) Model

We denote the representations in character level as **character representations**. In this work, we use Convolutional Neural Networks (CNNs) to calculate the character representations (as shown in Fig. 2). There are two layers in the CNNs we used: convolutional layer and pooling layer.

Convolutional layers can help HRCE to extract higher level features from the input matrix [22]. The character sequence of the word w_t is denoted as $C_t = (c_0^t, c_1^t, ..., c_{|C|}^t)$. Given a filter $W_f \in R^{l \times m}$, a feature s_i^c is generated according to Equation 3.

$$s_i^c = g_c(W_f \cdot [e_{char}(\hat{c}_i^t) : e_{char}(\hat{c}_{i+l-1}^t + b_f)]) \tag{3}$$

where b_f is a bias term, $e_{char}(\hat{c}_i^t)$ is the embedding of the character $\hat{c}_i^t$, $g_c(\cdot)$ is a ReLU function. This process can be repeated for various filters.

Then, we use pooling layer to further abstract the features generated from convolution layer. In this work, we use max-pooling operation in the pooling layer. Max-pooling is to choose the highest value on each dimension of vector to capture the most important feature. With pooling layers, we can generate a fixed-length vector from feature maps. This vector is denoted as $r_{char}(w_t)$. We use $r_{char}(w_t)$ as the character representation of the word w_t.

To capture the representations in three different levels (word level, concept level and character level), we sum the representations in three different levels as the new representations of words(as shown in Fig. 2). Then, We use Bidirectional Recurrent Neural Networks (BiRNNs) to capture the information of the sentence by looking both the past and the future. It is described in Equation 4.

$$h_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t \tag{4}$$

where $\oplus$ is a concatenation operation, $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are calculated by Equation 5 and 6 respectively.

$$\overrightarrow{h}_t = g_r(\overrightarrow{h}_{t-1}, e(w_t) + r_{con.}(w_t) + r_{char}(w_t)) \tag{5}$$

$$\overleftarrow{h}_t = g_r'(\overleftarrow{h}_{t+1}, e(w_t) + r_{con.}(w_t) + r_{char}(w_t)) \tag{6}$$

where $e(w_t)$ is the embedding of the t-th word, h_t is the hidden state of the t-th timestep, $\overrightarrow{h}_t$ is the t-th forward hidden state while $\overleftarrow{h}_t$ is the t-th backward hidden state, $g_r(\cdot)$ and $g_r'(\cdot)$ are non-linear transformations which are always long-short term memory units (LSTM) [13] or gated recurrent units (GRU) [8]. In this work, we use LSTM which is parameterized as follows.

$$\begin{aligned}
\mathbf{f_t} &= \sigma(\hat{W}_f \mathbf{x_t} + \hat{U}_f \mathbf{\hat{h}_{t-1}} + \hat{b}_f) \\
\mathbf{i_t} &= \sigma(\hat{W}_i \mathbf{x_t} + \hat{U}_i \mathbf{\hat{h}_{t-1}} + \hat{b}_i) \\
\mathbf{o_t} &= \sigma(\hat{W}_o \mathbf{x_t} + \hat{U}_o \mathbf{\hat{h}_{t-1}} + \hat{b}_o) \\
\mathbf{l_t} &= tanh(\hat{W}_c \mathbf{x_t} + \hat{U}_c \mathbf{\hat{h}_{t-1}} + \hat{b}_c) \\
\mathbf{c_t} &= \mathbf{f_t} \circ \mathbf{c_{t-1}} + \mathbf{i_t} \circ \mathbf{l_t} \\
\mathbf{\hat{h}_t} &= \mathbf{o_t} \circ tanh(\mathbf{c_t})
\end{aligned}$$

where $\sigma(\cdot)$ is a sigmoid function, $\circ$ is an element wise multiplication, $\mathbf{x_t}$ and $\mathbf{\hat{h}_t}$ are the input and the hidden state at the $\mathbf{t}$-th timestep respectively. $\hat{W}_f, \hat{W}_i, \hat{W}_o, \hat{W}_c, \hat{U}_f, \hat{U}_i, \hat{U}_o, \hat{U}_c, \hat{b}_f, \hat{b}_i, \hat{b}_o$ and $\hat{b}_c$ are weighted matrices to be learned.

Table 1 Statistics of the Dataset

Statistics	Train	Dev	Test
#Sentence	7262	1813	2262
#Simile sentence	3315	786	987
#Literal sentence	3947	1027	1275
#Token	214073	53083	66913
#Unique token	23934	10298	11872
#Tenor	3365	813	1005
#Vehicle	3333	790	996
Maximum sentence length	231	159	348
Average sentence length	29.5	29.3	29.6
Minimum sentence length	5	8	8

We denote the hidden states given by Equation 4 as $H = (h_1, h_2, ..., h_{|S|})$. Suppose that the collections of labels is L. We denote the labels of words as $Y = (y_1, y_2, ..., y_{|S|})$ where $y_i \in L$. We denote $\psi(H, Y)$ as the score of the sequence and it is calculated according to Equation 7.

$$\psi(H, Y) = \sum_{t=0}^{L} A_{y_t, y_{t+1}} + \sum_{t=1}^{L} P_{t, y_t} \tag{7}$$

where $A \in \mathbb{R}^{|L| \times |L|}$ is a transition matrix and $A_{y_t, y_{t+1}}$ records the score of a transition from current tag y_t to next tag y_{t+1}, P_{t,y_t} represents the score of assigning tag y_t to x_t.

All the scores (P_{t,y_t}) consist of an emission matrix $\mathbb{P} = (p_1, p_2, ..., p_n)$. Its dimension is $n \times k$. p_t is a k-dimension vector. We use a non-linear transformation to calculate p_t as described in Equation 8.

$$p_t = W_p \cdot tanh(W_h h_t) \tag{8}$$

where h_t is the t-th hidden state in HRCE, W_p and W_h are weighted matrices to be learned. p_t can be seen as a tag score vector given the current word without considering context words [21]. The probability of tag sequence Y is:

$$p(Y|H) = \frac{\psi(H, Y)}{\sum_{Y' \in \widetilde{Y}} e^{\psi(H, Y')}} \tag{9}$$

where $\widetilde{Y}$ indicates all possible sequences. In training process, the loss function is:

$$\begin{aligned} Loss &= -log(p(Y|H)) \\ &= -\psi(H, Y) + log \sum_{Y' \in \widetilde{Y}} e^{\psi(H, Y')} \end{aligned} \tag{10}$$

where $\psi(H, Y)$ is calculated according to Equation 7.

4 Experiments

4.1 Dataset

We use the dataset collected by Liu et al. [21]. They collect more than 20,000 Chinese student essays and label every sentence as a simile sentence or not. After that, the annotators further annotate boundaries of tenors and vehicles in simile sentences. The statistics of this dataset is shown in Table 1. There are 7,262 sentences in the training set, 1,813 sentences in the validation set and 2,262 sentences in the test set respectively. This dataset is available online[2].

4.2 Evaluation Metrics

Vehicles and tenors should be extracted in pairs when extracting simile components. Thus, we calculate the precision, recall and F-measure in pairs. These metrics are calculated according to Equation 11, 12 and 13.

$$F_1^e = 2 \cdot P_e \cdot R_e / (P_e + R_e) \tag{11}$$

$$P_e = TP_e / (TP_e + FP_e) \tag{12}$$

$$R_e = TP_e / (TP_e + FN_e) \tag{13}$$

where P_e and R_e are the precision and recall in the extraction task, TP_e is the number of correct vehicle-tenor pairs which are recognized by models correctly, FP_e is the number of pairs which are recognized by models incorrectly, FN_e is the number of vehicle-tenor pairs which are failed to be recognized by models.

However, it is difficult to count the vehicles and tenors in pairs, since there may be more than one vehicle and one tenor in sentences. Moreover, the number of vehicles and tenors may be different. To solve this problem, we simplify the way of counting vehicle-tenor pairs by following previous works [21]. Suppose that $\#vehicle$ and $\#tenor$ are the number of vehicles and tenors in a sentence. The number of vehicle-tenor pairs ($\#pair$) is calculated by Equation 14.

$$\#pair = \begin{cases} 0, & \text{if } \#vehicle \text{ or } \#tenor = 0 \\ max(\#vehicle, \#tenor), & \text{else} \end{cases} \tag{14}$$

When $\#vehicle$ and $\#tenor$ are both larger than 0, we use the maximum number between $\#vehicle$ and $\#tenor$ as $\#pair$. If one of them is 0, $\#pair$ is counted as 0.

In Table 2, we show some cases of counting vehicle-tenor pairs. In case 1, the number of vehicles is 3, and the number of tenor is 2. Both two numbers are not 0. Thus, the number of vehicle-tenor pairs is counted as 3. In case 2, the numbers of vehicles and tenors are both 4. In this case, $\#pair$ is also counted as 4.

[2] https://github.com/cnunlp/Chinese-Simile-Recognition-Dataset

Table 2 Cases of Calculating Vehicle-Tenor Pairs

ID	$\#vehicle$	$\#tenor$	$\#pair$
1	3	2	3
2	4	4	4
3	0	1	0
4	0	0	0

In simile sentences, there are usually at least one vehicle and one tenor. However, models may fail to extract any vehicle or tenor from a simile sentence. Thus, there are some cases whose $\#vehicle$ or $\#tenor$ is 0. In case 3 (Table 2), $\#vehicle$ is 0. Therefore, $\#pair$ is set to be 0. In case 4, $\#pair$ is also 0 when $\#vehicle$ and $\#tenor$ are both 0.

4.3 Compared Models

In this work, we explore the performance of simile component extraction with the following models.

- **CRF**. Following the previous work [21], we also compare the performance between our model and a CRF model. The features of this model is extracted with manually templates [21] and include the tokens and their POS tags within a fixed context window whose size is set to be 5. The dependency parsing based features are also used to capture dependencies between words.
- **Component Extraction (CE) model**. It is the extraction model proposed by Liu et al. [21]. The input of CE is word embedding. Then they use a BiLSTM to process the word embeddings. The hidden states from BiLSTM are fed into a CRF layer to get labels for each word.
- **HRCE**. It is the proposed model in this article. HRCE represents a word in three different views: word level, concept level and character level.
- **HRCE-Char**. It is a variant of HRCE. In this model, each word is represented in word level and character level. We compare the performance between HRCE and HRCE-Char to explore the effectiveness of the representations in concept level.
- **HRCE-Con.**. It is another variant of HRCE. Words in this model is represented in two different views: word level and concept view. Comparing the performance between HRCE and HRCE-Con. can help us demonstrate the effectiveness of character representations.

4.4 Experiment Setup

To compare with the models in the work of Liu et al. [21], we set our experiment settings as closed to the work of Liu et al. [21] as possible. The number of hidden units is set to be 128. The dimension of activation layers for simile sentence classification and component extraction are set to be 32 and 64 respectively. We use a dropout layer [31] between the word embedding layer and the bidirectional LSTM layer. We set the dropout rate as 0.5. We use AdaDelta [38] as the optimizer. We stop our training when the F-measure in validation set is no longer increasing more than ten epochs. Then we use the best performing model in validation set to be the test model. The learning rate is set to be 1.0. We train our models with Nvidia GTX 1080Ti. The size of all the embeddings are set to be 50 and they are initialized from a uniform distribution $[-1, 1]$. The height l and the width m of the filters in CNNs are set to be 1 and 50 respectively. We extract the top 5 concepts of each word from CN-Probase. We implement our models with TensorFlow[3].

Table 3 Experiment Results

Model	P_e	R_e	F_1^e
CRF [21]	31.57%	36.98%	34.06%
CE [21]	55.80%	64.89%	59.98%
HRCE-Char	55.91%	70.06%	63.18%
HRCE-Con.	53.35%	70.43%	60.71%
HRCE	**56.76%**	**72.29%**	**63.58%**
*CE [21]	57.41%	70.15%	63.06%
*HRCE	**58.24%**	**70.99%**	**63.97%**

4.5 Experiment Result

4.5.1 Evaluation Result

The experiment results are shown in Table 3. According to our experiments, HRCE significantly outperforms CE, HRCE-Char and HRCE-Con. in F_1^e (p-value < 0.005).

All the neural models (CE, HRCE-Char, HRCE-Con and HRCE) outperforms CRF. The F_1^e of CRF is 34.06%, while all the F_1^e of neural models are higher than 59%. It demonstrates that the features extracted by neural networks are more effective when extracting simile components.

HRCE and its variants get a better performance than CE. The F_1^e of CE is 59.98%. The value of HRCE-Char and HRCE-Con. are 63.18% and 60.71% respectively. Both of them are higher than the value of CE. HRCE gets the best F_1^e (63.58%) in our experiments. It shows the effectiveness of concept representations and character representations.

Although multi-task learning is not our focus, we also train HRCE along with a language model to explore its performance in multi-task learning. HRCE and the language model share the weighted matrices in BiRNNs. This model is denoted as *HRCE in Table 3. The *CE in Table 3 is the best multi-task model in the work of Liu et al. [21].

In Table 3, the F_1^e of *CE is 63.06%. It is 0.52% lower than the F_1^e of our single task model (HRCE). With the help

[3] https://tensorflow.google.cn

(1)	**Target**	把 [我][4824] 捂得 像 个 “ [粽子][1] ” [I][4824] am covered as a " [zongzi][1] "
	CE	把 我 捂得 像 个 “ 粽子 ” I am covered as a " zongzi "
	HRCE-Char	把 我 捂得 像 个 “ 粽子 ” I am covered as a " [zongzi] "
	HRCE-Con.	把 我 捂得 像 个 “ 粽子 ” [I] am covered as a " [zongzi] "
	HRCE	把 我 捂得 像 个 “ 粽子 ” [I] am covered as a " [zongzi] "
(2)	**Target**	月亮[74] 高挂 在 蔚蓝 的 夜空 中 ， 像 是 被 镶上 了 一道 金色 的 圆圈 the moon[74] in the night sky is like being inlaid with a golden circle
	CE	[月亮] 高挂 在 蔚蓝 的 夜空 中 ， 像 是 被 镶上 了 一道 金色 的 圆圈 the [moon] in the night sky is like being inlaid with a golden circle
	HRCE-Char	[月亮] 高挂 在 蔚蓝 的 夜空 中 ， 像 是 被 镶上 了 一道 金色 的 圆圈 the [moon] in the night sky is like being inlaid with a golden circle
	HRCE-Con.	月亮 高挂 在 蔚蓝 的 夜空 中 ， 像 是 被 镶上 了 一道 金色 的 圆圈 the moon in the night sky is like being inlaid with a golden circle
	HRCE	月亮 高挂 在 蔚蓝 的 夜空 中 ， 像 是 被 镶上 了 一道 金色 的 圆圈 the moon in the night sky is like being inlaid with a golden circle
(3)	**Target**	只 知道 [网络][7] 像 [毒][3] 一样 ， 一吸 就 上瘾 ， 有些 学生 为了 去 网吧 而 逃学 [internet][7] is just like [drug][3]. some addicted students play truant to go to internet bars
	CE	只 知道 网络 像 毒 一样 ， 一吸 就 上瘾 ， 有些 学生 为了 去 网吧 而 逃学 internet is just like drug . some addicted students play truant to go to internet bars
	HRCE-Char	只 知道 [网络] 像 [毒] 一样 ， 一吸 就 上瘾 ， 有些 学生 为了 去 网吧 而 逃学 [internet] is just like [drug] . some addicted students play truant to go to internet bars
	HRCE-Con.	只 知道 网络 像 毒 一样 ， 一吸 就 上瘾 ， 有些 学生 为了 去 网吧 而 逃学 internet is just like drug . some addicted students play truant to go to internet bars
	HRCE	只 知道 [网络] 像 [毒] 一样 ， 一吸 就 上瘾 ， 有些 学生 为了 去 网吧 而 逃学 [internet] is just like [drug] . some addicted students play truant to go to internet bars
(4)	**Target**	两 [眼睛][220] 也 是 圆 的 ， 睁开眼 皮 ， 它们 就 像 两 [双胞胎][0] 似地 行动 一致 [eyes][220] are round . when they are open , they move together just like [twins][0]
	CE	两 眼睛 也 是 圆 的 ， 睁开眼 皮 ， 它们 就 像 两 双胞胎 似地 行动 一致 eyes are round . when they are open , they move together just like twins
	HRCE-Char	两 [眼睛] 也 是 圆 的 ， 睁开眼 皮 ， 它们 就 像 两 [双胞胎] 似地 行动 一致 [eyes] are round . when they are open , they move together just like [twins]
	HRCE-Con.	两 [眼睛] 也 是 圆 的 ， 睁开眼 皮 ， 它们 就 像 两 双胞胎 似地 行动 一致 [eyes] are round . when they are open , they move together just like twins
	HRCE	两 [眼睛] 也 是 圆 的 ， 睁开眼 皮 ， 它们 就 像 两 [双胞胎] 似地 行动 一致 [eyes] are round . when they are open , they move together just like [twins]

Tenor: [ ] **Vehicle:** [ ]

Fig. 3 Extraction Cases From Compared Models of hybrid representations, words are represented more accurately in HRCE. Therefore, HRCE can outperforms the best multi-task model of Liu et al. [21]. After applying multi-task learning into our model, the weighted matrices can be trained more sufficiently. Thus, *HRCE can get 63.97% in F_1^e which is 0.91% higher than the F_1^e of *CE.

4.5.2 Case Study

To further explore the performance of the models in our experiments, we sample some cases and show them in Fig. 3. Words in blue square brackets indicate tenors while words in red square brackets indicate vehicles. The first line in each case is the target and the numbers in the top right corner are the word frequencies in the training set.

In case 1, the tenor should be "I" and the vehicle should be "zongzi". The word "zongzi" is a rare word and CE fails to extract it correctly. However, The word "I" is a high frequency word. It appears 4824 times in the training set. CE also fails to extract such a high frequency word. In HRCE-Char, it successfully extracts "zongzi" with the help of character representations. It still fails to extract the word "I". In HRCE-Con. and HRCE, both "I" and "zongzi" are successfully extracted. The difference between these two models and other models is that there are concept representations in these two model. This case can show the effectiveness of concept representations. Case 2 can also demonstrate the same result. The sentence in case 2 is a literal sentence. Thus, there are no tenor or vehicle in this sentence. However, CE and HRCE-Char label the word "moon" as a tenor incorrectly. The frequency of the word "moon" is 74 and it is not a rare word. This problem is solved after integrating concept information. Both HRCE-Con. and HRCE can avoid this mistake.

In case 3, the tenor and vehicle should be "internet" and "drug" respectively. Both "internet" and "drug" are rare words in the training sets. CE and HRCE-Con. fails to extract both of them. With the help of character representations, HRCE-Char and HRCE can extract both of them successfully. Similar results can be found in case 4. The tenor and vehicle are "eyes" and "twins" respectively. The word "eyes" appears 220 times in the training set. All the compared models except CE can successfully extract the word "eyes". However, the word "twins" is an unseen word in the training set. Only HRCE-Char and HRCE can extract the word "twins" successfully. Both case 3 and 4 demonstrate that character representations can help HRCE to extract rare words and unseen words more accurately.

In all the cases, only HRCE can extract both tenors and vehicles successfully. Therefore, it is important to integrate both character representations and concept representations into simile component extraction models.

5 Conclusion

In this article, we propose a Hybrid Representation based Component Extraction (HRCE) model to extract the simile components (tenors and vehicles). Each word in HRCE is represented as three different levels: word level, concept level and character level. The representation in concept level can help HRCE to identify whether two words are in the same concept domain more accurately. Moreover, the representation in character level can help HRCE to label rare words or unseen words more accurately. We conduct experiments to compare the performance between HRCE and current extraction model. In our experiments, HRCE significantly outperforms other compared models. These results also demonstrate the effectiveness of concept representations and character representations.

In the future, we will try to further improve the performance of extraction model by integrating other information (e.g. sentence structure). What's more, it is also interesting to apply concept representation and character representation into other metaphor analysis tasks.

Acknowledgements This work presented in this paper is partially supported by the Fundamental Research Funds for the Central Universities, SCUT (Nos. 2017ZD048, D2182480), the Tiptop Scientific and Technical Innovative Youth Talents of Guangdong special support program (No.2015TQ01X633), the Science and Technology Planning Project of Guangdong Province (No.2017B050506004), the Science and Technology Program of Guangzhou (Nos. 201704030076, 201802010027). The research described in this paper has been supported by a collaborative research grant from the Hong Kong Research Grants Council (project no. C1031-18G).

References

1. Akbik A, Blythe D, Vollgraf R (2018) Contextual string embeddings for sequence labeling. In: Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, pp 1638–1649
2. Akbik A, Bergmann T, Vollgraf R (2019) Pooled contextualized embeddings for named entity recognition. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp 724–728
3. Alzaidy R, Caragea C, Giles CL (2019) Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, pp 2551–2557, DOI 10.1145/3308558.3313642
4. Bingel J, Søgaard A (2017) Identifying beneficial task relations for multi-task learning in deep neural net-

works. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, pp 164–169
5. Carbonell JG (1980) Metaphor: A key to extensible semantic analysis. In: Proceedings of the 18th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '80, pp 17–21, DOI 10.3115/981436.981441
6. Caruana R (1998) Multitask learning. In: Learning to Learn., pp 95–133, DOI 10.1007/978-1-4615-5529-2_5
7. Chen L, Moschitti A (2019) Transfer learning for sequence labeling using source model and target data. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019., pp 6260–6267
8. Cho K, van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder–decoder approaches. In: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, pp 103–111, DOI 10.3115/v1/W14-4012
9. Ekbal A, Saha S (2016) Simultaneous feature and parameter selection using multiobjective optimization: application to named entity recognition. Int J Machine Learning & Cybernetics 7(4):597–611, DOI 10.1007/s13042-014-0268-7
10. Ekbal A, Saha S, Sikdar UK (2016) On active annotation for named entity recognition. Int J Machine Learning & Cybernetics 7(4):623–640, DOI 10.1007/s13042-014-0275-8
11. Gómez-Rodríguez C, Vilares D (2018) Constituent parsing as sequence labeling. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pp 1314–1324
12. Hanks P (2012) The roles and structure of comparisons, similes, and metaphors in natural language (an analogical system). Prose (in honor of the Dickens Bicentennial) p 5
13. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computation 9(8):1735–1780
14. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991, 1508.01991
15. Jang H, Jo Y, Shen Q, Miller M, Moon S, Rosé CP (2016) Metaphor detection with topic transition, emotion and cognition in context. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers
16. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. In: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pp 260–270
17. Lei X, Cai Y, Xu J, Ren D, Li Q, Leung H (2019) Incorporating task-oriented representation in text classification. In: Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, Part II, pp 401–415, DOI 10.1007/978-3-030-18579-4_24
18. Li B, Yu Ll, Shi M, Qu Wg (2008) Computation of chinese simile with" xiang". Journal of Chinese Information Processing 22(6):27–32
19. Li Y, Wang J, Wang S, Liang J, Li J (2019) Local dense mixed region cutting + global rebalancing: a method for imbalanced text sentiment classification. Int J Machine Learning & Cybernetics 10(7):1805–1820, DOI 10.1007/s13042-018-0858-x
20. Liu A, Du J, Stoyanov V (2019) Knowledge-augmented language model and its application to unsupervised named-entity recognition. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp 1142–1150
21. Liu L, Hu X, Song W, Fu R, Liu T, Hu G (2018) Neural multitask learning for simile recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp 1543–1553
22. Ma X, Hovy EH (2016) End-to-end sequence labeling via bi-directional lstm-cnns-crf. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers
23. Miwa M, Bansal M (2016) End-to-end relation extraction using lstms on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, pp 1105–1116, DOI 10.18653/v1/P16-1105
24. Niculae V, Danescu-Niculescu-Mizil C (2014) Brighter than gold: Figurative language in user generated com-

parisons. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp 2008–2018
25. Peters ME, Ammar W, Bhagavatula C, Power R (2017) Semi-supervised sequence tagging with bidirectional language models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pp 1756–1765, DOI 10.18653/v1/P17-1161
26. Qadir A, Riloff E, Walker M (2015) Learning to recognize affective polarity in similes. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp 190–200, DOI 10.18653/v1/D15-1019
27. Qadir A, Riloff E, Walker MA (2016) Automatically inferring implicit properties in similes. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp 1223–1232, DOI 10.18653/v1/N16-1146
28. Rei M (2017) Semi-supervised multitask learning for sequence labeling. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pp 2121–2130, DOI 10.18653/v1/P17-1194
29. Ren D, Cai Y, Lei X, Xu J, Li Q, Leung H (2019) A multi-encoder neural conversation model. Neurocomputing 358:344–354, DOI 10.1016/j.neucom.2019.05.071
30. Shutova E, Teufel S, Korhonen A (2013) Statistical metaphor processing. Computational Linguistics 39(2):301–353, DOI 10.1162/COLI_a_00124
31. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1):1929–1958
32. Strzyz M, Vilares D, Gómez-Rodríguez C (2019) Sequence labeling parsing by learning across representations. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp 5350–5357
33. Strzyz M, Vilares D, Gómez-Rodríguez C (2019) Viable dependency parsing as sequence labeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp 717–723
34. Veale T (2012) A context-sensitive, multi-faceted model of lexico-conceptual affect. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, pp 75–79
35. Veale T, 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, vol 29
36. Wang X, Yang C, Guan R (2018) A comparative study for biomedical named entity recognition. Int J Machine Learning & Cybernetics 9(3):373–382, DOI 10.1007/s13042-015-0426-6
37. Wiseman S, Stratos K (2019) Label-agnostic sequence labeling by copying nearest neighbors. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers, pp 5363–5369
38. Zeiler MD (2012) ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701, `1212.5701`