Combining Part-of-Speech Tags and Self-Attention Mechanism for Simile Recognition

PENGFEI ZHANG, YI CAI, JUNYING CHEN, WENHAO CHEN, AND HENGJIE SONG

ABSTRACT

Simile recognition is to find simile sentences and extract the tenor and vehicle from these sentences. Previous works illustrate that tenors and vehicles are typically noun phrases. A word may have different part-of-speech (POS) labels (e.g., adjectives, adverbs, nouns, and verbs) in different sentences. It is important for the simile recognition task to identify a certain POS information for each word in a sentence. However, existing models use the same word embedding to represent a word, which cannot accurately represent the POS information of this word in different sentences. In this paper, we propose a neural network framework explicitly integrating the POS information into simile recognition task, with additional self-attention mechanism to better capture long term dependencies between any two tokens in sentences. The experimental results show that our proposed models significantly outperform previous state-of-the-art methods in the simile recognition task. We also present an analysis showing that the POS information and self-attention mechanism are effective for the simile recognition task.

INDEX TERMS

Simile recognition, part-of-speech, self-attention mechanism.

I. INTRODUCTION

A simile is a figure of speech that directly compares tenors and vehicles using connecting words such as “like” or “as”. These connecting words we called comparators [1]. In a simile sentence, the tenor acts as logical subject to which attributes are ascribed, while the vehicle is the compared object [2], [3] whose attributes are borrowed. For example, in the sentence “The leaf is like the butterfly”, “The leaf” is the tenor, and “the butterfly” is the vehicle.

Simile is widely used in human language, which can be viewed as a rhetorical device for making thoughts or expressions more vivid. With the help of simile, people can understand a sentence better. The analysis of the simile becomes active in recent years [4]–[9]. One important task in simile analysis is simile recognition. The simile recognition task is to find simile sentences and extract the corresponding simile components (tenors and vehicles) [10] from these simile sentences. This task includes two subtasks: simile sentence classification and simile component extraction. For instance, given a simile sentence “[The leaf] is like [the butterfly],” with annotated the target tenor “The leaf” and vehicle “the butterfly”. The goal of the simile sentence classification task is to automatically classify this sentence to similes, and the goal of the simile components extraction task is to recognize the tenor and the vehicle precisely.

Simile recognition is very important in the dialogue system. A simile is a figure of speech that uses the vehicle to explain the tenor or topic. In the dialogue system, if the model cannot correctly recognize the tenor of the simile, it is easy to generate responses with inconsistent topic, and the dialogue may be difficult to proceed. On the other hand, the study of the simile component extraction task can build a rich knowledge base of similes for the simile generation task. In the dialogue
system, if the dialogue model can use similes as responses, it can make conversations more interesting.

Traditional simile recognition methods rely largely on feature representation or pattern design. The former usually exploits a set of features derived from the heavy feature engineering. It is labor-intensive and time-consuming to construct these features with linguistic and syntactic cues. However, the performance of these feature-based methods is highly limited by the quality of the features. The latter depends largely on the designed pattern, while it is difficult to deal with sentences with complex structures. Recently, deep neural architectures have attracted increased attention. Benefit from the ability of the neural network architecture to automatically extract features, neural network-based models have been employed in the simile recognition task to reduce the feature engineering effort. They have achieved the state-of-the-art results on the simile recognition task.

Although remarkable improvements have been achieved by these neural network-based models in the simile recognition task, there are still some problems. One significant issue is that existing neural network-based models are difficult to recognize tenors and vehicles. Previous works illustrate that tenors and vehicles are typically noun phrases [1], [3]. Thus, the POS information is crucial to extract tenors and vehicles.

Meanwhile, we observe that a word may have different POS tags in different sentences. As shown in Table 1, there are three sentences. In this example, the word “奔跑 (running)” is a noun in the first sentence while it is a verb in the second sentence. We can see that the word “奔跑 (running)” is a vehicle in the first sentence while it is not in the second sentence. The main reason is that this word has different POS tags in the two sentences. It is important for the simile recognition task to identify a certain POS label for each word in a sentence. However, existing neural network-based models use word embeddings as their input. In this time, a word with different POS tags in different sentences are represented with a same vector. Therefore, it is difficult for these models to identify the POS information of a word only based on its word embedding. As a result, these models could not identify the simile tag (tenor, vehicle or others) of words with different POS labels correctly. Thus, how to exploit explicit POS information is a challenging problem.

Moreover, we observe that tenors and vehicles are similar in a certain aspect. Thus, it is important for simile recognition methods to consider the dependencies between the tenor and the vehicle in a sentence. The positions of the tenor and the vehicle in a sentence are uncertain in advance. Instead of directly considering the dependencies between the tenor and the vehicle, we can consider the dependencies of each word in the sentence. However, mainstream models are based on recurrent neural networks (RNNs). Previous works have showed that RNNs cannot conduct long-term dependencies between arbitrary two words in sentences [11]. Thus, the RNNs-based models would suffer the long-term dependencies. As a result, these models hardly identify a tenor and a vehicle when their distance is far away. In the last sentence at the Table 1, if the simile recognition model only focuses on the word “爱情 (love)”, it is difficult to identify its simile tag (tenor, vehicle or others) clearly. However, when the model explicitly captures the dependencies among “爱情 (love)”, “文字 (words)” and “像 (like)”, it is easy to identify “爱情 (love)” as vehicle. Thus, how to better capture the global dependencies of each word is another challenge.

To solve the above problems, we propose a neural network framework integrating the explicit POS information into the simile recognition task, with additional self-attention mechanism to better capture long-term dependencies in sentences. Specifically, we integrate explicit POS information into input to enrich the presentation of words with different POS tags. Then, we apply self-attention mechanism to help bidirectional long short term memory (BiLSTM) to better capture long-term dependencies with conducting direct dependencies between arbitrary two words in sentences. Finally, we evaluate our models on widely used simile recognition datasets. The experimental results show that our proposed models significantly outperform previous state-of-the-art methods and gain new benchmarks in the simile recognition task.

Our contributions are summarized as follows:

- We propose a neural network framework, which incorporates explicit POS information into simile recognition tasks. It can help our models identify words with different POS information more accurately. To the best of our knowledge, it is the first work which integrates POS information in the simile recognition task.
- To better capture the global dependencies of the whole sentence, we integrate self-attention mechanisms into our models. Self-attention mechanisms can help our model to build the connections between arbitrary two words. Thus, our models can identify tenors and vehicles more accurately when their distances are long.

**TABLE 1.** Examples of illustrating that tenors and vehicles are typically noun phrases.

<table>
<thead>
<tr>
<th>ID</th>
<th>Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[Learning] is like [running], only by constantly uphold can reach the destination</td>
</tr>
<tr>
<td>2</td>
<td>[She] is like a dislocated [wild horse], [running] fast on the track</td>
</tr>
<tr>
<td>3</td>
<td>The [words] are simple, they are beautiful and fragile, just like [love],</td>
</tr>
</tbody>
</table>

\(f\) denotes the tenor and \(v\) denotes the vehicle.
We conduct our experiments on the widely used Chinese Simile Recognition datasets, and the experimental results show that our proposed models can achieve better performance than previous state-of-the-art models.

The rest of this paper is organized as follows. In Section II, we briefly introduce the related works of simile recognition. Section III is a detailed description of our model. We give a detailed analysis of experimental results in Section IV. Section V discusses the future work and concludes this paper.

II. RELATED WORK
A. SIMILE ANALYSIS
A simile is a figure of speech that directly compares two things using connecting words such as “like” or “as” [10]. Recently, there are more and more researches about similes, which include sentiment analysis [12], [13], implicit properties inference [14] and components recognition in simile sentences [10]. In this paper, we focus on the simile recognition task. This task is to find simile sentences and extract the corresponding simile components from these sentences [10].

Simile recognition is a widely studied task in the natural language processing (NLP) field. There are various methods been proposed for simile recognition. They can be categorized into three classes: feature based [3], [15], pattern mining based [2], [16], and neural network based [10].

In feature based methods, different sets of features are derived from a heavy feature engineering. Then these features are fed into a classifier (e.g., support vector machine (SVM)) and a extractor (e.g., conditional random field (CRF)) for the simile sentence classification task and the simile components extraction task respectively. Li et al. [15] use a maximum entropy model to recognize simile sentences. The features exploited in their classifier including tokens and POS tags of the words around the comparator within a fixed contextual window. Thus, their model cannot capture complete context features when preventing the noise. For the simile components extraction task, they use a CRF model to combine the result of the classifier and the designed features, which may suffer the error propagation. In addition, their experiments are conducted on a small Chinese simile corpus only with 1586 sentences. The dataset used in our work is much larger.

Niculae and Danescu-Niculescu-Mizil [3] use dependency parse tree patterns to extract candidate simile components and then employed a classifier to recognize a comparison from figurative or literal in product reviews. This classifier is based on the candidate components. The main limitation is that their methods will suffer the error propagation problem. The performance of the classifier is largely limited by the accuracy of candidate components recognition.

Syntactic pattern mining methods are often used for extracting potential simile components [2], [16]. Niculae and Yaneva [2] use constituent parsing with GLARF [17] transformations in order to match several hand-written comparison patterns. However, the GLARF is only available in English. To relief this problem, Niculae [16] uses dependency parsing instead of the constituent parsing. Such pattern based methods is difficult to deal with sentences with complex structures. Intuitively, these methods are not very generalized. As a result, the coverage is relatively small.

Deep neural networks have emerged recently and can learn robust underlying features automatically with promising results. Liu et al. [10] use a bidirectional long short term memory (BiLSTM) to extract word-level and sentence-level features. Then, they feed these features into the neural CRF model to recognize tenors and vehicles for the simile components recognition task. For the simile classification task, they apply an attention layer to summarize all data information among these features to recognize the label (e.g. simile or literal) of a sentence. At the attention layer, they use an extra “attention” vector to automatically select relevant content for every sentence. The basic idea is that the contribution of words to the semantic of a sentence is different. In addition, they notice that the simile sentence classification task and the simile components extraction task can benefit each other and the interactions between them should not be ignored. Intuitively, the sentence classifier could make a more confident decision if the components extractor tells the classifier that the tenor and the vehicle likely exist. Liu et al. [10] also propose a neural multitask learning model for the simile recognition task, which jointly optimizes three goals: simile sentence classification, simile component extraction and language modeling. The language modeling is an auxiliary task to learn more sufficient local information. For each word, the goal of the language modeling is to predict the next word in a sentence. Previous works have demonstrated that the tenors and vehicles are typically nouns [1]. However, Liu et al. [10] do not consider its explicit POS information. As a result, their model could not identify the simile tags of words with different POS label correctly.

B. SELF-ATTENTION MECHANISM
Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Self-attention is usually integrated with an encoder-decoder architecture in many text generation tasks, which repeatedly process their input by selecting relevant content at every step. Cheng et al. [18] exploit a long short term memory network (LSTMN) based on the standard LSTM for machine reading. A standard LSTM processes a variable-length sequence by incrementally adding new content into a single memory unit, with gates controlling the extent to which new content should be memorized (input gate), old content should be erased (forget gate), and current content should be exposed (output gate). However, the standard LSTM cannot maintain unbounded memory when the size of the single memory unit is not large enough, the key idea behind the LSTMN is to use self-attention for inducing relations between tokens. For language inference, Parikh et al. [19] propose a neural architecture with intra-sentence attention mechanism and achieve previous state-of-the-art results. The idea behind their methods is to use self-attention to decompose the problem into subproblems...
that can be solved separately, thus making it trivially parallelizable. Besides, Paulus et al. [20] bring the self-attention mechanism into the abstract summarization task. Their model produces higher quality summaries according to the human evaluation. Recently, Vaswani et al. [21] propose a novel network architecture named Transformer for the machine translation task, which based mainly on self-attention mechanisms. With the remarkable results Transformer has achieved, there are more researches based on the self-attention mechanisms. We are the first to introduce the self-attention mechanism to the simile recognition task.

III. MODEL
In this paper, we propose a neural network framework integrating the explicit POS information into the simile recognition task, with additional self-attention mechanism to better capture long term dependencies between any two words in sentences. The general structure of our models is shown in Figure 1. The left model is for the simile sentence classification task while the middle model is for the simile component extraction task. The models mainly consist of three components: task-specific feature extractor, simile sentence classification module and simile component extraction module. We use task-specific feature extractor to extract the features in the sentence for the classification module and the extraction module respectively. Then, these features are fed into our classification module and extraction module to recognize simile sentences and simile components respectively. In the following, we will briefly introduce the simile recognition task and then describe the proposed models in detail.

A. TASK DESCRIPTION
Given a sentence with \( n \) words \( X^w = \{w_1, w_2, \cdots, w_N\} \), which contains a comparator. The simile recognition task is to find similes and extract simile components from these sentences. Notice that a sentence containing a comparator cannot guarantee that it is a simile sentence. For example, the sentence “The boy looks like his father” also contains the comparator, but it is a literal comparison because it does not trigger a cross-domain concept mapping.

Simile recognition can be divided into two subtasks:

**Simile Sentence Classification (SSC).** This subtask is a text classification problem, which need to classify the sentence \( X^w \) into predefined labels \( c \) (simile or literal).

**Simile Component Extraction (SCE).** This task can be viewed as a sequence labeling problem, which need to assign a simile tag \( y_i \) (tenor, vehicle or others) to each word \( w_i \).

B. TASK-SPECIFIC FEATURE EXTRACTOR
In our model, the task-specific feature extractor involves two parts: word-level feature extractor and task-specific sentence-level feature extractor.
1) WORD-LEVEL FEATURE EXTRACTOR
In this module, we first map the words and the corresponding POS tags to dense distributed representations respectively. We call such dense distributed representations embeddings. Then, we add the word embeddings and POS embeddings together to get a new word representation. Specifically, for a given sentence \( X^w = \{w_1, w_2, \ldots, w_N\} \) from CSRD dataset [10], we first get the corresponding POS tags \( X^p = \{p_1, p_2, \ldots, p_N\} \) with POS tagger, such as HITLTP toolkit\(^1\) and THULAC toolkit.\(^2\) Then we lookup word embedding vector from word embedding matrix for each word and POS embedding vector from POS embedding matrix for the corresponding POS tag. We add these two vector together to get a new word representation. The new word representation could be expressed as follows:

\[
\tilde{x}_i = x^w_i + x^p_i
\]

where \( x^w_i \in \mathbb{R}^{d_w} \) and \( x^p_i \in \mathbb{R}^{d_p} \) denotes the word embedding and POS embedding for \( w_i \).

2) TASK-SPECIFIC SENTENCE-LEVEL FEATURE EXTRACTOR
The task-specific sentence-level feature extractor is designed to extract private features from sentences for the simile sentence classification task and the simile component extraction task. It consists of two parts: private BiLSTM layer and self-attention mechanism.

\( a: PRIVATE BiLSTM LAYER\)
Recurrent neural networks (RNNs) [29] are widely used in natural language processing (NLP) for handling sequential data. Standard RNNs would suffer the problems of gradient vanishing and exploding. To relieve these problems, Long short term memory (LSTM) [30] were proposed. Standard LSTM only calculates the output of the current moment based on the previous state. Thus, LSTM can only leverage the past information and will ignore the future information [11]. However, the output of the current moment should be related both to the previous and future state. In order to leverage both the future and past information, bidirectional long short term memory (BiLSTM) is used in our proposed models, which introduce a forward and backward LSTM to capture the past information and future information in sentences. At the \( i \)-th time step, forward LSTM takes the hidden state from the previous time step and the word representation \( \tilde{x}_i \) from the current step as inputs, and produces a hidden state for current step. The backward LSTM can be considered as the reverse of the forward LSTM. The formal calculation process for the hidden state as follows:

\[
\begin{align*}
\hat{h}_i &= \text{LSTM}(\tilde{h}_{i-1}, \tilde{x}_i) \\
\tilde{h}_i &= \text{LSTM}(\hat{h}_{i+1}, \tilde{x}_i) \\
h_i &= [\hat{h}_i; \tilde{h}_i]
\end{align*}
\]

where \( X = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N) \) denotes the output of word-level feature extractor, \( \hat{h}_i \in \mathbb{R}^{d_h} \) and \( \tilde{h}_i \in \mathbb{R}^{d_h} \) are the \( i \)-th hidden states of the forward and backward LSTM respectively. \([;\] denotes the concatenation operation.

We propose a task-specific feature extractor, which assigns a private BiLSTM layer for task \( k \in \{SCE, SSC\} \). The private BiLSTM layer is utilized to extract task-specific features. Formally, for the two subtasks \( k \), the hidden states of private BiLSTM layer can be computed as follows:

\[
\begin{align*}
h^k_i &= \text{BiLSTM}(\tilde{x}_i, h^k_{i-1}, \theta_k)
\end{align*}
\]

where \( \theta_k \) denotes the private BiLSTM parameters of task \( k \).

\( b: SELF-ATTENTION MECHANISM\)
Long-term dependencies in RNNs are that the state of current moment may be affected by the state of long time ago. Self-attention mechanism is able to capture global dependencies by scaled dot-product mechanism without being affected by location information. Thus, we introduce the self-attention mechanism to help BiLSTM better capture long range dependencies of the sentence. Recently, multi-head self-attention mechanism has been proved more effective in various NLP tasks [11], [21], [31]. Therefore, we use the multi-head self-attention mechanism to capture different aspect context features in our proposed models.

Multi-head attention mechanism linearly projects the queries, keys and values \( n \) times with different linear projections respectively, where the query, keys, values, and output are all vectors. Then we perform the scaled dot-product attention for these projected versions in parallel. These results of the scaled dot-product attention are concatenated and once again projected to get the final representation \( H' \). Multi-head attention mechanism can be computed as follows:

\[
\begin{align*}
H' &= [head_1; head_2; \ldots; head_n]W_h \\
head_i &= \text{Attention}(QW_i^q, KW_i^k, VW_i^v) \\
\text{Attention}(Q, K, V) &= \text{softmax}(\frac{QK^T}{\sqrt{d}})V
\end{align*}
\]

where \( Q \in \mathbb{R}^{N \times 2d_h} \), \( K \in \mathbb{R}^{N \times 2d_h} \) and \( V \in \mathbb{R}^{N \times 2d_h} \) are query matrix, key matrix and value matrix, respectively. \( W_i^q \in \mathbb{R}^{d_q \times d_h} \), \( W_i^k \in \mathbb{R}^{d_k \times d_h} \), \( W_i^v \in \mathbb{R}^{d_v \times d_h} \) and \( W_h \in \mathbb{R}^{2d_h \times 2d_h} \) is the trainable parameter for linear projection layer and \( d_k = 2d_h/n \). \( n \) is the number of heads in the multi-head attention mechanism. \( d_h \) is the dimension of hidden units of BiLSTM, which equals to \( 2d_h \). In the self-attention mechanism, \( Q = K = V = H \). \( H = [h_1, h_2, \ldots, h_N] \) denotes the output of the RNNs layer.

C. TASK1: SIMILE SENTENCE CLASSIFICATION
The simile sentence classification task is to distinguish similes from literals, which is a text classification task. In the text classification task, words should not have the same contribution to the representation of the sentence meaning [32]. Hence, we adopt the attention mechanism to calculate the contributions or weights \( \alpha \) for each word in a sentence.

\(^1\)https://github.com/HIT-SCR/pyslt
\(^2\)https://github.com/thunlp/THULAC-Python
According to the calculated weight, the final representation \( h_{ssc}' \) is weighted and summed to get a sentence representation \( R \). The process can be formalized as follows:

\[
\alpha = \text{softmax}(\text{tanh}(H_{ssc} \cdot W_a)) \\
R = \alpha^T \cdot H_{ssc}'
\]  

(5)

where \( W_a \in \mathbb{R}^{2d_h+|L| \times 1} \) is a trainable parameter matrix. \( H_{ssc}' = [h_{ssc}'_1, h_{ssc}'_2, \ldots, h_{ssc}'_N] \) is the new word representations extracted by the task-specific feature extractor for the simile sentence classification task. \( R \) is the sentence representation generated by an attention layer.

Then, we get the label probability distribution of the sentence via a non-linear transformation layer and a softmax layer. The loss function is the cross entropy of the predicted label probability distribution:

\[
E_{ssc} = -\text{Log}(C')
\]

(6)

where \( W_c' \in \mathbb{R}^{d_{ssc} \times 2}, W_e \in \mathbb{R}^{2d_h+|L| \times d_{ssc}} \) are trainable parameters. Thus, \( C' \) is the probability distribution of the predicted class (i.e. simile or literal). \( C \) is the one-hot encoding of the ground-truth in the simile sentence classification task.

**D. TASK2: SIMILE COMPONENT EXTRACTION**

The simile component extraction task is to extract tenors and vehicles in a given sentence. This task can be viewed as a sequence labeling problem, which need to assign a tag to each word in a sentence. Conditional Random Field (CRF) [33] is widely used to deal with sequence labeling problem, which can effectively capture dependencies among tags. Instead of merely using the final representation \( h_i' \) to make tagging decisions, we build a CRF layer [34] on the feature extractor.

Formally, \( H_{ssc} = [h_{ssc}'_1, h_{ssc}'_2, \ldots, h_{ssc}'_N] \) is the output of the feature extractor for the simile component extraction task, \( Y = \{y_1, y_2, \ldots, y_N\} \) denotes the tag sequence for the given sentence \( X^w \), \( y_i \in L \) and \( |L| \) denotes the number of output tags. The CRF layer can be expressed as follows:

\[
P_i = W_{i}' \cdot \text{tanh}(W_{i} h_{ssc}'_i) \\
S(X^w, Y) = \sum_{i=0}^{N} T_{y_i, y_{i+1}} + \sum_{i=1}^{N} P_{i, y_i}
\]

(7)

where \( W_{i} \in \mathbb{R}^{d_{ssc} \times 2d_h}, W_{i}' \in \mathbb{R}^{d_{ssc} \times 2} \) are trainable parameters. \( P_{i, y_i} \) represents the score that the word \( w_i \) belongs to the \( y_i \)-th tag. \( T \in \mathbb{R}^{|L| \times |L|} \) is a transition score matrix among all simile tags and \( T_{y_i, y_{i+1}} \) represents the score of a transition from current label \( y_i \) to next label \( y_{i+1} \). In decoding, we use Viterbi algorithm [33] to get the predicted tag sequence with the largest score, as shown in follows.

\[
\overline{Y} = \text{argmax}_{Y \in Y_t} S(X^w, Y)
\]

(8)

where \( Y_t \) is all candidate tag sequences for \( X^w \).

In training, the loss function is the negative log-likelihood objective of the correct simile tag sequence:

\[
p(Y|X^w) = \frac{e^{S(X^w, Y)}}{\sum_{Y \in Y_t} e^{S(X^w, Y)}} \\
E_{ssc} = -\log p(Y|X^w) = -S(X^w, Y) + \log \sum_{Y \in Y_t} e^{S(X^w, Y)}
\]

(9)

where \( p(Y|X^w) \) is the probability of the correct tag sequence.

**E. TRAINING**

The final loss function of our proposed model can be defined as follows:

\[
E = E_{ssc} \cdot I(k) + E_{ssc} \cdot (1 - I(k))
\]

(10)

where \( k \in \{SSC, SCE\} \) is the name of two subtasks. \( E_{ssc} \) and \( E_{sce} \) is the loss functions of two subtasks respectively. \( I(k) \) is a switching function. It is defined as follows:

\[
I(k) = \begin{cases} 1 & \text{if } k \text{ is SSC} \\ 0 & \text{if } k \text{ is SCE} \end{cases}
\]

(11)

In the training phrase, we first select a task from \{SSC, SCE\}. Then, we use the train set from given task to update the parameters. Thus, the final parameters in the task-specific feature extractor for two subtasks are different.

**IV. EXPERIMENTS**

**A. DATASET**

To evaluate our proposed model on simile recognition task, we have done a lot of experiments on the open source Chinese Simile Recognition Dataset (CSRD). Each sentence is labeled with simile or literal, and it have already been conducted to word segmentation with HITLTP toolkit. Thus, each token is annotated with tenor, vehicle or others in IOBES scheme (indicating Inside, Outside, Beginning, Ending, and Single). Different prefixes are used to distinguish the tenor and the vehicle components. For example, \(tb\) and \(vb\) denote the beginning of a tenor and a vehicle respectively. All types and meanings of the tags are listed in table 2. In our experiments, we used the same training, development and testing splits as [10]. Table 3 gives the details of the dataset.

**B. SETTINGS**

For hyper-parameter settings, we adjust them according to the performance on development set of the simile component extraction task. We set the word embedding size \( d_w \) to 50. The size of LSTM hidden state in each direction \( d_h \) is 128. The number of projections \( n \) in the self-attention mechanism is 4. We set the batch size to 20. The dimension of nonlinear transformation for the simile component extraction task \( d_{ssc} \) and the simile sentence classification task \( d_{sce} \) are set to 64 and 32 respectively. We apply a dropout layer [35] between
TABLE 2. All types and meanings of the tags.

<table>
<thead>
<tr>
<th>Tag name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>tb</td>
<td>The beginning of a tenor</td>
</tr>
<tr>
<td>tm</td>
<td>The inside of a tenor</td>
</tr>
<tr>
<td>te</td>
<td>The ending of a tenor</td>
</tr>
<tr>
<td>ts</td>
<td>The single token which is a tenor</td>
</tr>
<tr>
<td>vb</td>
<td>the beginning of a vehicle</td>
</tr>
<tr>
<td>vm</td>
<td>the inside of a vehicle</td>
</tr>
<tr>
<td>ve</td>
<td>the ending of a vehicle</td>
</tr>
<tr>
<td>vs</td>
<td>The single token which is a vehicle</td>
</tr>
<tr>
<td>O</td>
<td>The outside token</td>
</tr>
</tbody>
</table>

TABLE 3. Statistics of the CSRD dataset.

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

For parameters initialization, we use Xavier initializer \cite{03} to initialize trainable parameters. The pre-trained word embeddings\footnote{https://github.com/cnunlp/Chinese-Simile-Recognition-Dataset/tree/master/scripts} used in our experiments are same as \cite{09}, which are pre-trained on a large essay corpus crawled from the web by using word2vec toolkit \cite{10}. The POS embeddings is randomly initialized with the same dimension as the pre-trained word embeddings. Batch Normalization \cite{11} is used in self-attention mechanism to accelerating self-attention network training. We adopt the AdaDelta \cite{12} as our optimizer with the initial learning rate of 1.0. The early stop strategy \cite{13} is used to finish the training process. Our models are implemented with Tensorflow framework \cite{14}. All the experiments are conducted on a server with one GTX 1080Ti GPU.

For evaluation, we used the pair-wise level \cite{15} Precision (PP), Recall (PR) and F1 score (PF1) as metrics for the simile component extraction task, while using the Precision (P), Recall (R) and F1 score (F1) as metrics for the simile sentence classification task. Formally, the pair-wise level metrics can be expressed as follows:

\[
PP = \frac{C}{B} \\
PR = \frac{C}{A} \\
PF1 = \frac{2PP \times PR}{PP + PR} \\
\tag{12}
\]

where \(A\) is the number of tenor-vehicle pairs in the test set, \(B\) denotes the number of predicted tenor-vehicle pairs, and \(C\) is the number of the correct tenor-vehicle pairs. The correct tenor-vehicle pair means both simile components are correct.

TABLE 4. The experimental results for simile sentence classification task.

<table>
<thead>
<tr>
<th>Models</th>
<th>P(%)</th>
<th>R(%)</th>
<th>PF1(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>65.23 *</td>
<td>47.52 *</td>
<td>54.98 *</td>
</tr>
<tr>
<td>Random Forest (POS)</td>
<td>76.61 *</td>
<td>78.32 *</td>
<td>77.45 *</td>
</tr>
<tr>
<td>SC</td>
<td>77.51 *</td>
<td>88.95 *</td>
<td>82.84 *</td>
</tr>
<tr>
<td>Multitask(SC+LM)</td>
<td>80.21 *</td>
<td>91.05 *</td>
<td>85.25 *</td>
</tr>
<tr>
<td>SC+Self-Attention</td>
<td>80.18</td>
<td>88.96</td>
<td>84.34</td>
</tr>
<tr>
<td>SC+POS</td>
<td>77.27</td>
<td>93.01</td>
<td>84.41</td>
</tr>
<tr>
<td>SC+Self-Attention+POS</td>
<td>80.44</td>
<td>91.69</td>
<td>85.70</td>
</tr>
</tbody>
</table>

\* denotes the results are cited from \cite{09}.

A component is judged to be correct only if both the boundary and the tag exactly match the ground-truth.

C. EVALUATION OF SIMILE SENTENCE CLASSIFICATION TASK

In this section, we will show the experimental results of our proposed model and previous state-of-the-art models for the simile sentence classification task in CSRD dataset. Table 4 gives the experimental results, which are reported with the precision (P), recall (R), and F1 score (F1).

1) COMPARISONS

The comparison models are described as follows:

**Random Forest.** This method follows the model proposed by \cite{01}, which uses manually designed features based on the candidate simile components to distinguish similes from literals. The features include: 1) bag-of-words; 2) corresponding occurrence within constituents; 3) word embeddings of the candidate simile components.

**Random Forest (POS).** This model is based on \cite{02}, which uses handcrafted features around the comparator to recognize the similes. The handcrafted features involve two parts: 1) the tokens and POS tags of the words around the comparator within a fixed window; 2) the tokens, POS tags, and dependency relation tags of the words that have dependency relations with the comparator.

**SC.** This model is based on \cite{09}, which is the state-of-the-art single-task neural model for simile sentence classification.

**Multitask(SC+LM).** This model is based on \cite{09}, which is previous multitask learning model for simile sentence classification. It jointly optimizes two tasks including simile sentence classification, and language modeling.

**SC+Self-Attention.** This model is a simplified version of our proposed model for the simile sentence classification task by ignoring the explicit POS embedding.

**SC+POS.** This model is a simplified version of our proposed model without the self-attention mechanism.

**SC+Self-Attention+POS.** This model is our proposed model for simile sentence classification task, which include the feature extractor and the simile sentence classification module. The full architecture is described in Section 3.
2) RESULTS AND DISCUSSION

Table 4 shows the performances of all comparisons for the simile sentence classification task. The results are reported with the precision (P), recall (R), and F1 score (F1). In the first block, we show the experimental results of the feature based methods. In the second block, we give the performance of previous state-of-the-art single-task and multitask learning method. In the last block, we report the performances of our model (Self-Attention+POS) and its simplified versions.

Random Forest performs poorly than Random Forest (POS). The reason may be that the classification seriously depends on the accuracy of the candidate simile components recognition, which brings error propagation. In addition, Random Forest based on related component features only has difficulty to identify the exact POS tag for each token, which further decreases the performance. Random Forest (POS) considers the explicit POS information within fixed context window and achieves better performances than Random Forest. This confirms our intuition that POS information is effective for the simile recognition task.

Moreover, we have other observations:

- Neural network based methods can outperform feature based methods in the simile sentence classification task.
- Both neural network based method and feature based method can benefit from the POS information. Random Forest (POS) considers the explicit POS information within a fixed context window and outperforms Random Forest significantly. SC+POS exploits the explicit POS embeddings and achieves the better performance than SC. It demonstrates that POS information is effective for simile sentence classification. And the POS information mainly improves the recall.
- The self-attention mechanism is effective for the simile sentence classification task. SC+Self-Attention applies the self-attention mechanism and obtains superior performances than SC. It illustrates the effectiveness of self-attention mechanism for simile classification. The self-attention mechanism mainly improve the precision.
- Our proposed model for simile sentence classification task can outperform previous state-of-the-art single-task model significantly. The improvement of F1 score can reach to 2.86% compared with SC.

D. EVALUATING SIMILE COMPONENT EXTRACTION

In this section, we will show the experimental results of our proposed model and previous state-of-the-art models for the simile component extraction task. Table 5 shows the experimental results, which are reported with the pair-wise precision (PP), recall (PR), and F1 score (PF1).

1) COMPARISONS

We compare our proposed model with the following methods for the simile component extraction task.

CRF. Conditional Random Field (CRF) is a standard solution for sequence labeling problem. Since the simile component extraction task is a sequence labeling problem, CRF model is used as our baseline for simile component extraction task, which is a feature based method.

CE. This model is based on [10], which is state-of-the-art single-task model for simile component extraction. It is a neural network based method.

Multitask(CE+LM). This model is based on [10], which is previous state-of-the-art multitask learning model. It jointly optimizes two tasks: simile component extraction and language modeling.

CE+Self-Attention. This model is a simplified version of our proposed model for the simile component extraction task without integrating the explicit POS information.

CE+POS. This model is a simplified version of our proposed model by ignoring the self-attention mechanism.

CE+Self-Attention+POS. This model is our proposed model for simile component extraction task, which applies the POS embeddings and self-attention mechanism to CE. The full architecture is described in Section III.

2) RESULTS AND DISCUSSION

Table 5 shows the results for simile component extraction. The results are reported with the pair-wise level precision (PP), recall (PR), and F1 score (PF1). In the first block, we show the experimental results of the feature based methods. In the second block, we give the performance of previous state-of-the-art methods. In the last block, we report the performance of our proposed model (Self-Attention+POS) and its simplified versions.

According to the experimental results of Table 5, we have following observations:

- Neural network based methods outperform feature based methods in the simile component extraction task.
- The POS information is also effective for simile component extraction. CE+POS considers the POS information and improves PF1 score from 59.98% to 64.00% as compared with CE, which indicates the POS information is very effective for simile component extractor.
- The Self-Attention mechanism is effective for simile component extraction. When compared with CE, CE+Self-Attention improves the performance with the help of information learned from self-attention, which proves that the self-attention mechanism is effective for simile component extractor.
- Our proposed model for simile component extraction task can outperform both state-of-the-art single-task

<table>
<thead>
<tr>
<th>Models</th>
<th>PP(%)</th>
<th>PR(%)</th>
<th>PF1(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF</td>
<td>31.57*</td>
<td>36.98*</td>
<td>34.06*</td>
</tr>
<tr>
<td>CE</td>
<td>55.80*</td>
<td>64.89*</td>
<td>59.98*</td>
</tr>
<tr>
<td>Multitask(CE+LM)</td>
<td>57.41*</td>
<td>70.15*</td>
<td>63.06*</td>
</tr>
<tr>
<td>CE+Self-Attention</td>
<td>55.23</td>
<td>71.89</td>
<td>62.47</td>
</tr>
<tr>
<td>CE+POS</td>
<td>57.79</td>
<td>71.70</td>
<td>64.00</td>
</tr>
<tr>
<td>CE+Self-Attention+POS</td>
<td>58.91</td>
<td>74.65</td>
<td>65.85</td>
</tr>
</tbody>
</table>

* denotes the results are cited from [10].

TABLE 5. The experimental results for simile component extraction task.
TABLE 6. Examples for the effectiveness of the POS information integrating and the self-attention mechanism.

<table>
<thead>
<tr>
<th>ID</th>
<th>Model</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Golds</td>
<td>Please drag the [ground], as bright as a [mirror], and polish the window</td>
</tr>
<tr>
<td></td>
<td>CE</td>
<td>Please drag the ground as bright as a [mirror], and polish the window</td>
</tr>
<tr>
<td></td>
<td>CE+Self-Attention</td>
<td>Please drag the ground as bright as a [mirror], and polish the window</td>
</tr>
<tr>
<td></td>
<td>CE+POS</td>
<td>Please drag the ground as bright as a [mirror], and polish the window</td>
</tr>
<tr>
<td></td>
<td>CE+Self-attention+POS</td>
<td>Please drag the ground as bright as a [mirror], and polish the window</td>
</tr>
</tbody>
</table>
|    |                                | **There are two small [pockets], on either side of the bag, they are like two big [ears],** book两侧还有两个小的[兜]，它们就像两个大[耳朵]。
|    |                                | **There are two small pockets on either side of the bag, they are like two big [ears]** book两侧还有两个小的兜，它们就像两个大耳朵。
|    |                                | **There are two small pockets on either side of the bag, they are like two big [ears]** book两侧还有两个小的兜，它们就像两个大耳朵。
|    |                                | **There are two small pockets on either side of the bag, they are like two big [ears]** book两侧还有两个小的兜，它们就像两个大耳朵。
|    |                                | **There are two small pockets on either side of the bag, they are like two big [ears]** book两侧还有两个小的兜，它们就像两个大耳朵。|
| S2 | Golds                          | [It], keeps flapping its wings, looking left and right, like a very funny [clown]， |
|    | CE                             | [It], keeps flapping its wings, looking left and right, like a very funny clown， |
|    | CE+Self-Attention               | [It], keeps flapping its wings, looking left and right, like a very funny clown， |
|    | CE+POS                         | [It], keeps flapping its wings, looking left and right, like a very funny clown， |
|    | CE+Self-Attention+POS          | [It], keeps flapping its wings, looking left and right, like a very funny clown， |
|    |                                | **I have a cute little [rabbit] with all white, like a [small snowball]** 我有一只可爱的小[兔子]，全身雪白雪白的，像一个[小雪球]。 |
|    |                                | **I have a cute little rabbit with all white, like a [small snowball]** 我有一只可爱的小兔子，全身雪白雪白的，像一个[小雪球]。 |
|    |                                | **I have a cute little rabbit with all white, like a [small snowball]** 我有一只可爱的小兔子，全身雪白雪白的，像一个[小雪球]。 |
|    |                                | **I have a cute little rabbit with all white, like a [small snowball]** 我有一只可爱的小兔子，全身雪白雪白的，像一个[小雪球]。 |
|    |                                | **I have a cute little rabbit with all white, like a [small snowball]** 我有一只可爱的小兔子，全身雪白雪白的，像一个[小雪球]。 |
| S3 | Golds                          | The [world], the [reality], like the [servant]， is growing up with me  |
|    | CE                             | The world, the reality, like the [servant]， is growing up with me  |
|    | CE+Self-Attention               | The world, the reality, like the [servant]， is growing up with me  |
|    | CE+POS                         | The world, the reality, like the [servant]， is growing up with me  |
|    | CE+Self-Attention+POS          | The world, the reality, like the [servant]， is growing up with me  |

\( \dagger \) denotes the tenor and \( \sigma \) denotes the vehicle.

and multitask learning methods significantly. According to the results, our proposed model achieves 5.87% improvement compared with previous state-of-the-art single-task model CE and 2.79% improvement compared with the state-of-the-art multitask learning method Multitask(CE+LM).

3) ABOLATION STUDY
To further explore the efficacy of the key components, we perform ablation study as noted in the last block of Table 4 and Table 5. The results show that both the simile sentence classification task and the simile components extraction task can benefit from the combination of the self-attention...
mechanism and the explicit POS information integrating. Both the self-attention mechanism and the POS information is helpful for improving the performance for simile recognition. The self-attention mechanism mainly improves the performance of precision in the simile sentence classification task while it improves the recall in simile components extraction. Moreover, the explicit POS information integrating improves the recall in simile sentence classification while it can improves both the precision and recall in simile components extraction. We report and discuss the ablation experimental results from the following aspects.

**The effect of the self-attention mechanism.** In the simile components extraction task, CE+Self-Attention improves PF1 score from 59.98% to 62.47% as compared with CE, and CE+Self-Attention+POS gets about 1.9% absolute gains as compared with CE+POS. In the simile sentence classification task, SC+Self-Attention gains about 1.5% F1 improvement as compared with SC, and SC+Self-Attention+POS achieves about 1.3% improvement compared with SC+POS. It illustrates the effectiveness of self-attention mechanism both for classification and extraction.

**The effect of the explicit POS information integrating.** By integrating the explicit POS information, CE+POS boosts the performance as compared with CE, showing about 4% PF1 improvement in the simile components extraction task. SC+POS improves F1 score from 82.84% to 84.41% as compared with SC while SC+POS+Self-Attention achieves about 1.36% improvement as compared with SC+Self-Attention in the simile sentence classification task. It demonstrates that the explicit POS information integration is effective for both the simile sentence classification task and the simile components extraction task.

**The effect of combining the self-attention mechanism and the explicit POS information.** SC+Self-Attention+POS can achieve the best performance among SC+Self-Attention and SC+POS. The improvement of F1 score can reach to 2.86% compared with the best single task model SC. CE+Self-Attention+POS can achieve the best performance among CE+Self-Attention and CE+POS. The improvement of PF1 score can reach to 5.87% compared with previous state-of-the-art single task model CE.

**The impact of the POS tagger to the final results.** To further explore the impact of the explicit POS information on the simile recognition task, we separately conduct POS tagging with different POS taggers, such as HITLTP, THULAC, PkuSeg, Jieba, SnowNLP, FoolNLTK, and Stanford CoreNLP. As shown in Table 7 and Table 8, all of these POS tagger based models are outperform previous state-of-the-art single-task model significantly. Different POS taggers perform differently. The reason may be that both the classification task and extraction task depend on the explicit POS information, while POS tagging is based on the word segmentation conducted by the corresponding POS tagger, which brings difference. HITLTP performs best than the others both in the simile sentence classification task and the simile component extraction task, while the remaining POS taggers cannot align the POS tag sequence with the input token sequence split in advance by the HITLTP toolkit in CSRD dataset, which brings error propagation. Stanford CoreNLP toolkit performs poorly. The main reason is that thisS tool may not support Chinese well.

**The effect of the expanded LSTM.** According to Table 9 and Table 10, all standard LSTM based models perform worse on both subtasks than the corresponding BiLSTM based model. The main reason is that BiLSTM can leverage the past and future information while LSTM can only leverage the past information. It verifies that BiLSTM is effective for the simile recognition task. In addition, in those standard LSTM based models, the self-attention mechanism based model outperform the POS tagger based model. This also illustrates the effectiveness of the self-attention mechanism. The main reason may be that the self-attention mechanism can construct the direct relation between any two words in a sentence, which make it been less affected by the feature extractor based on the standard LSTM. Moreover, we find that the simile sentence classification task is less affected by the standard LSTM than the simile component extraction task. The reason may be that text categorization tasks are easier to learn than sequence labeling tasks.

### Tables

**Table 7. The experimental results for simile sentence classification task with different POS taggers.**

<table>
<thead>
<tr>
<th>POS taggers</th>
<th>P(%)</th>
<th>R(%)</th>
<th>F1(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITLTP</td>
<td>80.44</td>
<td>91.69</td>
<td>85.70</td>
</tr>
<tr>
<td>THULAC</td>
<td>79.30</td>
<td>90.07</td>
<td>84.33</td>
</tr>
<tr>
<td>PkuSeg</td>
<td>78.68</td>
<td>91.59</td>
<td>84.64</td>
</tr>
<tr>
<td>SnowNLP</td>
<td>80.43</td>
<td>90.37</td>
<td>85.11</td>
</tr>
<tr>
<td>Jieba</td>
<td>80.27</td>
<td>89.87</td>
<td>84.80</td>
</tr>
<tr>
<td>FoolNLTK</td>
<td>81.02</td>
<td>88.65</td>
<td>84.66</td>
</tr>
<tr>
<td>Stanford</td>
<td>79.15</td>
<td>88.86</td>
<td>83.72</td>
</tr>
</tbody>
</table>

**Table 8. The experimental results for simile component extraction task with different POS taggers.**

<table>
<thead>
<tr>
<th>POS taggers</th>
<th>PP(%)</th>
<th>PR(%)</th>
<th>F1(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITLTP</td>
<td>58.91</td>
<td>74.65</td>
<td>65.85</td>
</tr>
<tr>
<td>THULAC</td>
<td>56.46</td>
<td>74.56</td>
<td>64.26</td>
</tr>
<tr>
<td>PkuSeg</td>
<td>55.88</td>
<td>72.58</td>
<td>63.15</td>
</tr>
<tr>
<td>SnowNLP</td>
<td>57.33</td>
<td>72.09</td>
<td>63.87</td>
</tr>
<tr>
<td>Jieba</td>
<td>57.66</td>
<td>75.74</td>
<td>64.47</td>
</tr>
<tr>
<td>FoolNLTK</td>
<td>56.45</td>
<td>75.94</td>
<td>64.76</td>
</tr>
<tr>
<td>Stanford</td>
<td>52.36</td>
<td>71.01</td>
<td>60.28</td>
</tr>
</tbody>
</table>

---

5. https://github.com/lancopku/pkuseg-python
6. https://github.com/snowfly/snownlp
7. https://github.com/fxsjy/jieba
8. https://github.com/rockyzhengwu/FoolNLTK
9. https://stanfordnlp.github.io/CoreNLP

---

4) **DETAILED ANALYSIS**

**a: CASE STUDY**

POS information is very important for simile component extraction task, especially when the simile components have...
P. Zhang et al.: Combining POS Tags and Self-Attention Mechanism for Simile Recognition

FIGURE 2. The visualization of the self-attention mechanism of S3 and S4 in Table 6. (a): The visualization of S3. (b): The visualization of S4.

TABLE 9. The experimental results for simile sentence classification task with different LSTM.

<table>
<thead>
<tr>
<th>Models</th>
<th>P(%)</th>
<th>R(%)</th>
<th>F1(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC (LSTM)</td>
<td>77.55</td>
<td>88.55</td>
<td>82.69</td>
</tr>
<tr>
<td>SC+Self-Attention (LSTM)</td>
<td>79.65</td>
<td>89.45</td>
<td>83.82</td>
</tr>
<tr>
<td>SC+POS (LSTM)</td>
<td>78.15</td>
<td>89.87</td>
<td>83.60</td>
</tr>
<tr>
<td>SC+Self-Attention+POS (LSTM)</td>
<td>79.77</td>
<td>89.46</td>
<td>84.34</td>
</tr>
<tr>
<td>SC+POS</td>
<td>80.18</td>
<td>93.01</td>
<td>84.41</td>
</tr>
<tr>
<td>SC+Self-Attention+POS</td>
<td>80.44</td>
<td>91.69</td>
<td>85.70</td>
</tr>
</tbody>
</table>

* denotes the results are cited from [10].

TABLE 10. The experimental results for simile component extraction task with different LSTM.

<table>
<thead>
<tr>
<th>Models</th>
<th>PP(%)</th>
<th>PR(%)</th>
<th>PFI(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE (LSTM)</td>
<td>22.80</td>
<td>34.22</td>
<td>27.37</td>
</tr>
<tr>
<td>CE+Self-Attention (LSTM)</td>
<td>52.46</td>
<td>66.17</td>
<td>58.53</td>
</tr>
<tr>
<td>CE+POS (LSTM)</td>
<td>17.86</td>
<td>24.06</td>
<td>20.50</td>
</tr>
<tr>
<td>CE+Self-Attention+POS (LSTM)</td>
<td>56.09</td>
<td>70.81</td>
<td>62.60</td>
</tr>
<tr>
<td>CE+POS</td>
<td>55.80</td>
<td>64.89</td>
<td>59.98</td>
</tr>
<tr>
<td>CE+Self-Attention</td>
<td>55.23</td>
<td>71.89</td>
<td>62.47</td>
</tr>
<tr>
<td>CE+POS</td>
<td>57.79</td>
<td>71.70</td>
<td>64.00</td>
</tr>
<tr>
<td>CE+Self-Attention+POS</td>
<td>58.91</td>
<td>74.65</td>
<td>65.85</td>
</tr>
</tbody>
</table>

* denotes the results are cited from [10].

We use two sentences in CSRD test set as examples to reveal the effectiveness of our proposed model. As shown in Table 6, the POS tagger based models (e.g., CE+POS, CE+Self-Attention+POS) are able to recognize both the tenor and the vehicle correctly while the models without the explicit POS information (e.g., CE, CE+Self-Attention) can only recognize “镜子 (mirror)” as vehicle. In Chinese, the tenor “地” is often an adverb suffix (“-ly”), while it is a noun (“ground”) in the first example. When making tagging for the target tenor “地”, with the help of the explicit POS information integrating, the POS tagger based models can identify the POS information of these words with different POS tag in different context precisely. Then, they recognize the simile components correctly. In the second example, the tenor “地” is often a verb (“hold”), while it is a noun (“packet”) in this sentence. The POS tagger based models exploit the explicit POS information to erase the information gap, and then make correct simile tagging for the word “地”, while CE and CE+Self-Attention are failed to identify its simile tags. It demonstrates that the explicit POS information integrating is effective for simile recognition.

In the third example (S3), CE cannot correctly extract the vehicle “小丑 (clown)” in the simile sentence. Even with the help of the explicit POS information integrating, CE+POS is failed to recognize the simile tags of this word. The main reason may be that the vehicle “小丑 (clown)” and the tenor “它 (it)” are far apart. The self-Attention mechanism learns the dependencies between “小丑 (clown)” and “它 (it)”, therefore, the self-attention based models (e.g., CE+Self-Attention, CE+Self-Attention+POS) can correctly recognize the word as tenor. As shown in the Figure 2(a), when labelling the word “小丑 (clown)”, the self-attention mechanism can explicitly learn the dependencies with “它 (it)”. Thus, our model enables to correctly recognize the word as the vehicle. In the fourth example (S4), when making tagging with the word “兔子 (rabbit)”, if the extractor only focus on itself, it is difficult to identify the simile tag of this word. As shown in the Figure 2(b), our model can learn the direct dependencies between this word and “小雪球 (small snowball)”, and then make a correct decision. It shows that self-attention mechanism is also effective for the simile component extraction task.

Although both the explicit POS information integrating and the self-attention mechanism based models are effective for the simile component extraction task, there is still room for improvement. As shown in the last example (S5) in the Table 6, all models cannot capture the tenors completely.
when the targeted sentences have multiple tenors or vehicles. This implies that it is a challenging to extract complete tenors and vehicles from complex sentences.

**b: ERROR ANALYSIS**

According to the results of Table 4 and Table 5, while the greater improvement proves that our model is effective for the simile components extraction task. However, the improvement between our proposed model and the multitask learning method is small (0.45%). The main reason is that the simile sentence classification task highly depends the quality of the context information. However, language modeling in the multitask learning methods can learn better semantic and syntactic structure information, which is not sufficient in our proposed model. In the future, we will integrate the language modeling into our framework to generate more sufficient context representations to boost the performance.

In addition, the overall performance for the simile components extraction task is relatively lower. The main reason is that the simile components only exist in the simile sentences. Previous work demonstrates that it is difficult to recognize simile components in a sentence without the sentence label (simile or literal) [10]. According to the error cases, we notice that our proposed models will recognize the simile components in literals. To relief this problem, we will study the multitask learning model to find a effective method to combine the two subtasks in the future.

V. CONCLUSION

In this paper, we propose a neural network framework, which exploit the explicit POS information. Specifically, we introduce the self-attention mechanism in our model to better capture the dependencies between arbitrary two words in sentences. The experimental results on widely used dataset showed that our model perform better than the previous best single-task model on the simile sentence classification task, and outperform previous state-of-the-art models in the simile component extraction task. Both the explicit POS information and the self-attention mechanism are effective in the simile recognition task. The self-attention mechanism mainly improves the precision of our models, while the explicit POS information improves the recall.

In the future, we plan to expand our research from the following aspects: 1) introducing the language modeling into simile recognition to sufficient context representations to boost the overall performance; 2) integrating more syntactic cues (e.g., dependency parsing analysis) into the simile component extraction task to better extract simile components; 3) investigating multitask learning for simile recognition to better combine the two subtasks; 4) integrating simile components extraction into the dialogue system to generate the informative responses.

REFERENCES


[42] PENGFEI ZHANG was born in Shangrao, Jiangxi, China, in 1995. He received the B.S. degree in computer science and technology from the Jiangxi University of Finance and Economics, Nanchang, China, in 2017. He is currently pursuing the M.S. degree in software engineering with the South China University of Technology, Guangzhou, China. His research interest includes similar recognition, named entity recognition and dialogue systems. His awards include the Third Place (Third Prize) of the 2018 SMP Evaluation of Chinese Human–Computer Dialogue Technology (SMP2018-ECDT) and the Third Prize of the 12th Intel Cup College Students Innovation and Entrepreneurship Competition, in 2019.

[43] YI CAI (M’15) was born in Beihai, Guangxi, China, in 1980. He received the B.S. degree in polymer materials from Sichuan University, Chengdu, China, in 2003, and the M.S. and Ph.D. degrees in computer science from The Chinese University of Hong Kong, Hong Kong, in 2005 and 2009, respectively.

He was a Visiting Scholar with KEG, Tsinghua University and Nanyang Technological University. From 2009 to 2011, he held a postdoctoral position with the City University of Hong Kong, Hong Kong. He is currently a Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. He has published 90 articles, including CIKM, ER, WI, WISE, ICTAI, and JCST, and RW books. His research interests include recommendation systems, personalized search, semantic Web, and data mining. He reviewed articles from conferences and journals related to information retrieval, semantic Web, recommender systems, data mining, and database, including TKDE, TOIT, WWWJ, JCST, CIKM, and ER. He is a Program Committee Member of conferences, including ICWE, WAIM, ICEBE, and NDBC.

[44] JUNYING CHEN (S’11–M’13) received the B.E. degree in electronic and information engineering from Zhejiang University, in 2007, and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong, in 2013. She is currently an Associate Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. Her research interests include high-performance computing, deep learning, medical imaging, and image processing.

[45] WENHAO CHEN received the B.E. degree in computer science and technology from Central South University, in 2007, and the M.Phil. degree in computer science and engineering from The Chinese University of Hong Kong, in 2010. He is currently pursuing the Ph.D. degree with the Management Science Department, City University of Hong Kong. He has many years of industrial experience in data science and has worked in Bloomberg for financial data and news data products. His research interests include financial data mining and machine learning.

[46] HENGJIE SONG received the B.Sc. degree in electrical engineering and the M.Sc. degree in aerospace computer design from the Harbin Institute of Technology, in 1998 and 2002, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore. He is currently a Professor with the School of Software Engineering, South China University of Technology, Guangzhou, China. His research interests include data mining and artificial intelligence. He has served as a PC Reviewer for the 21th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.