Incorporating Sememes into Chinese Definition Modeling

Liner Yang, Cunliang Kong, Yun Chen, Yang Liu, Qinan Fan, and Erhong Yang

***Abstract*—Chinese definition modeling is a challenging task that generates a dictionary definition in Chinese for a given Chinese word. To accomplish this task, we built two novel datasets based on Chinese Concept Dictionary (CCD) and Chinese WordNet (CWN) respectively. Each dataset contains triples of a word, sememes, and a corresponding definition. We present two novel models to improve Chinese definition modeling: the Adaptive-Attention model (AAM) and the Self- and Adaptive-Attention Model (SAAM). AAM successfully incorporates sememes for generating the definition with an adaptive attention mechanism. It has the capability to decide which sememes to focus on and when to pay attention to sememes. SAAM further replaces recurrent connections in AAM with self-attention and relies entirely on the attention mechanism, reducing the path length between word, sememes and definition. Experiments on both datasets demonstrate that by incorporating sememes, our model can generate definitions with more concrete information. And the best model that we proposed outperforms the state-of-the-art method by a large margin on both datasets.**

***Index Terms*—Definition modeling, knowledge bases, self-attention, sememes.**

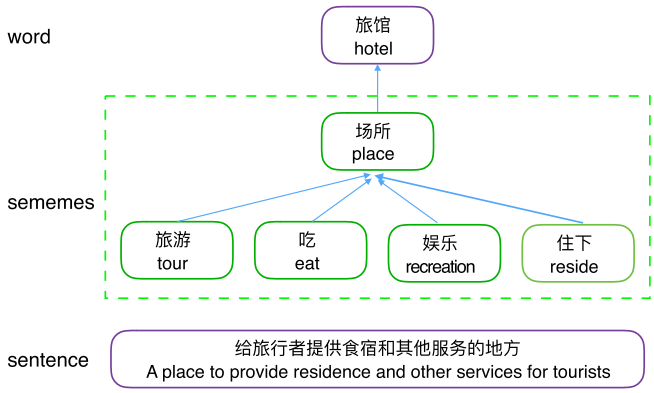

Fig. 1. An example of the CDM dataset. The word "旅馆" (hotel) has five sememes, which are "场所" (place), "旅游" (tour), "吃" (eat), "娱乐" (recreation) and "住下" (reside).

I. Introduction

CHINESE definition modeling refers to the task of automatically generating a Chinese definition for a specific word. The definition generated here can describe the semantic meanings of the word. This task can be employed to assist in the compilation of dictionaries

Most people have encountered words they don't recognize while reading, especially for second language learners like Chinese as a Foreign Language (CFL) learners. At this time, people often turn to dictionaries for help. However, already compiled dictionaries are often not updated in time and won't include words that relatively new. On the other hand, second language learners are often not familiar with the language they are learning, and using dictionaries might be difficult. In the contrast, it may be a good choice to generate definitions of words. Therefore, it is of great significance to study how to automatically generate definitions.

Definition modeling was first proposed by Noraset *et al.* [1] as a tool to evaluate different word embeddings. Gadetsky *et al.* [2] extended the work by incorporating word sense disambiguation to generate context-aware word definition. Both methods are based on recurrent neural network encoder-decoder framework without attention. In contrast, this paper formulates the definition modeling task as an automatic way to accelerate dictionary compilation.

In this work, we introduce two novel datasets built from CCD [3], [4] and CWN [5] respectively. These two datasets consists of 131,633 entries in total where each entry contains a word, the sememes of a specific word sense, and the definition of the same word sense in Chinese. Sememes are **minimum semantic units** of word meanings, and the meaning of each word sense is typically composed of several sememes, as illustrated in Figure 1. For a given word sense, we annotate the sememes according to HowNet [6]. Since sememes have been widely used in improving word representation learning [7] and word similarity computation [8], we argue that sememes can benefit the task of definition modeling.

We propose two novel models to incorporate sememes into Chinese definition modeling: the Adaptive-Attention Model (AAM) and the Self- and Adaptive-Attention Model (SAAM).

Manuscript received August 29, 2019; revised January 14, 2020; accepted March 20, 2020. Date of publication April 20, 2020; date of current version June 5, 2020. This work is supported in part by the funds of Beijing Advanced Innovation Center for Language Resources under Grant TYZ19005, in part by the National Key R&D Program of China under Grant 2018YFB1005103, in part by the National Natural Science Foundation of China under Grants 61925601, 61761166008 and in part by the Research Project of the National Language Commission under Grant ZDI135-105. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Imed Zitouni. *(Corresponding author: Liner Yang.)*

Liner Yang, Cunliang Kong, Qinan Fan, and Erhong Yang are with the Beijing Language and Culture University, Beijing 100083, China (e-mail: lineryang@gmail.com; cunliang.kong@outlook.com; blcufqn@hotmail.com; yerhong@blcu.edu.cn).

Yun Chen is with the Shanghai University of Finance and Economics, Shanghai 200433, China (e-mail: yunchen@mail.shufe.edu.cn).

Yang Liu is with the Tsinghua University, Beijing 100084, China (e-mail: liuyang2011@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TASLP.2020.2987754

Both models are based on the encoder-decoder framework. The encoder maps word and sememes into a sequence of continuous representations, and the decoder then attends to the output of the encoder and generates the definition one word at a time. Different from the vanilla attention mechanism, the decoder of both models employ the *adaptive attention mechanism* to decide which sememes to focus on and when to pay attention to sememes at one time [9]. Following Noraset *et al.* [1] and Gadetsky *et al.* [2], the AAM is built using recurrent neural networks (RNNs). However, recent works demonstrate that attention-based architecture that entirely eliminates recurrent connections can obtain new state-of-the-art in neural machine translation [10], constituency parsing [11] and semantic role labeling [12]. In the SAAM, we replace the LSTM-based encoder and decoder with an architecture based on self-attention. This fully attention-based model allows for more parallelization, reduces the path length between word, sememes and the definition, and can reach a new state-of-the-art on the definition modeling task. To the best of our knowledge, this is the first work to introduce the attention mechanism and utilize external resource for the definition modeling task.

Experiments on both novel datasets show that our proposed AAM and SAAM outperform the state-of-the-art approach with a large margin. After the model generation, we randomly selected part of the data from the generated results for manual evaluation. The manual evaluation results show that by efficiently incorporating sememes, the SAAM can generate more concrete and accurate definitions.

II. Methodology

The definition modeling task is to generate an explanatory sentence for the interpreted word. For example, given the word "旅馆" (hotel), a model should generate a sentence like this: "给旅行者提供食宿和其他服务的地方" (A place to provide residence and other services for tourists). Since distributed representations of words have been shown to capture lexical syntax and semantics, it is intuitive to employ word embeddings to generate natural language definitions.

Previously, Noraset *et al.* [1] proposed several model architectures to generate a definition according to the distributed representation of a word. We briefly summarize their model with the best performance in Section II-A and adopt it as our baseline model.

Inspired by the works that use sememes to improve word representation learning [7] and word similarity computation [8], we propose the idea of incorporating sememes into definition modeling. Sememes can provide additional semantic information for the task. As shown in Figure 1, sememes are highly correlated to the definition. For example, the sememe "场所" (place) is related with the word "地方" (place) of the definition, and the sememe "旅游" (tour) is correlated to the word "旅行者" (tourists) of the definition. Therefore, to make full use of the sememes in datasets, we propose AAM and SAAM for the task, in Section II-B and Section II-C, respectively.

A. Baseline Model

The baseline model [1] is implemented with a recurrent neural network based encoder-decoder framework. Without utilizing the information of sememes, it learns a probabilistic mapping $P(y|x)$ from the word x to be defined to a definition $y = [y_1, \ldots, y_T]$, in which y_t is the t-th word of definition y.

More concretely, given a word x to be defined, the encoder reads the word and generates its word embedding $\boldsymbol{x}$ as the encoded information. Afterward, the decoder computes the conditional probability of each definition word y_t depending on the previous definition words $y_{<t}$, as well as the word being defined x, i.e., $P(y_t|y_{<t}, x)$. $P(y_t|y_{<t}, x)$ is given as:

$$P(y_t|y_{<t}, x) \propto \exp\left(g(y_t, \boldsymbol{z}_t)\right) \tag{1}$$

$$\boldsymbol{z}_t = f(\boldsymbol{z}_{t-1}, y_{t-1}, \boldsymbol{x}), \tag{2}$$

where $\boldsymbol{z}_t$ is the decoder's hidden state at time t, f and g are a recurrent nonlinear function such as LSTM and GRU, and $\boldsymbol{x}$ is the embedding of the word being defined. Then the probability of $P(y|x)$ can be computed according to the probability chain rule:

$$P(y|x) = \prod_{t=1}^{T} P(y_t|y_{<t}, x) \tag{3}$$

We denote all the parameters in the model as θ and the definition corpus as $D_{x,y}$, which is a set of word-definition pairs. Then the model parameters can be learned by maximizing the log-likelihood:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{\langle x,y\rangle \in D_{x,y}} \log P(y|x;\theta) \tag{4}$$

B. Adaptive-Attention Model

Our proposed model aims to incorporate sememes into the definition modeling task. Given the word to be defined x and its corresponding sememes $s = [s_1, \ldots, s_N]$, we define the probability of generating the definition $y = [y_1, \ldots, y_t]$ as:

$$P(y|x, s) = \prod_{t=1}^{T} P(y_t|y_{<t}, x, s) \tag{5}$$

Similar to Eq. 4, we can maximize the log-likelihood with the definition corpus $D_{x,s,y}$ to learn model parameters:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{\langle x,s,y\rangle \in D_{x,s,y}} \log P(y|x, s;\theta) \tag{6}$$

The probability $P(y|x, s)$ can be implemented with an adaptive attention based encoder-decoder framework, which we call Adaptive-Attention Model (AAM). The new architecture consists of a bidirectional RNN as the encoder and a RNN decoder that adaptively attends to the sememes during decoding a definition.

a) Encoder: Similar to Bahdanau *et al.* [13], the encoder is a bidirectional RNN, consisting of forward and backward RNNs. Given the word to be defined x and its corresponding sememes $s = [s_1, \ldots, s_N]$, we define the input sequence of vectors for the

encoder as $\boldsymbol{v} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]$. The vector $\boldsymbol{v}_n$ is given as follows:

$$\boldsymbol{v}_n = [\boldsymbol{x}; \boldsymbol{s}_n] \tag{7}$$

where $\boldsymbol{x}$ is the vector representation of the word x, $\boldsymbol{s}_n$ is the vector representation of the n-th sememe s_n, and $[\boldsymbol{a}; \boldsymbol{b}]$ denote concatenation of vector $\boldsymbol{a}$ and $\boldsymbol{b}$.

The forward RNN $\overrightarrow{f}$ reads the input sequence of vectors from $\boldsymbol{v}_1$ to $\boldsymbol{v}_N$ and calculates a forward hidden state for position n as:

$$\overrightarrow{\boldsymbol{h}_n} = f(\boldsymbol{v}_n, \overrightarrow{\boldsymbol{h}_{n-1}}) \tag{8}$$

where f is an LSTM or GRU. Similarly, the backward RNN $\overleftarrow{f}$ reads the input sequence of vectors from $\boldsymbol{v}_N$ to $\boldsymbol{v}_1$ and obtain a backward hidden state for position n as:

$$\overleftarrow{\boldsymbol{h}_n} = f(\boldsymbol{v}_n, \overleftarrow{\boldsymbol{h}_{n+1}}) \tag{9}$$

In this way, we obtain a sequence of encoder hidden states $\boldsymbol{h} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N]$, by concatenating the forward hidden state $\overrightarrow{\boldsymbol{h}_n}$ and the backward one $\overleftarrow{\boldsymbol{h}_n}$ at each position n:

$$\boldsymbol{h}_n = \left[\overrightarrow{\boldsymbol{h}_n}; \overleftarrow{\boldsymbol{h}_n}\right] \tag{10}$$

The hidden state $\boldsymbol{h}_n$ captures the sememe- and word-aware information of the n-th sememe.

b) Decoder: As attention-based neural encoder-decoder frameworks have shown great success in image captioning [14], document summarization [15] and neural machine translation [13], it is natural to adopt the attention-based recurrent decoder in Bahdanau *et al.* [13] as our decoder. The vanilla attention attends to the sememes at every time step. However, not all words in the definition have corresponding sememes. For example, sememe "住下" (reside) could be useful when generating "食宿" (residence), but none of the sememes is useful when generating "提供" (provide). Besides, language correlations make the sememes unnecessary when generating words like "和" (and) and "给" (for).

Inspired by Lu *et al.* [9], we introduce the adaptive attention mechanism for the decoder. At each time step t, we summarize the time-varying sememes' information as sememe context, and the language model's information as LM context. Then, we use another attention to obtain the context vector, relying on either the sememe context or LM context.

More concretely, we define each conditional probability in Eq. 5 as:

$$P(y_t|y_{<t}, x, s) \propto \exp\left(g(y_t; \boldsymbol{z}_t)\right) \tag{11}$$

$$\boldsymbol{z}_t = f(\boldsymbol{z}_{t-1}, \boldsymbol{m}_{t-1}, y_{t-1}, \boldsymbol{c}_t) \tag{12}$$

$$\boldsymbol{c}_t = \text{Adaptive}(\boldsymbol{z}_{t-1}, \boldsymbol{m}_{t-1}, \boldsymbol{h}_{1:N}), \tag{13}$$

where $\boldsymbol{c}_t$ is the context vector from the output of the adaptive attention module at time t, $\boldsymbol{z}_t$ and $\boldsymbol{m}_t$ are the decoder's hidden state and memory cell at time t respectively.

Here we give details contained in the adaptive attention module. To obtain the context vector $\boldsymbol{c}_t$, we first compute the sememe context vector $\hat{\boldsymbol{c}}_t$ and the LM context $\boldsymbol{o}_t$. Similar to the vanilla attention, the sememe context $\hat{\boldsymbol{c}}_t$ is obtained with a soft attention mechanism as:

$$\hat{\boldsymbol{c}}_t = \sum_{n=1}^{N} \alpha_{tn} \boldsymbol{h}_n, \tag{14}$$

where

$$\alpha_{tn} = \frac{\exp(e_{tn})}{\sum_{i=1}^{N} \exp(e_{ti})}$$
$$e_{tn} = \boldsymbol{W}_{\hat{c}}^T [\boldsymbol{h}_n; \boldsymbol{m}_{t-1}] \tag{15}$$

Since the decoder's hidden states store syntax and semantic information for language modeling, we compute the LM context $\boldsymbol{o}_t$ with a gated unit, where the input is the definition word y_t and the previous hidden state $\boldsymbol{z}_{t-1}$:

$$\boldsymbol{o}_t = \boldsymbol{g}_t \odot \tanh(\boldsymbol{z}_{t-1})$$
$$\boldsymbol{g}_t = \sigma(\boldsymbol{W}_g [y_{t-1}; \boldsymbol{z}_{t-1}] + \boldsymbol{b}_g). \tag{16}$$

Once the sememe context vector $\hat{\boldsymbol{c}}_t$ and the LM context $\boldsymbol{o}_t$ are ready, we can generate the context vector with an adaptive attention layer as:

$$\boldsymbol{c}_t = \beta_t \boldsymbol{o}_t + (1 - \beta_t) \hat{\boldsymbol{c}}_t, \tag{17}$$

where

$$\beta_t = \frac{\exp(e_{to})}{\exp(e_{to}) + \exp(e_{t\hat{c}})}$$
$$e_{to} = \boldsymbol{w}_c^T [\boldsymbol{o}_t; \boldsymbol{z}_{t-1}]$$
$$e_{t\hat{c}} = \boldsymbol{w}_c^T [\hat{\boldsymbol{c}}_t; \boldsymbol{z}_{t-1}], \tag{18}$$

β_t is a scalar in range [0,1], which controls the relative importance of LM context and sememe context.

Once we obtain the context vector $\boldsymbol{c}_t$, we update the decoder's hidden state and generate the next word according to Eq. 12 and Eq. 11, respectively.

C. Self- and Adaptive-Attention Model

Recent works demonstrate that an architecture entirely based on attention can obtain new state-of-the-art in neural machine translation [10], constituency parsing [11] and semantic role labeling [12]. SAAM adopts similar architecture and replaces the recurrent connections in AAM with self-attention. Such architecture not only reduces the training time by allowing for more parallelization, but also learns better the dependency between word, sememes and tokens of the definition by reducing the path length between them.

a) Encoder: Given the word to be defined x and its corresponding ordered sememes $s = [s_1, \ldots, s_N]$, we combine them as the input sequence of embeddings for the encoder, i.e., $\boldsymbol{v} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]$. The n-th vector $\boldsymbol{v}_n$ is defined as:

$$\boldsymbol{v}_n = \begin{cases} \boldsymbol{x}, & n = 0 \\ \boldsymbol{s}_n, & n > 0 \end{cases} \tag{19}$$

where $\boldsymbol{x}$ is the vector representation of the given word x, and $\boldsymbol{s}_n$ is the vector representation of the n-th sememe s_n.

Although the input sequence is not time ordered, position n in the sequence carries some useful information. First, position

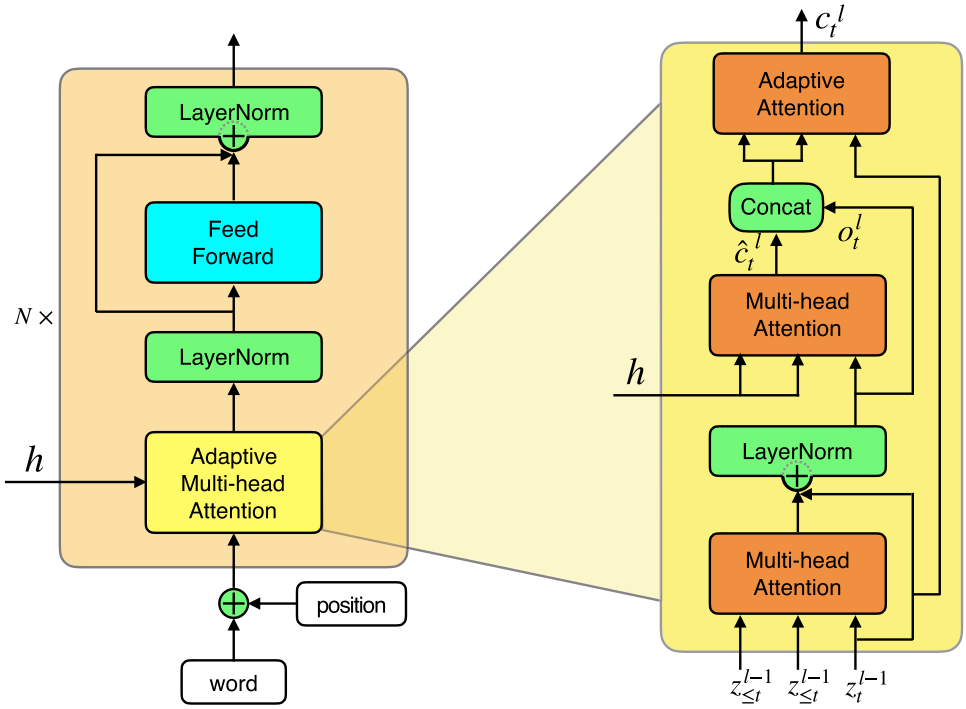

Fig. 2. An overview of the decoder for the SAAM. The left sub-figure shows our decoder contains N identical layers, where each layer contains two sublayer: adaptive multi-head attention layer and feed-forward layer. The right sub-figure shows how we perform the adaptive multi-head attention at layer l and time t for the decoder. $\boldsymbol{z}_t^l$ represents the hidden state of the decoder at layer l, time t. $\boldsymbol{h}$ denotes the output from the encoder stack. $\hat{\boldsymbol{c}}_t{}^l$ is the sememe context, while $\boldsymbol{o}_t^l$ is the LM context. $\boldsymbol{c}_t^l$ is the output of the adaptive multi-head attention layer at time t.

0 corresponds to the word to be defined, while other positions correspond to the sememes. Secondly, sememes are sorted into a logical order in HowNet. For example, as the first sememe of the word "旅馆" (hotel), the sememe "场所" (place) describes its most important aspect, namely, the definition of "旅馆" (hotel) should be "的地方" (a place for ...). Therefore, we add learned position embedding to the input embeddings for the encoder:

$$\boldsymbol{v}_n = \boldsymbol{v}_n + \boldsymbol{p}_n \tag{20}$$

where $\boldsymbol{p}_n$ is the position embedding that can be learned during training.

Then the vectors $\boldsymbol{v} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]$ are transformed by a stack of identical layers, where each layers consists of two sub-layers: multi-head self-attention layer and position-wise fully connected feed-forward layer. Each of the layers are connected by residual connections, followed by layer normalization [16]. We refer the readers to [10] for the detail of the layers. The output of the encoder stack is a sequence of hidden states, denoted as $\boldsymbol{h} = [\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_N]$.

b) Decoder: The decoder is also composed of a stack of identical layers. In Vaswani *et al.* [10], each layer includes three sublayers: masked multi-head self-attention layer, multi-head attention layer that attends over the output of the encoder stack and position-wise fully connected feed-forward layer. In our model, we replace the two multi-head attention layers with an adaptive multi-head attention layer. Similarly to the adaptive attention layer in AAM, the adaptive multi-head attention layer can adaptivelly decide which sememes to focus on and when to attend to sememes at each time and each layer. Figure 2 shows the architecture of the decoder.

Different from AAM that uses single head attention to obtain the sememe context and gate unit to obtain the LM context, SAAM utilizes multi-head attention to obtain both contexts. Multi-head attention performs multiple single head attentions in parallel with linearly projected keys, values and queries, and then combines the outputs of all heads to obtain the final attention result. We omit the detail here and refer the readers to [10]. Formally, given the hidden state $\boldsymbol{z}_t^{l-1}$ at time t, layer $l-1$ of the decoder, we obtain the LM context with multi-head self-attention:

$$\boldsymbol{o}_t^l = \mathrm{MultiHead}(\boldsymbol{z}_t^{l-1}, \boldsymbol{z}_{\leq t}^{l-1}, \boldsymbol{z}_{\leq t}^{l-1}) \tag{21}$$

where the decoder's hidden state $\boldsymbol{z}_t^{l-1}$ at time t, layer $l-1$ is the query, and $\boldsymbol{z}_{\leq t}^{l-1} = [\boldsymbol{z}_1^{l-1}, \ldots, \boldsymbol{z}_t^{l-1}]$, the decoder's hidden states from time 1 to time t at layer $l-1$, are the keys and values. To obtain better LM context, we employ residual connection and layer normalization after the multi-head self-attention. Similarly, the sememe context can be computed by attending to the encoder's outputs with multi-head attention:

$$\hat{\boldsymbol{c}}_t{}^l = \mathrm{MultiHead}(\boldsymbol{o}_t^l, \boldsymbol{h}, \boldsymbol{h}) \tag{22}$$

where $\boldsymbol{o}_t^l$ is the query, and the output from the encoder stack $\boldsymbol{h} = [\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_N]$, are the values and keys.

Once obtaining the sememe context vector $\hat{\boldsymbol{c}}_t{}^l$ and the LM context $\boldsymbol{o}_t^l$, we compute the output from the adaptive attention

TABLE I
STATISTICS OF THE DATA SETS. THE COLUMNS IN THE TABLE ARE THE NUMBER OF INTERPRETED WORDS, ENTRIES, TOKENS OF DEFINITIONS, AND SEMEMES. JIEBA CHINESE TEXT SEGMENTATION TOOL IS USED DURING SEGMENTATION

Dataset	# words	# entries	# tokens	# sememes
CCD				
Train	27,047	94,029	568,361	160,792
Valid	1,503	5,218	31,956	8,966
Test	1,502	5,270	31,543	8,851
CWN				
Train	6,575	21,736	198,314	40,848
Valid	823	2,606	23,684	4,743
Test	824	2,774	25,169	5,183

layer with:

$$\boldsymbol{c}_t^l = \beta_t^l \boldsymbol{o}_t^l + (1 - \beta_t^l)\hat{\boldsymbol{c}}_t{}^l, \tag{23}$$

where

$$\begin{aligned} \beta_t^l &= \frac{\exp(e_{to})}{\exp(e_{to}) + \exp(e_{t\hat{c}})} \\ e_{to}^l &= (\boldsymbol{w}_c^l)^T [\boldsymbol{o}_t^l; \boldsymbol{z}_t^{l-1}] \\ e_{t\hat{c}}^l &= (\boldsymbol{w}_c^l)^T [\hat{\boldsymbol{c}}_t{}^l; \boldsymbol{z}_t^{l-1}] \end{aligned} \tag{24}$$

III. EXPERIMENTS

In this section, we will first introduce the construction process of the CDM dataset, then the experimental results and analysis.

A. Dataset

To verify our proposed models, we construct two novel datasets for the Chinese definition modeling task, hich are built from Chinese Concept Dictionary (CCD) [4] and Chinese WordNet (CWN) [5] respectively. Each entry in datasets is a triple that consists of: the interpreted word, sememes and a definition for a specific word sense, where the sememes are annotated with HowNet [6].

Concretely, take the process of building a dataset from CCD as an example, for a common word in HowNet and CCD, we first align its definitions from CCD and sememe groups from HowNet, where each group represents one word sense. We define the sememes of a definition as the combined sememes associated with any token of the definition. Then for each definition of a word, we align it with the sememe group that has the largest number of overlapping sememes with the definition's sememes. With such aligned definition and sememe group, we add an entry that consists of the word, the sememes of the aligned sememe group and the aligned definition. Each word can have multiple entries in the dataset, especially the polysemous word. To improve the quality of the created dataset, we filter out entries that the definition contains the interpreted word, or the interpreted word is among function words, numeral words and proper nouns.

After processing, we obtain the dataset that contains 131,633 entries in total. We divide the CCD and CWN datasets according to the unique interpreted words into training set, validation set and test set. Table I shows the detailed data statistics. We also count the overlap proportion of sememes and words used in gold-standard definitions. The result shows that the coincidence rate is quite low. Therefore, sememes can provide additional semantic information, but it wouldn't reduce the difficulty of the definition generation task. The statistical results are shown in Table II.

TABLE II
OVERLAP PROPORTION OF SEMEMES AND WORDS USED IN GOLD-STANDARD DEFINITIONS

	Train	Valid	Test
CCD	1.89%	1.89%	1.94%
CWN	1.45%	1.36%	1.43%

B. Settings

We show the effectiveness of our proposed models on both datasets. All the embeddings, including word and sememe embeddings, are fixed 300 dimensional word embeddings pretrained on the Chinese Gigaword corpus (LDC2011T13). All definitions are segmented with Jiaba Chinese text segmentation tool[1] and we use the resulting unique segments as the decoder vocabulary. To measure the quality of the definitions generated by models, we calculate BLEU scores to evaluate the difference between the generated results and the gold-standard definitions. In this way, the closer the generated results is to the gold-standard definitions, the higher the score obtained. And we compute the BLEU score using a script provided by Moses, following Noraset *et al.* [1]. However, the BLEU score can only measure literal proximity and cannot measure semantic similarity, so the BLEU metric is not completely suitable for the task of definition generation. Therefore, we organized manual evaluations and conducted a qualitative analysis of the results generated by the model. In order to compare the differences between the models and the effect of incorporating sememes on the model, we conducted the following experiments on both CCD and CWN datasets.

c) Baseline w/ sememes: This is a variant of model proposed by Noraset *et al.* [1]. The original model only accepts the vector representations of the interpreted word as input, so we incorporate sememes into the model by adding the word vector with vectors of all its corresponding sememes altogether. The rest of the model remains the same as the original model.

d) Baseline w/o sememes: Following the same experimental setup with Noraset *et al.* [1], we use a two-layer LSTM network as the recurrent component. And we add a trainable embedding from character-level CNN to the fixed word embedding to achieve the best performance.

e) AAM w/ sememes: For comparision, we also use a two-layer LSTM network as the recurrent component. The model receives sememes as the input sequence. And each sememe vector is concatenated with the interpreted word vector at each time step. If a word is not included in HowNet, i.e. has no sememes, we use the word itself as a sememe for substitution.

[1][Online]. Available: https://github.com/fxsjy/jieba

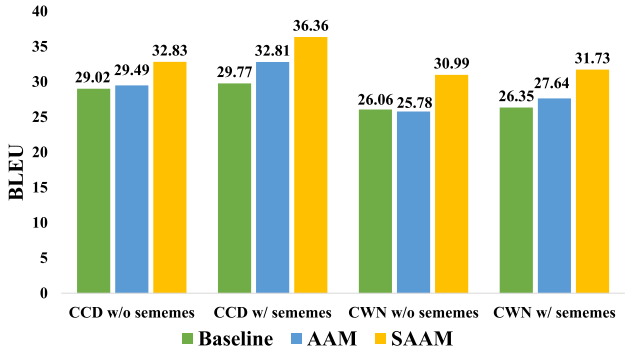

Fig. 3. Experimental results of the three models on CCD and CWN test sets.

f) AAM w/o sememes: In this scenario, we only use the vector of the interpreted words as input. To make the input format meet the model requirements, we concatenate the vector with a replicated vector of itself in the encoder portion.

g) SAAM w/ sememes: Since the self-attention mechanism is capable of modeling long-distance dependencies, we put the interpreted word as the first token of the input sequence, while the rest tokens of the input sequence are sememes. Similarly, if a word has no sememes, we use the word itself as a sememe.

h) SAAM w/o sememes: This experiment is similas to the previous case, except that we have deleted the sememes in the input sequence.

In all the experiments above, we choose the model with the highest BLEU score on the validation set. The selected model is employed to calculate the final BLEU score on the test set. For comparison purpose, the adam [17] optimizer is used in all experiments, and the beam size is set to 1 during the test phase. However, due to differences in model architectures and implementation, the number of training epochs, batch size, and learning rate are different in each experiment. More detailed parameter settings cab be found in our code.

C. Results and Analysis

a) Main Results: We report the experimental results on both CCD and CWN test sets in Figure 3. It shows that both of our proposed models, namely AAM and SAAM, achieve good results and outperform the baseline by a large margin. On the CCD dataset, AAM and SAAM incorporated sememes improve over the baseline that doesn't use sememes with +3.79 BLEU and +7.43 BLEU respectively. On the CWN dataset, the same comparison results in +1.58 BLEU and +5.67 BLEU respectively.

The results show that sememes are very useful in definition generation. Compared with the case of w/ and w/o sememes, the performance of SAAM improved +6.59 and +5.38 on CCD and CWN respectively. Considering that sememes can help distinguish the meaning of words and provide additional auxiliary information, we think this result is very reasonable.

In the baseline model, directly adding the sememe vectors to the word vector only slightly improves the performance. This indicates that how to encode sememes also has a very important impact on the results. The experimental results show that the way in which the sememes are encoded as sequences in AAM and SAAM can significantly improve the model performance.

Among all models, SAAM w/ sememes achieves the new state-of-the-art, with a BLEU score of 36.36 on the CCD test set and 31.73 on the CWN test set respectively, demonstrating the effectiveness of sememes and the architecture of SAAM.

b) Ablation Study: We also conduct an ablation study on the CCD dataset to evaluate the various choices we made for SAAM. We consider three key components: position embedding, the adaptive attention layer, and the incorporated sememes. As illustrated in table IV, we remove one of these components and report the performance of the resulting model on validation set and test set. We also list the performance of the baseline and AAM for reference.

It demonstrates that all components benefit the SAAM. Removing position embedding is 0.31 BLEU below the SAAM on the test set. Removing the adaptive attention layer is 0.43 BLEU below the SAAM on the test set. Sememes affects the most. Without incoporating sememes, the performance drops 3.53 BLEU on the test set.

c) Manual Evaluation: In order to further compare our proposed model with the baseline model and mitigate the shortcomings of the BLEU metric, we performed a manual evaluation on the CWN dataset. We randomly selected 200 samples from the test set of CWN and let four native Chinese speakers rate these definitions. Prior to distributing to the raters, we shuffled the order of all definitions and deleted there source information. So the raters wouldn't know which model each definition was generated from. Each rater scores the definitions from 1 to 5 points as: 1) completely wrong or self-definition, 2) correct topic with wrong infomation, 3) correct but incomplete, 4) small details missing, 5) correct. We made the scoring criteria following Ishiwatari *et al.* [18]. The averated scores are reported in table V. The manual evaluation results indicates that the generated definitions of SAAM is significantly better than that of the baseline model. In addition, SAAM can generate definitions with more concrete information by incorporating sememes.

d) Qualitative Analysis: Table III lists some example definitions generated with different models. For each word-sememes pair, the generated three definitions are ordered according to the order: Baseline, AAM and SAAM. For AAM and SAAM, we use the model that incorporates sememes. These examples show that with sememes, the model can generate more accurate and concrete definitions. For example, for the word "旅馆" (hotel), the baseline model fails to generate definition containing the token "旅行者"(tourists). However, by incoporating sememes, especially the sememe "旅游" (tour), AAM and SAAM successfully generate "旅行者"(tourists). Manual inspection of others examples also supports our claim.

However, SAAM w/ sememes also performed poorly in some cases. Table VI lists some of the failed examples. In example # 1, the model failed to capture the key information conveyed by the sememe "教育" (education). In example # 2 and # 3, the only sememe couldn't provide useful additional information to

TABLE III
EXAMPLE DEFINITIONS GENERATED BY OUR MODELS. BASELINE REPRESENTS NORASET *ET AL.* [1]. NOTE THAT BASELINE DO NOT UTILIZE SEMEMES, WHILE THE AAM AND SAAM MODELS BOTH USE SEMEMES

Word	Sememes	Model	Generated Definitions
气压计 (barometer)	用具(tool) 测量(measure) 力量(strength) 气(gas)	Baseline	测量轨道刻度盘的仪表 (Instrument for measuring track dial.)
		AAM	测量大气压力的装置 (An instrument that measures atmospheric pressure.)
		SAAM	测量大气压力的装置 (An instrument that measures atmospheric pressure.)
啼鸣 (birdsong)	喊(cry) 禽(bird)	Baseline	发出哀鸣的声音 (To make a whining sound.)
		AAM	发出大的声音 (To make a loud noise.)
		SAAM	鸟类发出的特有的声音 (The unique cry of birds.)
旅馆 (hotel)	场所(place) 旅游(tour) 吃(eat) 娱乐(recreation) 住下(reside)	Baseline	为人们提供食宿的地方 (A place to provide residence to people.)
		AAM	为旅行者提供食宿的地方 (A place to provide residence to tourists.)
		SAAM	给旅行者提供食宿和其他服务的地方 (A place to provide residence and other services to tourists.)

TABLE IV
ABLATION STUDY: BLEU SCORES ON THE CCD VALIDATION SET AND TEST SET. FOR THE LAST THREE ROWS, WE REMOVE POSITION EMBEDDING, THE ADAPTIVE ATTENTION LAYER OR SEMEMES INFORMATION FROM SAAM MODEL

Model	Valid	Test
SAAM	**37.11**	**36.36**
−position	36.79	36.05
−adaptive	35.69	35.93
−sememes	32.26	32.83

TABLE V
RESULTS OF MANUAL EVALUATION. HERE WE GIVE THE AVERAGE SCORE OF EACH RATER ON EACH MODEL. AND THE LAST COLUMN IS THE AVERAGE SCORE OF ALL RATERS ON EACH MODEL

	01	02	03	04	Mean
Gold	4.80	4.37	4.42	4.79	4.59
SAAM w/	3.13	2.79	2.89	3.42	3.05
SAAM w/o	2.85	2.33	2.48	3.10	2.69
Baseline w/o	2.03	1.65	1.79	2.34	1.95

TABLE VI
EXAMPLES THAT SAAM FAILED TO GENERATE DEFINITIONS

# 1	
Input Word:	同班 (in the same class)
Sememes:	归属值 (attachment value) 似 (alike) 教育 (education)
Generated Def:	形容有共同的对象的。 Describe with common objects.
# 2	
Input Word:	东区 (east district)
Sememes:	东区 (east district)
Generated Def:	空间范围内西边的区域。 The area west of the space.
# 3	
Input Word:	人造 (man-made)
Sememes:	人为 (artificial)
Generated Def:	形容会造成很强的生理或化学反应的。 Described to cause a strong physiological or chemical reaction.

the model, where the sememe in # 2 is the interpreted word itself and the sememe in # 3 is a synonym of the interpreted word.

IV. RELATED WORK

A. Definition Modeling

Distributed representations of words, or word embeddings [19] were widely used in the field of NLP in recent years. Since word embeddings have been shown to capture lexical semantics, Noraset *et al.* [1] proposed the definition modeling task as a more transparent and direct representation of word embeddings. This work is followed by Gadetsky *et al.* [2], who studied the problem of word ambiguities in definition modeling by employing latent variable modeling and soft attention mechanisms. Both works focus on evaluating and interpreting word embeddings. In contrast, we incorporate sememes to generate word sense aware word definition for dictionary compilation.

B. Knowledge Bases

Recently many knowledge bases (KBs) are established in order to organize human knowledge in structural forms. By providing human experiential knowledge, KBs are playing an increasingly important role as infrastructural facilities of natural language processing.

HowNet [20] is a knowledge base that annotates each concept in Chinese with one or more sememes. HowNet plays an important role in understanding the semantic meanings of concepts in human languages, and has been widely used in word representation learning [7], word similarity computation [21] and sentiment analysis [22]. For example, Niu *et al.* [7] improved word representation learning by utilizing sememes to represent various senses of each word and selecting suitable senses in contexts with an attention mechanism.

Chinese Concept Dictionary (CCD) is a WordNet-like semantic lexicon [3], [23], where each concept is defined by a set of synonyms (SynSet). CCD has been widely used in many NLP tasks, such as word sense disambiguation [23].

Another WordNet-like semantic lexicon in Chinese is the Chinese WordNet (CWN) [5]. The design criterion of CWN is to build a complete and robust knowledge system which also embodies a precise expression of semantic relations.

C. Self-Attention

Self-attention is a special case of attention mechanism that relates different positions of a single sequence in order to compute a representation for the sequence. Self-attention has been successfully applied to many tasks recently [10]–[12], [24]–[26].

Vaswani *et al.* [10] introduced the first transduction model based on self-attention by replacing the recurrent layers commonly used in encoder-decoder architectures with multi-head self-attention. The proposed model called Transformer achieved the state-of-the-art performance on neural machine translation with reduced training time. After that, Tan *et al.* [12] demonstrated that self-attention can improve semantic role labeling by handling structural information and long range dependencies. Kitaev and Klein [11] further extended self-attention to constituency parsing and showed that the use of self-attention helped to analyze the model by making explicit the manner in which information is propagated between different locations in the sentence.

Self-attention has many good properties. It reduces the computation complexity per layer, allows for more parallelization and reduces the path length between long-range dependencies in the network. In this paper, we use self-attention based architecture in SAAM to learn the relations of word, sememes and definition automatically.

V. Conclusion

We introduce the Chinese definition modeling task that generates a definition in Chinese for a given word and sememes of a specific word sense. This task is useful for dictionary compilation. To achieve this, we built two novel datasets with word-sememes-definition triples. We propose two novel methods, AAM and SAAM, to generate word sense aware definition by utilizing sememes. Experiments on the CCD and CWN datasets show that our proposed AAM and SAAM outperform the state-of-the-art approach with a large margin. By efficiently incorporating sememes, the SAAM achieves the best performance with significant improvement. We release the code of this work at https://github.com/blcuicall/AutoDict.

References

[1] T. Noraset, C. Liang, L. Birnbaum, and D. Downey, "Definition modeling: Learning to define word embeddings in natural language," in *Proc. Assoc. Adv. Artif. Intell.*, 2017, pp. 3259–3266.

[2] A. Gadetsky, I. Yakubovskiy, and D. Vetrov, "Conditional generators of words definitions," in *Proc. Assoc. Comput. Linguist.*, 2018, pp. 266–271.

[3] J. Yu, S. Yu, Y. Liu, and H. Zhang, "Introduction to Chinese concept dictionary," *J. Chin. Lang. Comput.*, vol. 11, no. 2, pp. 169–181, 2001.

[4] Y. Liu and S. Yu, *Multilingual Concept Dictionary*. Peking University Open Research Data Platform, 2017.

[5] C.-R. Huang *et al.*, "Chinese wordnet: Design, implementation, and application of an infrastructure for cross-lingual knowledge processing," *J. Chin. Inf. Process.*, vol. 24, no. 2, pp. 14–24, 2010.

[6] Z. Dong and Q. Dong, *HowNet and the computation of meaning*. World Scientific, 2006.

[7] Y. Niu, R. Xie, Z. Liu, and M. Sun, "Improved word representation learning with sememes," in *Proc. Assoc. Comput. Linguist.*, 2017, pp. 2049–2058.

[8] Q. Liu and S. Li, "Word similarity computing based on how-net," *Int. J. Comput. Linguist. Chin. Lang. Process.*, 2002, pp. 59–76.

[9] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 375–383.

[10] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[11] N. Kitaev and D. Klein, "Constituency parsing with a self-attentive encoder," in *Proc. Assoc. Comput. Linguist.*, 2018, pp. 2676–2686.

[12] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, "Deep semantic role labeling with self-attention," in *Proc. Assoc. Adv. Artif. Intell.*, 2018, pp. 4929–4936.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[14] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[15] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. Assoc. Comput. Linguist.*, 2017, pp. 1073–1083.

[16] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450*, 2016.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–13.

[18] S. Ishiwatari, H. Hayashi, N. Yoshinaga, G. Neubig, S. Sato, M. Toyoda, and M. Kitsuregawa, "Learning to describe unknown phrases with local and global contexts," in *Proc. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 3467–3476.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[20] Z. Dong and Q. Dong, "Hownet - a hybrid language and knowledge resource," in *Proc. Int. Conf. Natural Lang. Process. Knowl. Eng.*, 2003, pp. 820–824.

[21] Q. Liu, "Word similarity computing based on hownet," *Comput. Linguist. Chin. Lang. Process.*, vol. 7, no. 2, pp. 59–76, 2002.

[22] X. Fu, G. Liu, Y. Guo, and Z. Wang, "Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon," *Know.-Based Syst.*, vol. 37, pp. 186–195, 2013.

[23] J. Yu, "Wsd and closed semantic constraint," in *Proc. SIGHAN@COLING*, 2002, pp. 1–5.

[24] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. Empirical Methods Natural Lang. Process.*, 2016, pp. 551–561.

[25] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in Empirical Methods Natural Lang. Process., 2016, pp. 2249–2255.

[26] Z. Lin *et al.*, "A Structured Self-attentive Sentence Embedding," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–15.

Liner Yang received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2018. He is currently a Lecturer at the School of Information Sciences, Beijing Language and Culture University, Beijing. His research interests include natural language processing and intelligent computer-assisted language learning.

Cunliang Kong is currently a Ph.D. Student at the College of Information Science, Beijing Language and Culture University, Beijing, China. He is supervised by Prof. Erhong Yang and Dr. Liner Yang. His research interests focus on natural language processing and text generation.

Yun Chen received the B.S. degree in microelectronics from Tsinghua University, Beijing, China, in 2013 and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong, Hong Kong, China, in 2018. Currently, she is an Assistant Professor at School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China. From 2018 to 2019, she was a Researcher at Noah's Ark Lab, Huawei, Hong Kong, China. She was a Visiting Scholar at Tsinghua University from 2016 to 2017 and New York University from 2017 to 2018. She is broadly interested in machine learning and natural language processing, especially neural machine translation and pre-trained language models.

Yang Liu was born in 1979. He received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2007. He is currently a Professor at the Department of Computer Science and Technology, Tsinghua University, Beijing. His research interests focus on natural language processing.

Qinan Fan was born in 1995. She is now a Graduate Student at the College of Information Science of the Beijing Language and Culture University. She is supervised by Prof. Erhong Yang and Dr. Liner Yang. Her major research interests are natural language processing and text generation.

Erhong Yang received the M.S degree in computer science from Shanxi University, Taiyuan, Shanxi, China, in 1989, and the Ph.D. degree in linguistics from the Beijing language and Culture University, Beijing, China, in 2005. She is Executive Deputy Director of Beijing Advanced Innovation Center for Language Resource, Beijing Language and Culture University. Her research interests include language resources, computational linguistics. She developed a large-scale Chinese segmentation and annotation corpus shared on Chinese LDC, and she managed the development of the Print Media Monitoring Corpus, based on which she conducted annual language usage survey, carried out real-time language resources monitoring according to big media data, and regularly released media wording and phrasing as well as media buzzwords to the society each year.