Evaluating Semantic Rationality of a Sentence: A Sememe-Word-Matching Neural Network based on HowNet

Shu Liu[1], Jingjing Xu[2], and Xuancheng Ren[2]

[1] Center for Data Science, Beijing Institute of Big Data Research, Peking University
[2] MOE Key Lab of Computational Linguistics, School of EECS, Peking University
{shuliu123,jingjingxu,renxc}@pku.edu.cn

Abstract. Automatic evaluation of semantic rationality is an important yet challenging task, and current automatic techniques cannot effectively identify whether a sentence is semantically rational. Methods based on the language model do not measure the sentence by rationality but by commonness. Methods based on the similarity with human written sentences will fail if human-written references are not available. In this paper, we propose a novel model called Sememe-Word-Matching Neural Network (SWM-NN) to tackle semantic rationality evaluation by taking advantage of the sememe knowledge base HowNet. The advantage is that our model can utilize a proper combination of sememes to represent the fine-grained semantic meanings of a word within specific contexts. We use the fine-grained semantic representation to help the model learn the semantic dependency among words. To evaluate the effectiveness of the proposed model, we build a large-scale rationality evaluation dataset. Experimental results on this dataset show that the proposed model outperforms the competitive baselines.

Keywords: Semantic Rationality · Sememe-Word Matching Nerual Network · HowNet.

1 Introduction

Recently, tasks involving natural language generation have been attracting heated attention. However, it remains a problem of how to measure the quality of the generated sentences most reasonably and efficiently. Such sentence as Chomsky's famous words, "colorless green ideas sleep furiously" [4], is correct in syntax but irrational in semantics. Conventional methods involve human judgments of different quality metrics. However, it is both labor-intensive and time-consuming. In this paper, we explore an important but challenging problem: how to automatically identify whether a sentence is semantically rational. Based on this problem, we propose an important task: Sentence Semantic Rationality Detection (SSRD), which aims to identify whether the sentence is rational in semantics. The task can benefit many natural language processing applications that require the evaluation of rationality and can also provide insights to resolve the irrationality in the generated sentences.

There are some automatic methods to evaluate the quality of a sentence. However, methods based on the language model [12, 3] do not measure the sentence by rationality but by commonness (i.e. the probability of a sentence in the space of all possible sentences). Considering that the uncommon sentences are not always irrational, this approach is not a suitable solution. Similarity-based methods such as BLEU [18], ROUGE [15], SARI [23] will fail if human-written references are not available. For some statistical feature-based methods such as decision tree [7], they only use statistical information of the sentence. However, it is also essential to use semantic information in evaluation.

The main difficulty in the evaluation of semantic rationality is that it requires systems with high ability to understand selectional restrictions. In linguistics, *selection* denotes the ability of predicates to determine the semantic content of their arguments. Predicates select their arguments, which means that they limit the semantic content of their arguments. The following example illustrates the concept of selection. For a sentence "The building is wilting", the argument "the building" violates the selectional restrictions of the predicate, "is wilting". To address this problem, we propose to take advantage of the sememe knowledge which gives a more detailed semantic information of the word. Using this sort of knowledge, a model would learn the selectional restrictions between words better.

Words can be represented with semantic sub-units from a finite set of limited size. For example, the word "lover" can be approximately represented as "{Human | Friend | Love | Desired}". Linguists define *sememes* as semantic sub-units of human languages [2] that express semantic meanings of concepts. One of the most well-known sememe knowledge bases is HowNet [5]. HowNet has been widely used in various Chinese NLP tasks, such as word sense disambiguation [6], named entity recognition [14] and word representation [17]. Zeng et al. [24] propose to expand the Linguistic Inquiry and Word Count [20] lexicons based on word sememes. There are also some works on sememe prediction. Xie et al. [22] predict lexical sememe via word embeddings and matrix factorization. Li et al. [13] conduct sememe prediction to learn semantic knowledge from unstructured textual Wiki descriptions. Jin et al. [9] incorporate characters of words in lexical sememe prediction.

In this work, we address the task of automatic semantic rationality evaluation by using the semantic information expressed by sememes. We design a novel model by combining word-level information with sememe-level semantic information to determine whether the sememes of the words are compatible so that the sentence does not violate common perception. We divide our model into two parts: a word-level part and a sememe-level part. First, the word-level part gets the context for each word. Next, we use the context of each word to select its proper sememe-level information. Finally, we detect whether a word violates the selectional restrictions of context words by word-level and sememe-level, respectively. Our main contributions are listed as follows:

- We propose the task of automatically detecting sentence semantic rationality and we build a new and large-scale dataset for this task.
- We propose a novel model called SWM-NN that combines sentence information with its sememe information given by the Chinese knowledge base HowNet. Experimental results show that the proposed method outperforms the baselines.

2 Proposed Method

To detect the semantic rationality of the sentence, we should represent the sentence into fine-grained semantic units. We deal with the task of SSRD with the aid of the sentence representation and its semantic representation.

Based on this motivation, we propose an SWM-NN model (see Figure 1). This model can make use of HowNet, which is a well-known Chinese semantic knowledge base. The overall architecture of SWM-NN consists of several parts: a word-level attention LSTM, a matching mechanism between the word-level and the sememe-level part, and a sememe-level attention LSTM. We first introduce the structures of HowNet, and then we describe the details of different components in the following sections.

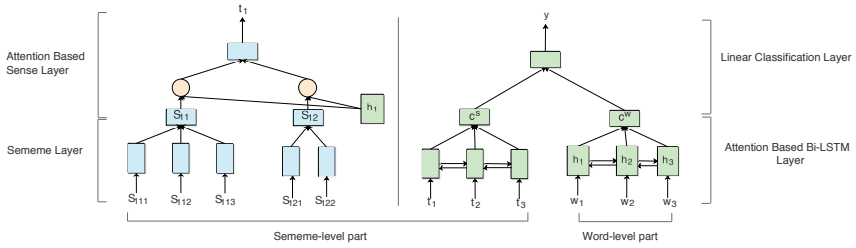

Fig. 1. The overview of SWM-NN model. The sentence first goes through the word-level Bi-LSTM with self-attenion to get context information. To query the sense and the sememe information, each sense is first represented as the average of sememes. The senses of a word are dynamically combined based on the correponding word-level context, forming a compositional semantic word representation, which then passes through the sememe-level Bi-LSTM to get context from another view. In this figure, the word w_1 has two senses $\boldsymbol{s}_{11}$ and $\boldsymbol{s}_{12}$. The sense $\boldsymbol{s}_{11}$ has three sememes $\boldsymbol{s}_{111}$, $\boldsymbol{s}_{112}$, $\boldsymbol{s}_{113}$. The sense $\boldsymbol{s}_{12}$ has two sememes $\boldsymbol{s}_{121}$, $\boldsymbol{s}_{122}$.

2.1 Sememes, Senses and Words in HowNet

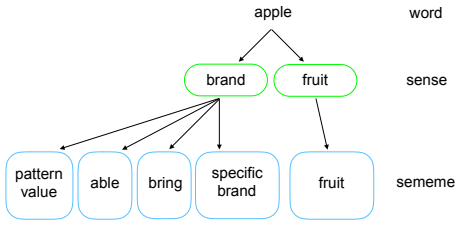

Fig. 2. Examples of sememes, senses, and words. We translate them into English.

HowNet annotates precise **senses** to each word, and for each **sense**, HowNet annotates the significant of parts and attributes represented by **sememes**. Figure 2 shows the sememe annotations of the word “apple”. The word “apple” actually has two main

senses: one is a sort of juicy fruit "fruit", and the other is a famous computer brand "brand". The latter **sense** "Apple brand" indicates a computer brand, and thus has **sememes** "computer", "bring", "Special Brand".

We introduce the notations used in the following sections as follows. Given a sentence $\boldsymbol{s}$ consisting of a sequence of words $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n\}$, we embed the one-hot representation of the i-th word $\boldsymbol{d}_i$ to a dense vector $\boldsymbol{w}_i$ through a word embedding matrix. For the i-th word $\boldsymbol{d}_i$, there can be multiple senses $\boldsymbol{s}_j^{(\boldsymbol{d}_i)}$ in HowNet. Each sense $\boldsymbol{s}_j^{(\boldsymbol{d}_i)}$ consists of several sememe words $\overline{\boldsymbol{d}}_k^{(s_j)}$ in HowNet. The one-hot representation of the sememe word $\overline{\boldsymbol{d}}$ is embedded to a dense vector $\boldsymbol{x}$ through a sememe embedding matrix.

2.2 Word-Level Attention LSTM

To detect the rationality using sentence information, we use a Bi-LSTM encoder with local attention in the word-level part. We first compute the context output $\boldsymbol{o}^w$ from the source sentence $\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_L\}$:

$$\overrightarrow{\boldsymbol{o}}_i^w, \overrightarrow{\boldsymbol{h}}_i^w = \mathrm{LSTM}_{word}(\boldsymbol{w}_i, \overrightarrow{\boldsymbol{h}}_{i-1}^w) \tag{1}$$

$$\overleftarrow{\boldsymbol{o}}_i^w, \overleftarrow{\boldsymbol{h}}_i^w = \mathrm{LSTM}_{word}(\boldsymbol{w}_i, \overleftarrow{\boldsymbol{h}}_{i+1}^w) \tag{2}$$

$$\boldsymbol{h}_i^w = [\overrightarrow{\boldsymbol{h}}_i^w; \overleftarrow{\boldsymbol{h}}_i^w] \tag{3}$$

$$\boldsymbol{o}_i^w = [\overrightarrow{\boldsymbol{o}}_i^w; \overleftarrow{\boldsymbol{o}}_i^w] \tag{4}$$

where L is the number of words in the source sentence. Then, we use the context output $\boldsymbol{o}^w = \{\boldsymbol{o}_1^w, \boldsymbol{o}_2^w, \cdots, \boldsymbol{o}_L^w\}$ to compute an attention vector $\boldsymbol{\alpha}^w = \{\alpha_1^w, \alpha_2^w, \cdots, \alpha_L^w\}$. Finally, we use the context output $\boldsymbol{o}^w$ and the attention vector $\boldsymbol{\alpha}^w$ to compute a word-level representation of the sentence $\boldsymbol{c}^w$. The calculation formulas are as follows:

$$\boldsymbol{u}_i^w = \tanh(\boldsymbol{W}_w \boldsymbol{o}_i^w + \boldsymbol{b}_w) \tag{5}$$

$$\alpha_i^w = \frac{\exp\left((\boldsymbol{u}_i^w)^T \boldsymbol{u}_w\right)}{\sum_j \exp\left((\boldsymbol{u}_j^w)^T \boldsymbol{u}_w\right)} \tag{6}$$

$$\boldsymbol{c}^w = \sum_i \alpha_i^w \boldsymbol{o}_i^w \tag{7}$$

where $\boldsymbol{W}_w$ and $\boldsymbol{b}_w$ are weight matrix and bias vector, respectively. $\boldsymbol{u}_w$ is a randomly initialized vector, which can be learned at the training stage. The attention mechanism is proposed in [1], which gives higher weights to certain features that allow better prediction. Through training, the certain feature is likely to be the word that destructs the rationality of the sentence in semantics.

2.3 Matching Mechanism Layer

In sememe-level part, we average the sememe embeddings to represent each sense of the word $\boldsymbol{d}$ at first:

$$\boldsymbol{s}_j^{(\boldsymbol{d})} = \frac{1}{m_j^{(\boldsymbol{d})}} \sum_k \boldsymbol{x}_k^{(\boldsymbol{s}_j)} \tag{8}$$

where $\boldsymbol{s}_j^{(\boldsymbol{d})}$ stands for the j-th sense embedding of the word $\boldsymbol{d}$. $m_j^{(\boldsymbol{d})}$, $\boldsymbol{x}_k^{(\boldsymbol{s}_j)}$ stands for the number of sememes and the k-th sememe embedding belonging to the j-th sense of $\boldsymbol{d}$ (i.e. $\boldsymbol{s}_j^{(\boldsymbol{d})}$), respectively. Hence, given a word $\boldsymbol{d}_i$, we can get the sense embedding matrix of $\boldsymbol{d}_i$, referred to as $\boldsymbol{S}^{(\boldsymbol{d_i})} = [\boldsymbol{s}_1^{(\boldsymbol{d_i})}, \boldsymbol{s}_2^{(\boldsymbol{d_i})}, \cdots, \boldsymbol{s}_{n_i}^{(\boldsymbol{d_i})}]$, where n_i stands for the number of senses belong to $\boldsymbol{d_i}$.

To match the appropriate senses and sememes to each word given a specific sentence, we add a matching mechanism that is based on global attention. Since the output of word-level LSTM $\boldsymbol{o}_i^w$ can be viewed as the contextual representation. For each word $\boldsymbol{d}_i$, we have the output state $\boldsymbol{o}_i^w$ in word-level LSTM and its sense embedding matrix $\boldsymbol{S}^{(\boldsymbol{d_i})} = [\boldsymbol{s}_1^{(\boldsymbol{d_i})}, \boldsymbol{s}_2^{(\boldsymbol{d_i})}, \cdots, \boldsymbol{s}_n^{(\boldsymbol{d_i})}]$.

We compute the sememe-level representation $\boldsymbol{t}_i$ of the word $\boldsymbol{d}_i$ as follows:

$$\beta_j = \frac{\exp\left(g(\boldsymbol{o}_i^w, \boldsymbol{s}_j^{(\boldsymbol{d}_i)})\right)}{\sum_k \exp\left(g(\boldsymbol{o}_i^w, \boldsymbol{s}_k^{(\boldsymbol{d}_i)})\right)} \tag{9}$$

$$\boldsymbol{t}_i = \sum_j \beta_j \boldsymbol{s}_j^{(\boldsymbol{d}_i)} \tag{10}$$

Here the score function g is computed as follows:

$$g(\boldsymbol{o}_i^w, \boldsymbol{s}_j^{(\boldsymbol{d}_i)}) = \tanh(\boldsymbol{W}_x \boldsymbol{o}_i^w) \odot \tanh(\boldsymbol{W}_y \boldsymbol{s}_j^{(\boldsymbol{d}_i)}) \tag{11}$$

where $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$ are model parameters, which can be learned at the training stage. Through matching mechanism layer, the fine-grained semantic dependency between words in a sentence can be modeled by the combination of sememes.

2.4 Sememe-level Attention LSTM

For each sentence $s = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n\}$, we can get its sememe-level sequences $\{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_n\}$ based on the computation mentioned above. We use a sememe-level attention LSTM, which is similar to the word-level attention LSTM, to get the sememe-level representation of the sentence $\boldsymbol{c}^s$.

2.5 Combining Information from the two parts

In order to avoid semantic rationality signals being dominated by sentence-level or sememe-level [25], we add gates controlled by the representations of two parts. We combine information from two parts as follows:

$$\boldsymbol{z}^w \propto \exp(\boldsymbol{W}^w \boldsymbol{c}^w + \boldsymbol{b}^w) \tag{12}$$

$$\boldsymbol{z}^s \propto \exp(\boldsymbol{W}^s \boldsymbol{c}^s + \boldsymbol{b}^s) \tag{13}$$

$$\boldsymbol{c} = \boldsymbol{c}^w \odot \boldsymbol{z}^w + \boldsymbol{c}^s \odot \boldsymbol{z}^s \tag{14}$$

where $\boldsymbol{z}^w + \boldsymbol{z}^s = \overrightarrow{1}$. Then the probability distribution of label is predicted by ($f(\cdot)$ refers to a non-linear function ReLU [16])

$$p = softmax(f(\boldsymbol{W}\boldsymbol{c} + \boldsymbol{b})) \tag{15}$$

$\boldsymbol{\theta}$ is the model parameter and y is the ground-truth label of the sentence, then the cross entropy loss is

$$\mathcal{L}(\boldsymbol{\theta}) = -y \log p(y|\boldsymbol{w}, \boldsymbol{s}, \boldsymbol{\theta}) \tag{16}$$

3 Experiments

In this section, we evaluate our model on the dataset we build for the SSRD task. Firstly, we introduce the dataset and the experimental details. Then, we compare our model with baselines. Finally, we provide the analysis and the discussion of the experimental results.

3.1 Dataset

We create our dataset by collecting Chinese Word Segmentation and Part-of-Speech Tagging corpus from China National Language Committee[3]. Then we divide this dataset into training, validation, and test set. To create sentences lacking semantic rationality (i.e. the negative sentences), we randomly do one of the following operations on every sentence in each set:

1. Replace word with the same POS word in vocabulary randomly. The vocabulary was created from our corpus.
2. Reverse the position of two words of the same POS randomly.

The negative sentences in [20] are generated by n-gram language model. However, this method may induce syntax errors rather than semantic errors. The two operations in this paper will only cause semantic errors. For example, Chomsky's famous semantically irrational sentence, "colorless green ideas sleep furiously", can be created by our method. In addition, our method includes some operations like exploiting polysemy-replacement, swapping semantic roles, etc.

In order to ensure the irrationality of these negative sentences, we operate sentences whose lengths are more than 8 and we do not replace or reverse the punctuation of the sentence. In the meantime, we ask the human annotators to check the irrationality of these negative sentences in our test set.

The details of each set are shown in Table 1 .

Dataset	**#Total**	**#Positive**	**#Negative**
Training set	160,000	80,000	80,000
Validation set	20,000	10,000	10,000
Test set	20,000	10,000	10,000

Table 1. Statistical information of the final dataset. **Positive** and **Negative** denote whether the sentence is semantic rational.

[3] `http://www.aihanyu.org/cncorpus/`

3.2 Experimental Details

We use accuracy as our evaluation metric instead of the F-score, precision, and recall because the positive and negative examples in our dataset are balanced. As the words and the sememes are different in meaning, we do not share their vocabulary. We build up vocabularies for words and sememes with the size of $50{,}000$ and $20{,}000$, respectively. Some words are not annotated and thus have no sememes in HowNet. We simply use the word itself as the sememe.

We use the same dimension of 128 for word embeddings and sememe embeddings, and they are randomly initialized and can be learned during training. Adam optimizer [11] is used to minimize cross entropy loss function. We apply dropout regularization [21] to avoid overfitting and clip the gradients [19] to the maximum norm of 3.0. During training, we train the model for 20 epochs and monitor its performance on the validation set after every 200 updates. Once training is finished, we select the model with the highest accuracy on the validation set as our final model and evaluate its performance on the test set.

3.3 Baseline Models

- **N-gram language model:** We use the best performing N-gram smoothing methods, the interpolated Kneser-Ney algorithm [12, 3]. The positive sentences in the training set are used to train the model. For detecting rationality, we calculate a threshold based on the validation set that maximizes the accuracy. Then, we predict the test set using the model and the threshold.
- **Traditional machine learning algorithms:** We use various machine learning classifiers to predict the labels based on the tf-idf features of the sentence. We compute the probability distribution of the label by inputting the sentence word sequence and its sememe word sequence to the model respectively. Then we ensemble the probability of both sequences to get the label prediction.
- **Neural networks models:** We apply two representative neural network models: Bi-LSTM [8] and CNN [10]. The neural network is used for learning the vector representation for the word sequence and the sememe sequence, respectively. Then both outputs are concatenated and serve as input to a linear classifier.
- **Human evaluation:** For 500 randomly chosen sentences, we provide human annotators with the true sentence and the permuted sentence. Then we ask them to select a better sentence. The result can be viewed as an upper bound for this task.

3.4 Results

In this subsection, we present the results of evaluation by comparing our proposed method with the baselines. Table 2 reports experimental results of various models. From the results, we can observe that:

- The proposed SWM-NN outperforms all the baselines except the human evaluation. Our model uses dual-attention mechanism that consists of local attention in both levels, and a global attention to match the word to its appropriate combination of the sememes. By properly incorporating knowledge in HowNet and information of the source sentence, our model is capable of making more accurate predictions.

Models	Accuracy
Interpolated Kneser-Ney	53.2%
Random Forest	60.5%
Linear SVM	58.7%
SVM	57.1%
Naive Bayes	54.6%
CNN	62.7%
Bi-LSTM	63.5%
SWM-NN	**69.1%**
Human Evaluation	**94.4%**

Table 2. Comparison between our proposed model and the baselines on the test set. Our proposed model is denoted as **SWM-NN**.

- We see that the interpolated Kneser-Ney language model get the lowest prediction accuracy in the baseline model. It partly verifies our arguments on resolving the task using language models.
- The traditional machine learning algorithms with sememe information only achieve the accuracy of 60.5% at best. Neural network based methods perform much better and beat other baselines. This shows that the generalization ability of neural networks is better (the positive sentences and their similar negative sentences only coexist in the same data set). However, the neural network with the sememe information given by HowNet only achieves the accuracy of 63.5% at best. It suggests merely providing the sememes to the models is not sufficient for detecting rationality. Further matching of sememes to check the compatibility of the sememes is crucial to the overall performance.

4 Analysis and Discussions

Here, we perform further analysis on the model, including the ablation study, error analysis, and some further experimental results.

4.1 Exploration on Internal Structure of the Model

As shown in Table 2, our SWM-NN model outperforms all the baselines. Compared with the baseline neural network model, the proposed model has a dual-attention mechanism, that is, (1) a local attention mechanism in both the word-level and the sememe-level and (2) a global attention mechanism to match information between two levels. In order to explore the impact of the internal structure of the model, we remove the components of our model in order. The performance is shown in Table 3.

- **w/o Match** means that we do not match the context of the word to its sememe information by the global attention mechanism. For each word $\boldsymbol{d}$, we only average all the sememe embedding to get $\boldsymbol{t}$ as follows:

$$\boldsymbol{t} = \sum_{j} \frac{1}{n} \boldsymbol{s}_j^{(d)} \tag{17}$$

Models	Accuracy	Decline
SWM-NN	68.7%	−−
w/o Match	67.6%	↓ 1.1%
w/o Dual-attention	63.1%	↓ 5.6%
w/o HowNet	67.0%	↓ 1.7%
w/o Word-level $\boldsymbol{c}^w$	67.9%	↓ 0.8%

Table 3. Ablation Study on the validation set.

n is the number of senses of this word.

- **w/o Dual-attention** means that we do not use dual attention mechanism (i.e. local attention in both levels and global attention between two levels) in the proposed model any more, which is the same as the Bi-LSTM in the baseline models.
- **w/o HowNet** means we do not use the knowledge given by HowNet. It is equivalent to our model without sememe-level local attention and matching mechanism.
- **w/o Word-level $\boldsymbol{c}^w$** means without word-level representation, that is, we only use $\boldsymbol{c}^s$ to predict label. But we still use other structures of SWM-NN.

From the results shown in Table 3, we can observe that:

- Without the knowledge in HowNet, the accuracy of the model drops by 1.7% (in **w/o HowNet**). The sememe knowledge given by HowNet can provide some fine-grained semantic information, and thus can help the task of SSRD.
- It is useful to model the relation between the sentence and HowNet knowledge more properly. We can observe that without the matching mechanism between the sememe-level and the word-level, the accuracy of the model drops by 1.1% (in **w/o Match**). It shows matching mechanism can give a more rational and fine-grained semantic representation of the sentence. Furthermore, this sort of representation can help the task of SSRD.
- The Dual-attention mechanism is of great help to our task. Without this mechanism, the accuracy of the model drops by 5.6% (in **w/o Dual-attention**). It shows this sort of hierarchical attention mechanism in SWM-NN can make use of the information of sentence and HowNet properly to achieve our task.
- Without the word-level representation of the sentence, the accuracy of the model drops by 0.8% (in **w/o Word-level $\boldsymbol{c}^w$**). It is a loss that cannot be ignored. Even if we get a proper sememe representation, the representation of sentence in word-level is also helpful in our task.

Based on the ablation studies above, every part of our model is necessary to achieve the best result in the task of SSRD.

4.2 Case Study

Here we show a sentence and its dual-attention weight visualization in the test set for case study. Table 4 shows an example that gets a correct prediction in our test set. This sentence is a negative sentence in the test set and the bold word is the word we replaced. We can see that the "Word-level attention" gives higher weights to the word "全总". It

Test Sentence	全总 等 单位 慰问 本 **教师** **市**
Word-level attention	全总 等 单位 慰问 本 教师 市
Matching attention	全总：全总 等：实体、属性、类型|功能词|相等|实体、等级|等待 单位：单位、量度|事务、从事、组织 慰问：安慰、问候 本：书|读物|己|事件、实体、根、部件|实体、根、部件|簿册|植物、身、部件|资金、金融|现在|特定 教师：人、教、教育、职位 市：地方、市|专、地方、市
Sememe-level attention	全总 等 单位 慰问 本 教师 市

Table 4. Some cases in the test set. Test Sentence 1 is a negative sentence. It is created by reversing the position of two words of the same POS randomly. The bold words are the words we replaced. Test Sentence 2 is a positive sentence. Word-level attention, Matching attention, and Sememe-level attention show the dual-attention mechanism visualization during prediction. In "Matching attention", the symbol "|" separates different senses of the word.

might because the word "全总" is the abbreviation for the word "全市总工会 (National Federation of Trade Unions)" in Chinese so that it confuses the word-level model. But this sort of situation is not conducive to the prediction. In the "Matching attention", we can see that the global attention mechanism weights are mainly correct except the word "等 (sort)". After the matching mechanism, we can observe that "Sememe-level attention" gives higher weights to the wrong word "教师 (teacher)" and "市 (city)". This shows that in order to predict correctly, our model gives a higher attention to the wrong words.

4.3 Error Analysis

For error analysis, we first construct four datasets. The permuted sentences in each set are created as follows.

- **Dataset1:** Replace one word with the same POS randomly.
- **Dataset2:** Replace two words with the same POS randomly.
- **Dataset3:** Reverse the position of two words of the same POS randomly.
- **Dataset4:** Reverse the position of two words randomly.

We train our models on each training set and then evaluate on the corresponding test set. Meanwhile, we select 500 sentences from each set and ask the human annotators to annotate. Table 5 shows the results of each dataset.

From the results shown in Table 5, we can see that

- For the model, the most difficult dataset is the dataset1 where the permuted sentences differ in only one word from the true sentences. This shows that the number of words replaced is the biggest challenge for the model. It is partly because that replacing one word with the same POS randomly will exploit polysemy as most of the replaced words have more than one sememes in HowNet. Furthermore, the

Dataset	Model	Human
Dataset1	35.5%	5.0%
Dataset2	29.9%	2.2%
Dataset3	32.4%	8.6%
Dataset4	28.3%	1.4%

Table 5. Error rate of the model evaluation and the human evaluation for each set.

model is less effective in predicting dataset3, even though the other datasets are replaced by two words. It is partly because that reversing the position of two words of the same POS will swap semantic roles.

- For the human, the most difficult dataset is dataset3. This can also partly show that dataset3 is the most difficult dataset for judging semantic-rationality. As for the other three datasets, both the number of replacement word and the POS of replacement word affects human judgment.
- The result of human prediction is much better than that predicted by the model. Among all the datasets, however, the performance of the model on dataset3 is not as bad as the performance of humans on dataset3.

5 Conclusion

In this paper, we propose the task of sentence semantic rationality detection (SSRD), which aims to identify whether the sentence is rational in semantics. To deal with the difficulties in this task and overcome the disadvantages of current methods, we propose a Sememe-Word-Matching Neural Network model that not only considers the information of the sentences, but also makes use of the sememe information in knowledge base HowNet. Furthermore, our model selects the proper sememe information by the matching mechanism. Experimental results show that our model can outperform various baselines by a large margin.

Further experiments show that although our model has achieved promising results, there is still a big gap compared with the artificial results. How to make better use of other knowledge bases in this sort of task will be our future work.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bloomfield, L.: A set of postulates for the science of language. Language **2**(3), 153–164 (1926)
3. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. Computer Speech & Language **13**(4), 359–394 (1999)
4. Chomsky, N.: Three models for the description of language. IRE Transactions on information theory **2**(3), 113–124 (1956)
5. Dong, Z., Dong, Q.: Hownet And The Computation Of Meaning (With Cd-rom). World Scientific (2006)
6. Duan, X., Zhao, J., Xu, B.: Word sense disambiguation through sememe labeling. In: IJCAI. pp. 1594–1599 (2007)

7. Eneva, E., Hoberman, R., Lita, L.: Learning within-sentence semantic coherence. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (2001)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
9. Jin, H., Zhu, H., Liu, Z., Xie, R., Sun, M., Lin, F., Lin, L.: Incorporating chinese characters of words for lexical sememe prediction. arXiv preprint arXiv:1806.06349 (2018)
10. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1746–1751 (2014)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
12. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on. vol. 1, pp. 181–184. IEEE (1995)
13. Li, W., Ren, X., Dai, D., Wu, Y., Wang, H., Sun, X.: Sememe prediction: Learning semantic knowledge from unstructured textual wiki descriptions. arXiv preprint arXiv:1808.05437 (2018)
14. Li, W., Wu, Y., Lv, X.: Improving word vector with prior knowledge in semantic dictionary. In: Lin, C.Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) Natural Language Understanding and Intelligent Applications. pp. 461–469. Springer International Publishing, Cham (2016)
15. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out (2004)
16. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
17. Niu, Y., Xie, R., Liu, Z., Sun, M.: Improved word representation learning with sememes. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 2049–2058 (2017)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. pp. 311–318. Association for Computational Linguistics (2002)
19. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning. pp. 1310–1318 (2013)
20. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Linguistic inquiry and word count: Liwc 2001. Mahway: Lawrence Erlbaum Associates **71**(2001), 2001 (2001)
21. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
22. Xie, R., Yuan, X., Liu, Z., Sun, M.: Lexical sememe prediction via word embeddings and matrix factorization. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 4200–4206. AAAI Press (2017)
23. Xu, W., Napoles, C., Pavlick, E., Chen, Q., Callison-Burch, C.: Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics **4**, 401–415 (2016)
24. Zeng, X., Yang, C., Tu, C., Liu, Z., Sun, M.: Chinese liwc lexicon expansion via hierarchical classification of word embeddings with sememe attention (2018)
25. Zhang, M., Zhang, Y., Vo, D.T.: Gated neural networks for targeted sentiment analysis. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)