Sememe Enhanced Recurrent Neural Networks

Anonymous ACL submission

Abstract

Recurrent Neural Network (RNN) is a very prevailing neural network model which has been widely used in many sequence modeling tasks of NLP. Many works have been proposed to incorporate external knowledge into vanilla RNN. However, most previous work utilized either general knowledge outside the RNN structure or task-related knowledge for specific tasks. We argue that there exists general linguistic knowledge (e.g., sememes) which can be employed into the modeling of RNN cell, help better represent the semantics of sequence and improve the performance of sequence modeling tasks. In this paper, we present a general and effective framework which can integrate sememes of words, the minimum semantic units of human languages, into different variants of RNN. We evaluate our framework on several benchmark datasets including PTB and WikiText-2 for language modeling, SNLI for natural language inference. We also evaluate the transferring ability of our model on sentence representation tasks. Experimental results show evident enhancement of our framework comparing with the original RNNs.

1 Introduction

Recurrent neural network (RNN) (Rumelhart et al., 1988) and its variants, such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014), have been widely employed in various NLP tasks including language modeling (Mikolov et al., 2010), sentiment analysis (Nakov et al., 2016), natural language inference (Parikh et al., 2016), etc. To improve the sequence modeling ability of RNNs, most works focused on reforming the frameworks of RNNs (Schmidhuber, 1992; Graves et al., 2005; Bachman and Precup, 2015).

Meanwhile, some studies tried to incorporate external knowledge, especially the general semantic knowledge, i.e., WordNet (Miller, 1998) and ConceptNet (Speer and Havasi, 2013), into RNNs (Ahn et al., 2016; Yang and Mitchell, 2017; Parthasarathi and Pineau, 2018; Young et al., 2018). However, limited by the characteristics of the knowledge bases, existing works failed to inject these knowledge into the internal structure of RNNs, which could hinder the performance of knowledge incorporation. People also tried another direction by utilizing the knowledge which can be easily represented in numeric form. Although the knowledge can be absorbed into internal structures of RNNs, these models are task-related and only work on specific tasks (Wang et al., 2016; Xu et al., 2017; Ma et al., 2018).

In fact, there exists general linguistic knowledge which can be easily incorporated into internal structures of RNNs, i.e., *sememes*. Sememes are defined as the minimum semantic units of natural languages (Bloomfield, 1926). We know that words are the smallest language elements which can be used in isolation, but the meanings of words can be divided into sememes. For example, the meaning of “man” can be divided into the sum of meanings of “human”, “adult” and “male”. Linguists believe that the meanings of all the words can be represented with a limited set of sememes, which is homologous with the idea of semantic primitives (Wierzbicka, 1996).

Since sememes are implicit, people manually define a set of sememes and annotate words with them. By doing so, a sememe KB is built. HowNet is one of the most well-known sememe KBs, which contains over 100 thousand Chinese and English words annotated with about 2,000 predefined sememes. Since its publication, HowNet has been successfully applied to various NLP applications, e.g. word similarity computation (Liu and

Li, 2002), sentiment analysis (Fu et al., 2013), word representation learning (Niu et al., 2017) and lexicon expansion (Zeng et al., 2018). Gu et al. (2018) make an attempt to incorporate sememe knowledge into a LSTM-based model for language modeling, but in their work, sememe is only useful in the decoder step, while context don't make use of sememe information. To the best of our knowledge, no previous works leverage sememe knowledge in RNNs for modeling better text sequences.

In this paper, we carry out the first exploration of incorporating sememe knowledge into RNNs for enhancing the text sequence modeling ability. We present two kinds of methods to incorporate sememe knowledge into RNN models, namely intracellularly and extracelluarly sememe-incorporated RNNs. For intracellularly sememe-incorporated RNNs, we propose three different models to incorporate sememe knowledge into the internal structures of RNN cells. For extracelluarly sememe-incorporated RNNs, we design a corresponding model to influence the sentence representations of hidden states by sememes. In addition, the two kinds of methods can be combined together as the overall model. All of our proposed models have high adaptability, and we retrofit several typical RNNs including LSTM, GRU and their bidirectional variants for evaluation. We test our models on the benchmark datasets of three representative sequence modeling tasks including language modeling, sentence classification and sentence pair classification. Experimental results show that our sememe-incorporated RNNs achieve consistent performance improvement on all the tasks compared with corresponding unretrofitted RNNs.

2 Related Work

Recent years have witnessed rapid development of research on RNN models in NLP field. Most works tried to alter the frameworks of RNNs, such as integrating attention mechanism (Bachman and Precup, 2015), adding hierarchical structures (Schmidhuber, 1992) and introducing bidirectional modeling (Graves et al., 2005). Some work focused on incorporating different kinds of external knowledge into RNN models. General linguistic knowledge from famous KBs including WordNet and ConceptNet attracted considerable attention. However, they are relation-based knowledge and hard to be utilized in RNNs. Although some works attempted to incorporate this kind of knowledge into RNNs, they usually manipulated the knowledge on the hidden layers of RNNs and failed to inject it into the internal structures of RNNs (Ahn et al., 2016; Yang and Mitchell, 2017; Parthasarathi and Pineau, 2018; Young et al., 2018). Other work explored other knowledge like aspect type (Ma et al., 2018) and entity-attribute information (Xu et al., 2017), which can be absorbed into the cells of RNNs but are usually task-specific. To the best of our knowledge, there are no works incorporating general linguistic knowledge into the internal structures of RNNs to improve overall sequence modeling ability.

HowNet (Dong and Dong, 2003) is the most famous sememe KB, whose construction takes several linguistic experts more than two decades. After HowNet is published, its sememe knowledge has been employed in diverse NLP tasks including word similarity computation (Liu and Li, 2002), word representation learning (Niu et al., 2017), word sense disambiguation (Zhang et al., 2005; Duan et al., 2007), sentiment analysis (Dang and Zhang, 2010; Fu et al., 2013), lexicon expansion (Zeng et al., 2018), etc. There are also some works trying to expand HowNet (Xie et al., 2017; Jin et al., 2018) and transfer its sememe knowledge to other languages (Qi et al., 2018). Gu et al. (2018) exploited sememe knowledge in language modeling for the first time, and found sememe knowledge can improve the ability of LSTM to correctly predict next words.

3 Methodology

In this section, we detail our proposed methods of incorporating sememe knowledge into RNNs. We first introduce the sememe annotation in HowNet. Then we briefly introduce two popular RNNs, namely LSTM and GRU, together with their bidirectional variants. Next, we elaborately describe two kinds of ways to incorporate sememe knowledge, where we take unidirectional LSTM and GRU as examples but bidirectional LSTM and GRU can be retrofitted similarly. The first way is incorporating sememe knowledge into RNN cells, for which we propose three different implementation methods, and the second way is incorporating sememe knowledge based on RNN hidden states. Finally, we propose to combine the two sememe-incorporating ways together.

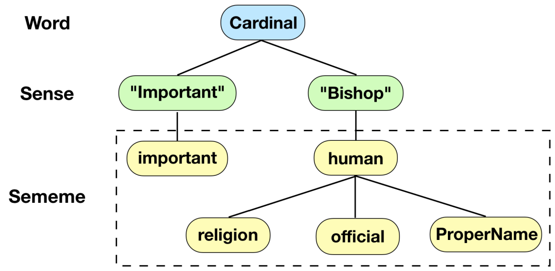

Figure 1: An example of how words are annotated with sememes in HowNet.

3.1 Sememe Hierarchy in HowNet

We introduce how words are connected with sememes in HowNet. Each word in HowNet may have one or several senses and each sense is annotated with several sememes with hierarchical structures. As shown in Figure 1, the word “cardinal” has two senses, namely “cardinal(important)” and “cardinal(bishop)”. The former one has only one sememe `important`, while the latter one has one main sememe `human` and three subsidiary sememes including `religion`, `official` and `ProperName`.

In this paper, we focus on the meanings of sememes and ignore their structures for simplicity. Therefore, we simply equip each word with a sememe set which consists of all the sememes of the word’s senses.

3.2 An Overview of LSTM and GRU

First of all, we give a brief introduction to two prevailing RNN models, i.e., LSTM and GRU, as well as their bidirectional variants.

At each step, LSTM takes current token’s embedding $\mathbf{x}_t$ as input to update its cell state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$. Compared with an ordinary RNN, LSTM employs a forget gate $\mathbf{f}_t$, an input gate $\mathbf{i}_t$ and an output gate $\mathbf{o}_t$ to alleviate the gradient vanishing issue. Given previous hidden state $\mathbf{h}_{t-1}$ and previous cell state $\mathbf{c}_{t-1}$, current cell state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$ can be computed by:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{x}_t;\mathbf{h}_{t-1}]+\mathbf{b}_f),\\
\mathbf{i}_t &= \sigma(\mathbf{W}_I[\mathbf{x}_t;\mathbf{h}_{t-1}]+\mathbf{b}_I),\\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c[\mathbf{x}_t;\mathbf{h}_{t-1}]+\mathbf{b}_c),\\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{c}}_t,\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{x}_t;\mathbf{h}_{t-1}]+\mathbf{b}_o),\\
\mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{c}_t),
\end{aligned} \tag{1}$$

where $\mathbf{W}_f$, $\mathbf{W}_I$, $\mathbf{W}_c$ and $\mathbf{W}_o$ are weight matrices and $\mathbf{b}_f$, $\mathbf{b}_I$, $\mathbf{b}_c$ and $\mathbf{b}_o$ are bias vectors. σ is sigmoid function, [ ] denotes concatenation operation and $*$ indicates element wise product. The structure of an LSTM cell is illustrated in Figure 2.

GRU simplifies LSTM by using fewer gates. The transition equations of GRU are as follows:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{x}_t;\mathbf{h}_{t-1}]+\mathbf{b}_z),\\
\mathbf{r}_t &= \sigma(\mathbf{W}_r[\mathbf{x}_t;\mathbf{h}_{t-1}]+\mathbf{b}_r),\\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[\mathbf{x}_t;\mathbf{r}_t * \mathbf{h}_{t-1}]+\mathbf{b}_h),\\
\mathbf{h}_t &= (\mathbf{1}-\mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \tilde{\mathbf{h}}_t,
\end{aligned} \tag{2}$$

where $\mathbf{W}_z$, $\mathbf{W}_r$ $\mathbf{W}_h$ are weight matrices, and $\mathbf{b}_z$, $\mathbf{b}_r$, $\mathbf{b}_h$ are bias vectors. Figure 3 shows the structure of an GRU cell.

Since both LSTM and GRU can only process sequences unidirectionally, BiLSTM and BiGRU are proposed to eliminate the restriction. BiLSTM and BiGRU have two sequences of cells: one processes the input sequence from left to right and the other from right to left. Hence, each token in the input sequence has two hidden states, which are concatenated into bidirectional hidden states:

$$\left[\overleftrightarrow{\mathbf{h}}_1, \overleftrightarrow{\mathbf{h}}_2, ..., \overleftrightarrow{\mathbf{h}}_T\right] = \begin{bmatrix} \overrightarrow{\mathbf{h}}_1, \overrightarrow{\mathbf{h}}_2, ..., \overrightarrow{\mathbf{h}}_T \\ \overleftarrow{\mathbf{h}}_1, \overleftarrow{\mathbf{h}}_2, ..., \overleftarrow{\mathbf{h}}_T \end{bmatrix}, \tag{3}$$

where T denotes the length of the input sequence.

3.3 Intracellularly Sememe-incorporated RNNs

In this subsection, we describe our proposed three methods of incorporating sememe knowledge into the internal structures of RNN cells. Notice that all the three methods are applicable to both LSTM and GRU, both unidirectional and bidirectional RNNs. And we mark the newly added variables and formulae red for clarity.

Simple Concatenation

The first approach is straightforward by simply concatenating the sum of sememe embeddings of a word with the corresponding word embedding. And the sememe knowledge can be injected into the RNN cells by word embeddings. Formally, given a word x_t, its sememe set is denoted by $\mathcal{S}_t = \{s_1, \cdots, s_{|\mathcal{S}_t|}\}$, where $|\cdot|$ denotes the cardinality of a set. To incorporate sememes, we substitute original word embedding $\mathbf{x}_t$ with sememe added embedding $\tilde{\mathbf{x}}_t$:

$$\begin{aligned}
\boldsymbol{\pi}_t &= \sum_{s\in\mathcal{S}_t} \mathbf{s},\\
\tilde{\mathbf{x}}_t &= [\mathbf{x}_t;\boldsymbol{\pi}_t],
\end{aligned} \tag{4}$$

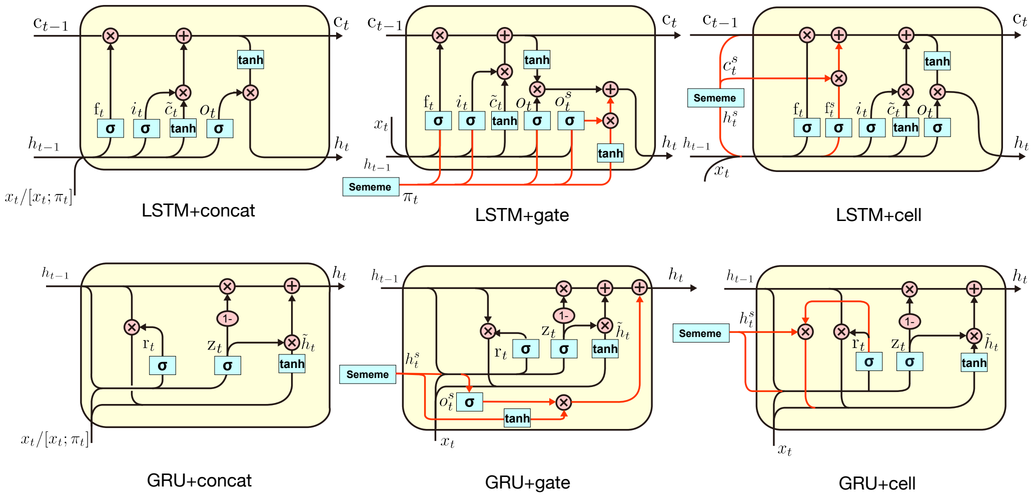

Figure 2: Structures of three intracellularly sememe-incorporated RNNs including LSTM-based models and GRU-based models.

where $\mathbf{s}$ denotes the sememe embedding of s and $\boldsymbol{\pi}_t$ represents the sememe knowledge embedding.

Adding Sememe Knowledge Output Gate

In the first method, sememe knowledge is indirectly injected into the RNN cells. Inspired by Ma et al. (2018), we propose the second method to directly incorporate sememes into RNN cells. More specifically, for LSTM, an additional sememe knowledge output gate o^s_t is added, which decides how much sememe knowledge will be encoded in the hidden state. The transition equations are as follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{x}_t;\mathbf{h}_{t-1};\boldsymbol{\pi}_t] + \mathbf{b}_f),\\
\mathbf{i}_t &= \sigma(\mathbf{W}_I[\mathbf{x}_t;\mathbf{h}_{t-1};\boldsymbol{\pi}_t] + \mathbf{b}_I),\\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c[\mathbf{x}_t;\mathbf{h}_{t-1}] + \mathbf{b}_c),\\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{c}}_t,\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{x}_t;\mathbf{h}_{t-1};\boldsymbol{\pi}_t] + \mathbf{b}_o),\\
\mathbf{o}^s_t &= \sigma(\mathbf{W}_{o^c}[\mathbf{x}_t;\mathbf{h}_{t-1};\boldsymbol{\pi}_t] + \mathbf{b}_{o^c}),\\
\mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{c}_t) + \mathbf{o}^s_t * \tanh(\mathbf{W}_c\boldsymbol{\pi}_t).
\end{aligned} \tag{5}$$

We can find that $\mathbf{o}^s_t * \tanh(\mathbf{W}_c\boldsymbol{\pi}_t)$ can directly add sememe knowledge to the original hidden state which only captures sequential information. In fact, this item functions as the sentinel vector used by Chen et al. (2017).

For GRU, we can add the sememe knowledge output gate in a similar way:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{x}_t;\mathbf{h}_{t-1};\boldsymbol{\pi}_t] + \mathbf{b}_z),\\
\mathbf{r}_t &= \sigma(\mathbf{W}_r[\mathbf{x}_t;\mathbf{h}_{t-1};\boldsymbol{\pi}_t] + \mathbf{b}_r),\\
\mathbf{o}^s_t &= \sigma(\mathbf{W}_o[\mathbf{x}_t;\mathbf{h}_{t-1};\boldsymbol{\pi}_t] + \mathbf{b}_o),\\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[\mathbf{x}_t;\mathbf{r}_t * \mathbf{h}_{t-1}] + \mathbf{b}_h),\\
\mathbf{h}_t &= (\mathbf{1} - \mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \tilde{\mathbf{h}}_t + \mathbf{o}^s_t \tanh(\boldsymbol{\pi}_t),
\end{aligned} \tag{6}$$

where $\mathbf{o}^s_t$ is the sememe knowledge output gate.

Introducing Sememe-RNN Cell

Although the second method adjusts RNN cell structure to incorporate sememes, the sememe knowledge may not be utilized sufficiently. In the retrofitted hidden state (e.g., $\mathbf{h}_t$ of LSTM in Equation (5)), the sequential information item $\mathbf{o}_t * \tanh(\mathbf{c}_t)$ bears the information of current word whose embedding has been processed by the forget gate, while the sememe knowledge item $\mathbf{o}^s_t * \tanh(\mathbf{W}_c\boldsymbol{\pi}_t)$ directly carries the embedding of sememe knowledge, where the two items are inconsistent.

To address the issue, we propose the third method. We regard the sememe knowledge as another information source like the previous token, and introduce an ordinary LSTM cell to process it. We feed the sememe knowledge embedding to the sememe-RNN cell and obtain the output including cell state and hidden state which encode the sememe knowledge. Then we design a forget gate for sememe knowledge and add the corresponding item to current cell state like previous cell state item. Formally, the transition equations are as follows:

$$\begin{aligned}
\mathbf{c}^s_t, \mathbf{h}^s_t &= LSTM^{(S)}(\boldsymbol{\pi}_t),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{x}_t;\mathbf{h}_{t-1};\mathbf{h}^s_t] + \mathbf{b}_f),\\
\mathbf{f}^s_t &= \sigma(\mathbf{W}^s_f[\mathbf{x}_t;\mathbf{h}_{t-1},\mathbf{h}^s_t] + \mathbf{b}^s_f),\\
\mathbf{i}_t &= \sigma(\mathbf{W}_I[\mathbf{x}_t;\mathbf{h}_{t-1};\mathbf{h}^s_t] + \mathbf{b}_I),\\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c[\mathbf{x}_t;\mathbf{h}_{t-1};\mathbf{h}^s_t] + \mathbf{b}_c),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{x}_t;\mathbf{h}_{t-1};\mathbf{h}^s_t] + \mathbf{b}_o),\\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{f}^s_t * \mathbf{c}^s_t + \mathbf{i}_t * \tilde{\mathbf{c}}_t,\\
\mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{c}_t),
\end{aligned} \tag{7}$$

where $\mathbf{c}^s_t$ and $\mathbf{h}^s_t$ are the cell state and hidden state

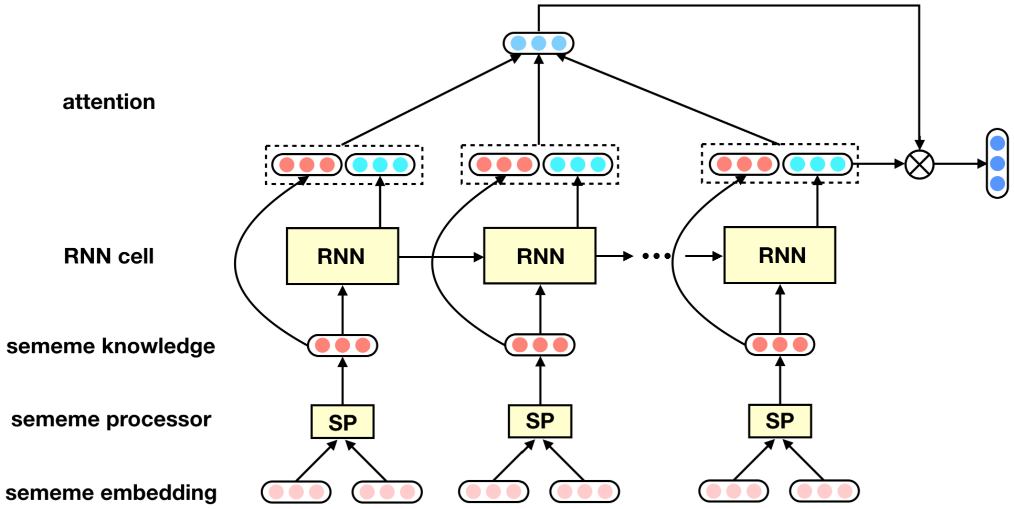

Figure 3: Overview of the sememe-incorporate LSTM framework. The lower part of this figure represents incorporating sememe knowledge into the internal structures of RNN cells, which we propose three different methods. And the upper part means incorporating sememe knowledge into hidden states.

of the sememe knowledge cell, and $\mathbf{f}_t^s$ is the corresponding forget gate.

Similarly, the transition equations for GRU are as follows:

$$\begin{aligned}
\mathbf{h}_t^s &= GRU^{(S)}(\boldsymbol{\pi}_t),\\
\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{x}_t; \mathbf{h}_{t-1}; \mathbf{h}_t^s] + \mathbf{b}_z),\\
\mathbf{r}_t &= \sigma(\mathbf{W}_r[\mathbf{x}_t; \mathbf{h}_{t-1}; \mathbf{h}_t^s] + \mathbf{b}_r),\\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[\mathbf{x}_t; \mathbf{r}_t * (\mathbf{h}_{t-1} + \mathbf{h}_t^s)] + \mathbf{b}_h),\\
\mathbf{h}_t &= (\mathbf{1} - \mathbf{z}_t) * \mathbf{h}_{t-1} + \mathbf{z}_t * \tilde{\mathbf{h}}_t,
\end{aligned} \tag{8}$$

where $\mathbf{h}_t^s$ is the hidden state of the sememe-GRU cell.

3.4 Extracellularly Sememe-incorporated RNNs

The above-mentioned three methods aim to incorporate sememe knowledge into the internal structures of RNN cells. In fact, we can also incorporate sememe knowledge into hidden states, which are directly related to the representation of input sequence. Inspired by Wang et al. (2016), we adopt a self-attention mechanism, where all the hidden states are considered together. As for the attention item, in the (query, key, value) triplet, the query is a trainable vector, the key is a matrix which contains all the hidden states as well as the sememe knowledge vectors of all the words in the input sequence, and the value is the hidden state matrix consisting of all the hidden states.

Formally, we define $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_T]$ as the hidden state matrix, and $\boldsymbol{\Pi} = [\boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_T]$ as the sememe knowledge matrix, then the final sequence representation is computed by :

$$\begin{aligned}
\mathbf{M} &= \begin{bmatrix} \mathbf{W}_h\mathbf{H} \\ \mathbf{W}_s\boldsymbol{\Pi} \end{bmatrix},\\
\boldsymbol{\alpha} &= \tanh(\mathbf{M}^T\boldsymbol{q}),\\
\mathbf{r} &= \mathbf{H}\boldsymbol{\alpha},\\
\mathbf{h}^* &= \tanh(\mathbf{W}_p\mathbf{r} + \mathbf{W}_x\mathbf{h}_T),
\end{aligned} \tag{9}$$

where $\boldsymbol{q}$ is the trainable query vector, $\mathbf{h}^*$ is the output sequence representation, and $\mathbf{W}_h$, $\mathbf{W}_s$, $\mathbf{W}_p$ and $\mathbf{W}_x$ are weight matrices. Notice that for the attention item $\boldsymbol{\alpha}$, we use activation function $\tanh$ rather than softmax function.

3.5 Combined Sememe-incorporated RNNs

In fact, we can combine each of the three intracellular sememe-incorporating methods in section 3.3 with the extracellular sememe-incorporating method in section 3.4 together.

4 Experiments

We evaluate our sememe-incorporated RNNs on several kinds of popular sequence modeling tasks, including language modeling (LM), sentence classification and sentence pair classification. And all the datasets we use are benchmark datasets. In addition, for the task of natural language inference (NLI), we also conduct a case study to demonstrate the effectiveness of sememes.

4.1 Language modeling

Dataset

We choose two benchmark LM datasets, namely Penn Tree Bank (PTB) (Marcus et al., 1993) and the WikiText-2 (Merity et al., 2016). PTB is made up of articles from the Wall Street Journal and contains 929k training tokens. Its vocabulary size is 10k. The WikiText-2 is derived from Wikipedia articles. It contains 2M training tokens and its vocabulary size is 33k.

Experimental Settings

We choose vanilla LSTM and GRU as the baseline methods. Because of the characteristic of LM task, we cannot incorporate sememes into hidden states. Therefore, only intracellularly sememe-incorporated RNNs are evaluated. And we use “+concat”, “+gate” and “+cell” as abbreviations of the three methods.

Following previous works, we try two sets of hyper-parameters including “medium” and “large”. For “medium”, the dimension of hidden states and word/sememe embeddings is set to 650, batch size is 20, bptt is 35 and dropout rate is 0.5. For “large”, the dimension of vectors is 1500, dropout rate is 0.65, and other hyper-parameters are the same as “medium”. The above-mentioned hyper-parameter settings are applied to both LSTM and GRU, both baseline methods and our models.

We also adopt the same training strategy for all the models. We choose SGD as the optimizer, whose initial learning rate is 20 for LSTM and 10 for GRU. The learning rate would be divided by 4 if no improvement is observed on the validation set. The maximum training epoch number is 40 and the gradient norm clip boundary is 0.25. In addition, all the word and sememe embeddings are randomly initialized.

Experimental Results

Table 1 shows the perplexity results of baseline methods and our models on the two benchmark datasets, where the results on both validation and test sets are given. From the table, we can observe that:

(1) All of the intracellularly sememe-incorporated RNNs achieve lower perplexity as compared to corresponding baseline method. And even the simplest “+concat” method decreases perplexity by about 3 on average. These results demonstrate the usefulness of sememes.

(2) Among the three different methods, “+cell” performs the best for LSTM and “+gate” performs the best for GRU, no matter using “medium” or “large” hyper-parameters.

Dataset	PTB		WikiText-2	
Model	Val	Test	Val	Test
LSTM(medium)	85.45	81.75	99.49	93.96
+concat	82.95	79.66	95.91	90.94
+gate	81.18	78.24	95.27	90.19
+cell	**81.12**	**77.58**	**94.58**	**89.35**
LSTM(large)	81.88	78.34	95.91	90.75
+concat	78.86	75.81	92.22	86.93
+gate	**77.09**	**73.95**	**90.18**	86.29
+cell	78.74	75.27	90.77	**85.62**
GRU(medium)	94.84	91.05	109.11	103.07
+concat	90.68	87.29	105.01	98.89
+gate	**88.87**	**85.12**	**103.13**	**96.97**
+cell	89.56	86.49	103.27	97.95
GRU(large)	92.68	89.6	107.75	101.52
+concat	90.99	87.57	104.56	97.97
+gate	**88.28**	**84.56**	102.84	96.71
+cell	88.93	85.74	**101.63**	**95.7**

Table 1: Perplexity results of baseline methods and our models on the PTB and WikiText-2 datasets.

4.2 Natural Language Inference

NLI, also known as Recognizing Textual Entailment (RTE), is a classical sentence pair classification task and has scale-appropriate benchmark datasets. We evaluate the ability to learn sentence representation of our models on the NLI task.

Dataset

We choose the most famous benchmark dataset, Stanford Natural Language Inference (SNLI) dataset. It contains 570k sentence pairs, which are manually classified into 3 categories, namely entailment, contradiction and neutral.

Experimental Settings

For the task of NLI, the whole input sequence can be seen during training. As a result, models of both directions and extracellularly sememe-incorporated RNNs can be utilized. Correspondingly, we select vanilla LSTM, GRU and their bidirectional variants BiLSTM and BiGRU as baseline methods. And we evaluate intracellularly sememe-incorporated RNNs (“+concat”, “+gate” and “+cell”), extracellularly sememe-incorporated RNNs, and the ensemble methods which combine both sememe-incorporating ways.

For all the models, we use the same hyper-parameters and training strategy. We use 300-dimensional word embeddings pre-trained by GloVe (Pennington et al., 2014), which are frozen during training. As for the sememe embeddings, their dimension is also set to 300 and they are randomly initialized using a normal distribution with mean 0 and variance 0.05. And the dimension of hidden states is 2048.

We employ a three-layer multi-layer perception network (MLP) plus a three-way softmax layer as the classifier, whose hidden layer size is 512 and input is a feature vector constructed from the embeddings of a pair of sentences. Following (Bowman et al. (2016); Mou et al. (2016)), we ran our models separately on two input sentences to obtain their embeddings $\mathbf{h}_{pre}$ and $\mathbf{h}_{hyp}$, then the feature vector $\mathbf{v}$ can be obtained by:

$$\mathbf{v} = \begin{bmatrix} \mathbf{h}_{pre} \\ \mathbf{h}_{hyp} \\ |\mathbf{h}_{pre} - \mathbf{h}_{hyp}| \\ \mathbf{h}_{pre} * \mathbf{h}_{hyp} \end{bmatrix}. \tag{10}$$

As for training, we still choose the SGD optimizer, whose initial learning rate is 0.1 and the weight factor is 0.99. And we divide the learning rate by 5 if no improvement is observed on the validation dataset.

Model	LSTM	GRU	BiLSTM	BiGRU
vanilla	80.58	81.33	81.67	81.40
+concat	81.39	82.15	81.60	82.48
+gate	81.35	82.18	81.91	82.20
+cell	81.39	**82.75**	82.16	**83.84**
+extra	82.14	81.55	82.20	82.32
+extra+concat	82.58	82.65	82.63	82.65
+extra+gate	**82.91**	82.55	82.76	82.50
+extra+cell	82.22	82.45	**83.00**	82.77

Table 2: Accuracy results of our models and baseline methods on the test set of SNLI dataset.

Experimental Results

Table 2 lists the accuracy results of all the models on the test set of SNLI. From this table, we can see that:

(1) All of our models achieve marked performance enhancement as compared with corresponding baseline method, which proves the usefulness of sememes in improving the sentence representation learning ability of RNNs.

(2) Among intracellularly sememe-incorporated RNNs, "+cell" achieves the best performance on whichever basic framework, which manifests the superiority of "+cell" in utilizing sememe knowledge.

(3) Extracellularly sememe-incorporated methods work better on LSTM and BiLSTM than on GRU and BiGRU. In addition, combined sememe-incorporated RNNs outperform the corresponding model only with extracellular method itself. And for LSTM and BiLSTM, the ensemble models also perform better than the models which incorporate sememes into RNN cells only. But for GRU and BiGRU, the models with "+cell" outperform the ensemble models.

Entailment Example
Hypothesis: Four men stand in a circle facing each other *playing* [`perform`, `reaction`, `MusicTool`] brass *instruments* [`MusicTool`, `implement`] which people watch them.
Premise: The men are playing *music* [`music`].

Contradict Example
Hypothesis: A group of women playing volleyball *indoors* [`location`, `house`, `internal`].
Premise: People are *outside* [`location`, `external`] tossing a ball.

Table 3: Two examples of hypothesis-premise pairs in the SNLI dataset. The words in italic are important words and their sememes are listed.

Case Study

Here we present two examples of how sememes are beneficial to handling the task of NLI. Table 3 exhibits two sentence pairs. The true relation between the sentences of the first sentence pair is "entailment". Our models yield the correct result while the baseline methods not. We notice that there are several important words whose sememes provide useful information. In the hypothesis sentence, both the words "playing" and "instruments" have the sememe `MusicTool`, which is semantically related to the sememe `music` of the word "music" in the premise sentence. We speculate that the semantic relatedness given by sememes assists our models in coping with this sentence pair.

As for the second sentence pair, whose relation is "contradict", is also classified correctly by our models. We find that the word "indoors" in the hypothesis sentence has the sememe `internal`, while the word "outside" in the premise sentence has the sememe `external`. The two sememes are a pair of opposites, which may explain why our models make the right judgment.

4.3 Other Sequence Modeling Tasks

To evaluate the sequence modeling ability of our models more thoroughly, we carry out experiments on other sentence modeling datasets.

	Model	LSTM				GRU			
	Task	SUBJ	MRPC	SICKE	SICKR	SUBJ	MRPC	SICKE	SICKR
Uni	vanilla	87.16	66.55	82.65	84.30	90.14	65.57	82.34	85.63
	+concat	86.16	72.06	82.61	84.79	89.33	72.87	83.28	**86.08**
	+gate	86.66	72.81	79.87	81.92	89.02	72.93	80.68	84.74
	+cell	87.44	72.41	80.07	84.17	89.52	**73.22**	82.39	85.28
	+att+void	**91.42**	**73.97**	83.95	85.38	90.24	72.64	82.34	78.87
	+extra+concat	90.05	71.54	**83.99**	**85.44**	**90.31**	68.87	81.83	80.47
	+extra+gate	90.63	69.39	82.77	85.29	90.05	75.28	82.24	81.38
	+extra+ ecll	90.93	71.83	83.44	84.99	89.46	72.17	**83.29**	77.12
Bi	vanilla	90.55	69.10	82.40	85.54	90.96	72.12	80.68	85.42
	+concat	90.25	71.65	83.09	86.25	91.05	72.31	83.02	85.93
	+gate	90.25	64.29	81.12	83.93	90.98	72.43	82.83	85.88
	+cell	90.08	71.30	82.28	85.40	90.60	72.93	82.10	85.80
	+extra+void	91.31	68.00	83.62	85.83	91.32	71.49	83.49	86.95
	+extra+concat	**91.57**	72.06	83.21	**87.23**	91.43	**73.15**	**84.46**	87.46
	+extra+gate	90.78	68.29	83.68	86.5	**91.57**	72.35	83.51	87.43
	+extra+cell	91.23	**74.32**	**84.15**	84.82	91.18	72.28	84.21	**88.19**

Table 4: Accuracy results of our models and baseline methods on the test sets of several sequence or sentence pair classification datasets.

Dataset

We choose four benchmark datasets: (1) SUBJ: a sentence-level sentiment analysis dataset containing 20k sentences which are actually customer reviews of goods on Amazon, and all the sentences are classified into two categories including positive and negative; (2) MRPC: a sentence paraphrase dataset containing 5.8k sentence pairs, and each sentence pair is labeled with 1 or 0 indicating whether there are paraphrase relation between the two sentences; (3) SICK-E: another NLI benchmark dataset containing 9.4k sentence pairs which are also classified into the three same categories as SNLI; (4) SICK-R: a semantic relatedness dataset comprising 9.4k sentence pairs and each sentence pair is labelled with an integer score between 0 and 5, which measures how related the pair of sentences are.

Experimental Settings

Following Conneau et al. (2017), we pretrain our models on the SNLI dataset and then transfer the models to the other datasets. In addition, we adopt the SentEval framework (Conneau and Kiela, 2018) as the evaluation tool, and we use its default settings.

Experimental Results

The results are listed in Table 4. It can be observed that most of our models outperform the baseline models. In general, incorporating sememes using both intracellular and extracellular method achieves the best performance. We also find that our models perform better in SICK-E/R than other tasks compared with prevailing architectures. We argue that it is because SICK-E/R is the same type of tasks with SNLI.

5 Conclusion and Future work

In this paper, we propose to incorporate sememe knowledge into RNNs for better text sequence modeling. In specific, we first employ three methods to incorporate sememes into RNN cells. Further, we propose a retrofitting mechanism to integrate sememes on the hidden states of RNNs. We combine both of them to obtain the overall models. We evaluate our model on the benchmark datasets of language modeling, natural language inference, sentence representations and transfer tasks to demonstrate the effectiveness of our models.

For future work, we plan the following research directions:

(1) HowNet organizes the word meaning in the form of sememe tree which contains structural information between different sememes. Such structural information can also be explored in future work. (2) In this paper, we compute the sum of all sememes of a word as its sememe embeddings. Intuitively, selecting and integrating relevant sememes according to the context would improve our models. (3) As a general linguistic knowledge, sememes should be useful in other framework as well. One can investigate the possibility to incorporate sememes into other frameworks.

References

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*.

Philip Bachman and Doina Precup. 2015. Variational generative stochastic networks with collaborative shaping. In *ICML*, pages 1964–1972.

Leonard Bloomfield. 1926. A set of postulates for the science of language. *Language*, 2(3):153–164.

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1466–1477.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *proceedings of EMNLP*.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *proceedings of EMNLP*.

Lei Dang and Lei Zhang. 2010. Method of discriminant for chinese sentence sentiment orientation based on hownet. *Application Research of Computers*, 4:43.

Zhendong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *Proceedings of NLP-KE*.

Xiangyu Duan, Jun Zhao, and Bo Xu. 2007. Word sense disambiguation through sememe labeling. In *Proceedings of IJCAI*, pages 1594–1599.

Xianghua Fu, Guo Liu, Yanyan Guo, and Zhiqiang Wang. 2013. Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon. *Knowledge-Based Systems*, 37:186–195.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*, pages 799–804. Springer.

Yihong Gu, Jun Yan, Hao Zhu, Zhiyuan Liu, Ruobing Xie, Maosong Sun, Fen Lin, and Leyu Lin. 2018. Language modeling with sparse product of sememe experts. In *Proceedings of EMNLP*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huiming Jin, Hao Zhu, Zhiyuan Liu, Ruobing Xie, Maosong Sun, Fen Lin, and Leyu Lin. 2018. Incorporating chinese characters of words for lexical sememe prediction. In *Proceedings of ACL*.

Qun Liu and Sujian Li. 2002. Word similarity computing based on hownet. *International Journal of Computational Linguistics & Chinese Language Processing*, 7(2):59–76.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

George Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 130.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pages 1–18.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved word representation learning with sememes. In *Proceedings of ACL*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *proceedings of EMNLP*.

Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. *proceedings of EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Fanchao Qi, Yankai Lin, Maosong Sun, Hao Zhu, Ruobing Xie, and Zhiyuan Liu. 2018. Cross-lingual lexical sememe prediction. In *Proceedings of EMNLP*.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 1988. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Jürgen Schmidhuber. 1992. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242.

Robert Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Anna Wierzbicka. 1996. *Semantics: Primes and universals: Primes and universals*. Oxford University Press, UK.

Ruobing Xie, Xingchi Yuan, Zhiyuan Liu, and Maosong Sun. 2017. Lexical sememe prediction via word embeddings and matrix factorization. In *Proceedings of AAAI*.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into lstm with recall gate for conversation modeling. *proceedings of IJCNN*, 3.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of ACL*, pages 1436–1446.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of AAAI*.

Xiangkai Zeng, Cheng Yang, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Chinese liwc lexicon expansion via hierarchical classification of word embeddings with sememe attention. In *Proceedings of AAAI*.

Yuntao Zhang, Ling Gong, and Yongcheng Wang. 2005. Chinese word sense disambiguation using hownet. In *Proceedings of International Conference on Natural Computation*.