Chinese LIWC Lexicon Expansion via Hierarchical Classification of Word Embeddings with Sememe Attention

Xiangkai Zeng,[*‡] **Cheng Yang,**[§] **Cunchao Tu,**[§] **Zhiyuan Liu,**[†§¶] **Maosong Sun**[§¶]
[‡]School of Computer Science and Engineering, Beihang University, Beijing, China
[§]Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, Beijing, China
[¶]Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, Xuzhou, China

Abstract

Linguistic Inquiry and Word Count (LIWC) is a word counting software tool which has been used for quantitative text analysis in many fields. Due to its success and popularity, the core lexicon has been translated into Chinese and many other languages. However, the lexicon only contains several thousand of words, which is deficient compared with the number of common words in Chinese. Current approaches often require manually expanding the lexicon, but it often takes too much time and requires linguistic experts to extend the lexicon. To address this issue, we propose to expand the LIWC lexicon automatically. Specifically, we consider it as a hierarchical classification problem and utilize the Sequence-to-Sequence model to classify words in the lexicon. Moreover, we use the sememe information with the attention mechanism to capture the exact meanings of a word, so that we can expand a more precise and comprehensive lexicon. The experimental results show that our model has a better understanding of word meanings with the help of sememes and achieves significant and consistent improvements compared with the state-of-the-art methods. The source code of this paper can be obtained from `https://github.com/thunlp/Auto_CLIWC`.

Introduction

Linguistic Inquiry and Word Count (Pennebaker, Booth, and Francis 2007, LIWC) has been widely used for computerized text analysis in social science. LIWC groups words into manually-defined coarse-to-fine grained categories, and it was originally developed to address content analytic issues in experimental psychology. Nowadays, there is an increasing number of applications across fields such as computational linguistics (Grimmer and Stewart 2013), demographics (Newman et al. 2008), health diagnostics (Bucci and Maskit 2005), social relationship (Kacewicz et al. 2014), etc.

Chinese is the most spoken language in the world (Lewis et al. 2009), but we cannot use the original LIWC to analyze Chinese text. Fortunately, Chinese LIWC (Huang et al. 2012) has been released to fill the vacancy. In this paper, we mainly focus on Chinese LIWC and use LIWC to stand for Chinese LIWC if not otherwise specified.

While LIWC has been used in a variety of fields, its lexicon only contains less than 7,000 words. This is insufficient because according to (Li and others 2008), there are at least 56,008 common words in Chinese. Moreover, LIWC lexicon does not consider the emerging words and phrases on the Internet. Therefore, it is reasonable and necessary to expand the LIWC lexicon so that it is more accurate and comprehensive for scientific research. One way to expand LIWC lexicon is to annotate the new words manually. However, it is too time-consuming and often requires language expertise to add new words. Hence, we propose to automatically expand LIWC lexicon. To the best of our knowledge, we are the first to expand LIWC lexicon automatically.

Automatically LIWC lexicon expansion faces the problems of **polysemy** and **indistinctness**. **Polysemy** means words and phrases have multiple meanings and are thereby classified into multiple irrelevant categories. **Indistinctness** means many categories in LIWC are fine-grained, and thus making it more difficult to distinguish them.

One important feature in LIWC lexicon is that categories form a tree structure hierarchy. Therefore, hierarchical classification algorithms such as Hierarchical SVM (Support Vector Machine) (Chen, Crawford, and Ghosh 2004), can be easily applied to LIWC lexicon expansion. However, these methods are often too generic, without considering the polysemy property of words and LIWC lexicon and indistinctness property of LIWC categories.

To address these issues, we propose to incorporate external annotated word information when expanding the lexicon. In linguistics, each word has one or more senses, and each sense consists of one or more sememes, which are defined as the smallest semantic language unit of meaning (Bloomfield 1926). In this paper, we use HowNet (Dong and Dong 2003) where words are annotated with relevant sememes. For polysemy problem, sememes can explicitly express different meanings of a word and make it possible to assign multiple labels. For indistinctness problem, sememe is also helpful in differentiating fine-grained categories since we have more detailed semantic information about the word meanings. Moreover, sememes should be given different weights

[*]This work was done when the first author was visiting Tsinghua University.

[†]Corresponding author: Zhiyuan Liu (liuzy@tsinghua.edu.cn).

when classifying words into different categories. Thus, we propose to use attention mechanism to better utilize sememe information. The experimental results show that our model achieves significant improvements as compared to the state-of-the-art methods.

The contributions of this paper can be summarized as follows:

- To the best of our knowledge, our model is the first attempt to expand LIWC lexicon automatically.
- To address the polysemy and indistinctness problems, we propose to use sememe information to distinguish meanings among words.
- To better utilize sememe information, we use attention mechanism to assign different weights to sememes when classifying each word.
- In the experiments, we show that our model outperforms the state-of-the-art methods significantly.

Related Work

In this section, we first introduce some previous works based on LIWC and then describe recent research in the hierarchical classification. Lastly, we briefly mention studies about HowNet.

The original English version of LIWC lexicon is one of the most well-known dictionaries in quantitative text analysis. It was first released in the early 1990s and has been updated several times, with the latest version released in 2015 (Pennebaker et al. 2015). Over these years, there have emerged numerous scientific findings across different fields with the help of the original lexicon. (Mehl et al. 2007) found that women and men both spoke about 16,000 words per day, dispelling the myth of female talkativeness. (Bucci and Maskit 2007) showed that word counting approach tends to be less biased than clinician's self-reports in therapeutic improvements. (Rohrbaugh et al. 2008) found that the use of *we* indicated relationship closeness and was even predictive of heart failure improvements. (Schwartz et al. 2013) analyzed personality, gender, and age in social media using an open-vocabulary approach. Due to the success and the importance of the original LIWC, (Huang et al. 2012) manually developed the first Chinese LIWC, and there is an increasing number of applications (Gao et al. 2013; Yu et al. 2016; Li et al. 2014) based on it. However, as a manually annotated dictionary, this lexicon only contains less than 7,000 words, which is a serious limitation compared to the number of common words. Hence, we propose to expand the lexicon automatically.

To the best of our knowledge, most previous works on lexicon expansion are based on feature engineering techniques (Bravo-Marquez, Frank, and Pfahringer 2015; Bravo-Marquez et al. 2016). Therefore, it requires lots of human knowledge to design features for a different lexicon and makes it non-trivial to adopt previous methods for LIWC expansion. Also, many of them cannot be formulated as a classification problem (Coden et al. 2014; da Silva Guimarães 2016). Since the categories in LIWC annotation form a tree hierarchy, we can adopt hierarchical classification methods for automatically LIWC expansion. As the first effort on LIWC lexicon expansion, we believe that the works in hierarchical classification problems are more correlated to LIWC expansion and more suitable for baseline comparisons.

For hierarchical classification, (Silla Jr and Freitas 2011) summarized a variety of them from different fields and categorized them into five approaches. Flat Classifier (Barbedo and Lopes 2007) is the simplest one to deal with hierarchical classification problems. In this approach, the classifier completely ignores the hierarchy and predicts only classes at the leaf nodes. Local Classifier Per Node (Fagni and Sebastiani 2007) consists of training one binary classifier for class. Local Classifier Per Parent Node (Silla and Freitas 2009) is the approach where, for each parent node in the class hierarchy, a multiclass classifier is trained to distinguish among its child nodes. Similar to Local Classifier Per Parent Node approach, Local Classifier Per Level (Clare and King 2003) trains one multiclass classifier, but at each level instead of each node. The last one is Global Classifier (Kiritchenko et al. 2006), where a single classification model is trained to take into account the hierarchy as a whole.

In recent years, there have been some attempts to use neural networks for hierarchical classification. (Cerri, Barros, and De Carvalho 2014) trained the multilayer perceptron level by level, and used the predictions of the neural network as inputs for the next neural network associated with the next hierarchical level. (Karn, Waltinger, and Schütze 2017) proposed to use RNN encoder-decoder for entity mention classification, which is a problem with a hierarchical class structure. The encoder-decoder performs classification by generating paths in the hierarchy from top node to leaf nodes. However, due to the polysemy and indistinctness problems, these methods are not suitable for LIWC expansion. Therefore, we propose to incorporate sememe information.

During these years, people manually annotate sememes and build many linguistic knowledge bases. HowNet (Dong and Dong 2003) is a typical knowledge base with annotated sememes, and it has far more words than LIWC. Therefore, we can use it directly to help expand the LIWC lexicon. HowNet has also been widely used in sentiment analysis (Fu et al. 2013) and word representation learning (Niu et al. 2017).

Problem Formalization

In this section, we first give an illustrative example of words and categories in LIWC lexicon. Then we give the formal definition of LIWC lexicon expansion problem.

Figure 1 is an example which demonstrates the LIWC categories of the word *apex*. We can see that the word *apex* belongs to two parent categories, namely *PersonalConcerns* and *relative*. The two parent categories have one child category each, which are *achieve* and *space* respectively.

As illustrated in Figure 1, LIWC categories are hierarchically organized and each word in LIWC can belong to more than one category at any given level of the hierarchy. In other words, it is considered to be *hierarchical multilabel*. Moreover, each word does not necessarily belong to a leaf cate-

gory. This is often referred to as non-mandatory prediction. Therefore, LIWC lexicon expansion is a **non-mandatory hierarchical multilabel classification** problem.

Formally, we follow the proposed framework for hierarchical classification problems in (Silla Jr and Freitas 2011). The LIWC lexicon expansion problem is described as the 3-tuple $< T, MPL, PD >$, where:

- T indicates that the classes are arranged into a tree structure;
- MPL (Multiple Paths of Labels) is equivalent to the term *hierarchical multilabel*;
- PD (Partial Depth Labeling) indicates that some instances have a partial depth of labeling, i.e., the value of the class label at some level (typically the leaf level) is unknown.

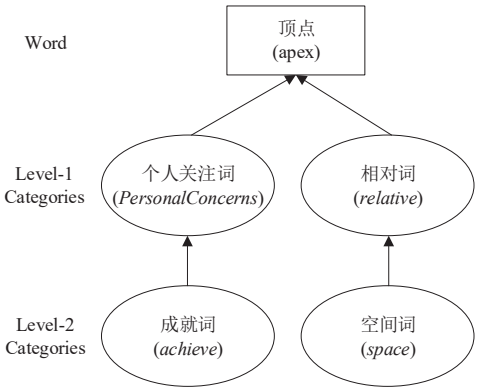

Figure 1: Example word *apex* and its categories in LIWC lexicon.

Methodology

In this section, we introduce our framework Hierarchical Decoder with Sememe Attention. It exploits the Sequence-to-Sequence decoder for hierarchical classification and utilizes the attention mechanism to incorporate sememe information to expand a better LIWC lexicon.

In the following sections, we first describe the structure of HowNet. Next, we introduce the decoder to hierarchically predict word labels. Finally, we propose to use attention mechanism in the decoder for incorporating sememe information into the model.

HowNet Structure

In this section, we discuss words, sememes and their relationship in HowNet. Each word in HowNet has one or multiple senses. For each sense, it is annotated by sememes and sememes can form complicated relations.

Figure 2 is the example word *apex* and its relating sememes. As illustrated in Figure 2, the word *apex* has two main senses: one means the highest achievement or excellence (*acme*), and another means a concept in geometry (*vertex*). The following layers are sememes annotating each sense. Sememe is usually considered as a semantic

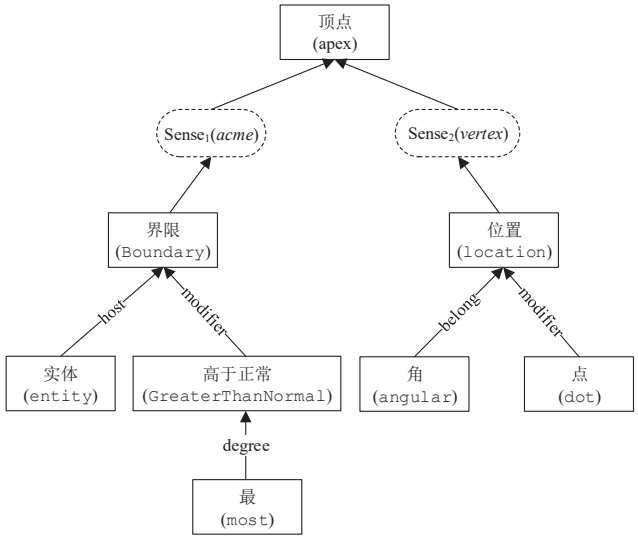

Figure 2: Example word *apex* and its senses and sememes in HowNet annotation.

language unit of meaning. For example, the sense *vertex* has sememes `dot`, `angular` and `location`, indicating that `dot`, `angular` and `location` represent their literal meaning and they serve as the basic unit of the sense *vertex*.

We can observe from Figure 2 that in HowNet, sememes and senses have various relations, such as *host* and *modifier*, and through the relations, they form complicated hierarchical structures. In this work, we only consider all sememes of each word as a sememe set without exploring their complicated relations, which is left in our future work.

Comparing Figure 1 with Figure 2, we can find out that sense *acme* and its relating sememes represent a specific meaning in word *apex* which can be classified into *PersonalConcerns* and *achieve*, and so is sense *vertex*. In other words, annotated sememes in HowNet can help us distinguish different meanings in words, and hence are useful in LIWC lexicon expansion.

Hierarchical Decoder

We model the hierarchical classification problem as a Sequence-to-Sequence decoding, where the input is the word embedding and the output is its hierarchical labels. The Sequence-to-Sequence models have been widely used in NLP for modeling sentences (Sutskever, Vinyals, and Le 2014).

Formally, let Y denotes the label set and π:$Y \rightarrow Y$ denotes the parent relationship where $\pi(y)$ is the parent node of $y \in Y$. Given a word x, its labels form a tree structure hierarchy. We then choose each path from root node to leaf node, and transform it into a sequence $\mathbf{y} = (y_1, y_2, \ldots, y_L)$ where $\pi(y_i) = y_{i-1}, \forall i \in [2, L]$ and L is the number of levels in the hierarchy. In this way, when the Hierarchical Decoder (HD) predicts a label y_i, it takes into consideration the probability of parent label sequence (y_1,...,y_{i-1}). Formally, the decoder defines a probability over the label sequence $\mathbf{y}$:

$$p(\mathbf{y}) = \prod_{i=1}^{L} p(y_i|(y_1, \ldots, y_{i-1}), x), \quad (1)$$

where $\mathbf{y} = (y_1, \ldots, y_L)$. A common approach for decoder is to use LSTM (Hochreiter and Schmidhuber 1997) so that each conditional probability is computed as

$$\begin{aligned} p(y_i|(y_1, \ldots, y_{i-1}), x) &= f(y_{i-1}, s_i) \\ &= o_i \circ \tanh(s_i), \end{aligned} \quad (2)$$

where

$$\begin{aligned} s_i &= f_i \circ s_{i-1} + z_i \circ \tilde{s}_i, \\ \tilde{s}_i &= \tanh(W_s \cdot [s_{i-1}, y_{i-1}] + b_s), \\ o_i &= \sigma(W_o \cdot [s_{i-1}, y_{i-1}] + b_o), \\ z_i &= \sigma(W_z \cdot [s_{i-1}, y_{i-1}] + b_z), \\ f_i &= \sigma(W_f \cdot [s_{i-1}, y_{i-1}] + b_f), \end{aligned} \quad (3)$$

where $\circ$ is an element-wise multiplication, σ represents sigmoid function and s_i is the i-th hidden state of the LSTM. W_s, W_o, W_z, W_f are weights and b_s, b_o, b_z, b_f are biases. o_i, z_i and f_i are known as output gate layer, input gate layer and forget gate layer respectively.

To take advantage of word embeddings, we define the initial state $s_0 = e_x$ where e_x represents the embedding of the word. In other words, the word embeddings are applied as the initial state of the decoder.

Specifically, the input of Hierarchical Decoder is word embeddings and label embeddings. First, we transform raw words into word embeddings by an embedding matrix $\mathbf{V} \in \mathbb{R}^{|V| \times d_w}$, where d_w is the word embedding dimension. Then, at each time step, we input label embeddings y, which is obtained by a label embedding matrix $\mathbf{Y} \in \mathbb{R}^{|Y| \times d_y}$, where d_y is the label embedding dimension. Here word embeddings are pre-trained and fixed during training.

Generally speaking, Hierarchical Decoder is expected to decode word labels hierarchically based on word embeddings. At each time step, it will predict the current label depending on previously predicted labels.

Hierarchical Decoder with Sememe Attention

The Hierarchical Decoder uses word embeddings as the initial state, then predicts word labels hierarchically as sequences. However, each word in the Hierarchical Decoder model has only one representation. This is insufficient because it is difficult to handle polysemy and indistinctness using a single real-valued vector. Therefore, we propose to incorporate sememe information.

Because different sememes represent different meanings of a word, they should have different weights when predicting word labels. Moreover, the same sememe should have different weights in different categories. Take the word *apex* in Figure 2 for example. The sememe `location` should have a relatively higher weight when the decoder chooses among the sub-classes of *relative*. However, when choosing among the sub-classes of *PersonalConcerns*, `location` should have a lower weight because it represents a relatively irrelevant sense *vertex*.

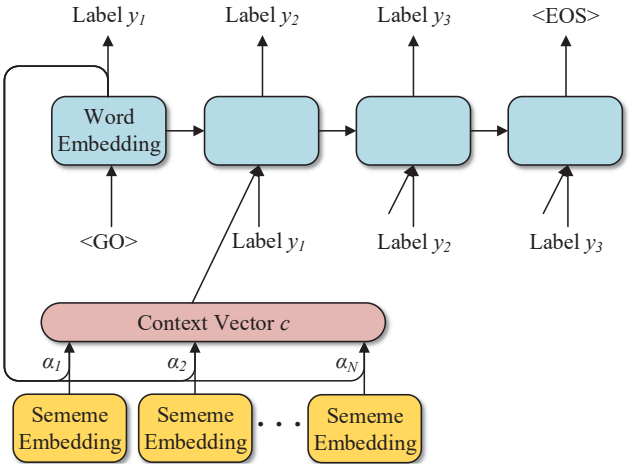

Figure 3: HDSA with word embeddings as the initial state.

To achieve the above goals, we propose to utilize the attention mechanism (Bahdanau, Cho, and Bengio 2014) to incorporate sememe information when decoding word label sequence. The structure of Hierarchical Decoder with Sememe Attention (HDSA) is illustrated in Figure 3.

Similar to the Hierarchical Decoder approach, we apply word embeddings as the initial state of the decoder. The primary difference is that the conditional probability is defined as:

$$p(y_i|(y_1, \ldots, y_{i-1}), x, c_i) = f([y_{i-1}, c_i], s_i), \quad (4)$$

where c_i is known as context vector. The context vector c_i depends on a set of sememe embeddings $\{h_1, \ldots, h_N\}$, acquired by a sememe embedding matrix $\mathbf{S} \in \mathbb{R}^{|S| \times d_s}$, where d_s is the sememe embedding dimension.

To be more specific, the context vector c_i is computed as a weighted sum of the sememe embedding h_j:

$$c_i = \sum_{j=1}^{N} \alpha_{ij} h_j. \quad (5)$$

The weight α_{ij} of each sememe embedding h_j is defined as

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{N} \exp(e_{ik})}, \quad (6)$$

where

$$e_{ij} = v^T \tanh(W_1 y_{i-1} + W_2 h_j), \quad (7)$$

is regarded as a score which indicates how well the sememe embedding h_j is associated when predicting the current label y_i. Here, $v \in \mathbb{R}^a, W_1 \in \mathbb{R}^{a \times d_y}$ and $W_2 \in \mathbb{R}^{a \times d_s}$ are weight matrices. a is the number of hidden units in attention model.

Intuitively, at each time step, HDSA chooses which sememes to pay attention to when predicting the current word label. In this way, different sememes can have different weights, and the same sememe can have different weights in different categories. With the support of sememe attention, HDSA can deal with the polysemy and indistinctness problems and thus can expand a more accurate and comprehensive LIWC lexicon.

Optimization and Implementation Details

Here, we present the implementation and optimization details of our model. The objective function is defined using cross entropy.

$$J = -\frac{1}{T}\sum_{n=1}^{T}\sum_{m}(y_{mn}\log(y'_{mn})), \tag{8}$$

where $y_{mn} \in \{0, 1\}$ represents whether word w_n belongs to label y_m, y'_{mn} represents the predicted probability of word w_n belonging to label y_m, computed by Equation (4), and T is the number of words. We use the Adam algorithm (Kingma and Ba 2014) to automatically adapt the learning rate of each parameter.

In the implementation, we utilize the LSTM network. When inferring word labels, we employ beam search to address the hierarchical multilabel issue. We empirically set a threshold δ and assign a label sequence $\mathbf{y}$ to a word only when the following inequality constraint is met.

$$\log p(\mathbf{y}) > \delta. \tag{9}$$

We also employ Recurrent Dropout (Semeniuta, Severyn, and Barth 2016) and Layer Normalization (Ba, Kiros, and Hinton 2016) to the LSTM to prevent overfitting. The Layer Normalization is applied before the internal non-linearity.

Experiments

Our experiments are intended to demonstrate the effectiveness of Hierarchical Decoder with Sememe Attention on LIWC lexicon expansion.

Dataset

We use the Chinese LIWC developed by (Huang et al. 2012) and its hierarchy depth is 3. we list its statistics in Table 1.

	Num. words	Num. labels
Overall	6,828	51
Level 1	6,828	10
Level 2	6,363	34
Level 3	589	7

Table 1: Statistics of LIWC lexicon.

We employ the sememe annotation in HowNet. The total number of words in HowNet is over 100,000 and the number of distinct sememes used in this paper is 1,617. We use Sogou-T as the corpus to learn both word embeddings and sememe embeddings. Sogou-T contains over 130 million web pages and is supplied by a Chinese commercial search engine.

Algorithms for Comparison

Since LIWC lexicon expansion is a hierarchical classification problem, we mainly choose hierarchical algorithms for comparison.

- **Top-down k-NN** (TD k-NN): Top-down decision making using k-NN at each parent label.
- **Top-down SVM** (TD SVM): Top-down decision making using SVM at each parent label.
- **Structural SVM** (Joachims, Finley, and Yu 2009): a margin rescaled structural SVM using the 1-slack formulation and cutting plane method. We use Pystruct[1] (Müller and Behnke 2014) for implementation.
- **CSSA** (Condensing Sort and Select Algorithm) (Bi and Kwok 2011): a hierarchical classification algorithm which can be used on both tree- and DAG-structured hierarchies.
- **HD** (Hierarchical Decoder): the Hierarchical Decoder *without* sememe attention.

The above Top-down approaches are also referred to as Local Classifier Per Parent Node Approach in (Silla Jr and Freitas 2011).

Experimental Settings

The word embedding matrix $\mathbf{V}$ and sememe embedding matrix $\mathbf{S}$ are pre-trained and fixed during the training process. Because sememes can be seen as words, we directly use their pre-trained word embeddings as sememe embeddings, and employ word2vec model to learn the embeddings. Specifically, We use Skip Gram model with embedding dimension $d_s = d_w = 300$, window size $K = 5$ and negative sampling number $NS = 5$. We keep the words with frequency over 50 and filter out the less frequent words in LIWC lexicon. For label embedding matrix $\mathbf{Y}$, we randomly initialize them and use backpropagation to update their value during training.

For a fair comparison, we use the same embeddings for all methods and use hold-out evaluation. For Top-down k-NN, we set $k = 5$. For Top-down SVM and structural SVM, we set regularization parameter $C = 1$ and convergence tolerance $tol = 0.01$. For CSSA, each sample is given 4 labels when prediction. In the Top-down approaches, each sample is given 1 label when choosing among child nodes.

For parameters in our model, we set a and d_y to be 300. When predicting word labels, we empirically set beam size of beam search to be 5 and $\delta = -1.6$. For Adam algorithm, we set the initial learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

When transforming the tree structured labels into label sequences, if a word has more than one path in the tree structure, we transform it into multiple label sequences. For example, if word x has labels $\{y_{i_1}, y_{i_2}, y_{i_3}, y_{i_4}\}$ and $\pi(y_{i_3}) = y_{i_2}$, $\pi(y_{i_4}) = y_{i_2}$, $\pi(y_{i_2}) = y_{i_1}$, we transform it into two sequences $\mathbf{y} = (y_{i_1}, y_{i_2}, y_{i_3})$ and $\mathbf{y}' = (y_{i_1}, y_{i_2}, y_{i_4})$. Hence, a word can match multiple sequences after the transformation.

Evaluation Metrics

We employ the widely-used Micro-F_1 and Macro-F_1 along with their Precision and Recall metrics to measure the performance of all the methods (Zhang and Zhou 2014).

- **Micro-$\mathbf{F_1}$** is a common metric used to evaluate classification algorithms and it gives equal weight to each example.

[1] https://pystruct.github.io/

	Overall		Level 1		Level 2		Level 3	
Model	Micro-F_1	W-M-F_1	Micro-F_1	W-M-F_1	Micro-F_1	W-M-F_1	Micro-F_1	W-M-F_1
TD k-NN	0.6198	0.6169	0.6756	0.6772	0.5716	0.5646	0.4884	0.4858
TD SVM	0.6283	0.6106	0.6858	0.6785	0.5766	0.5557	0.4503	0.4142
Structural SVM	0.6444	0.6448	0.7011	0.7010	0.5919	0.5919	0.5725	0.5718
CSSA	0.6511	0.6319	0.6880	0.6864	0.6172	0.5914	0.4729	0.4322
HD	0.7023	0.7000	0.7495	0.7476	0.6658	0.6614	0.6113	0.6064
HDSA	**0.7224**	**0.7204**	**0.7636**	**0.7616**	**0.6927**	**0.6874**	**0.6270**	**0.6234**

Table 2: Micro-F_1 and W-M-F_1 results in all and each layer(s).

	Overall		Level 1		Level 2		Level 3	
Model	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
TD k-NN	0.7230	0.5494	0.7718	0.6069	0.6912	0.4945	0.4905	0.4846
TD SVM	0.7297	0.5422	0.7707	0.6161	0.6988	0.4822	0.5855	0.3308
Structural SVM	0.6607	0.6341	0.7193	0.6843	0.6059	0.5862	0.5788	0.5769
CSSA	0.6168	0.6910	0.6068	**0.7973**	0.6306	0.6062	0.5797	0.3692
HD	0.7216	0.6859	0.7767	0.7238	0.6711	0.6604	0.6051	0.6169
HDSA	**0.7473**	**0.7001**	**0.7976**	0.7311	**0.7052**	**0.6804**	**0.6354**	**0.6308**

Table 3: W-M averaging Precision and Recall results in all and each layer(s)

- **Macro-$\mathbf{F_1}$** is also a commonly used metric and it gives equal weight to each class label. However, this may cause instability because categories in LIWC are highly imbalanced with some categories containing more than 1,000 instances while some only have less than 40 instances. Therefore, we use the weighted Macro-F_1 (W-M-F_1) to evaluate the performance of models.

Experimental Results

Table 2 and Table 3 shows the results of all baselines and our model. We have the following observations (the improvements are statistically significant using t-test at 0.01 level in overall scores):

First, it is clear from the table that both the HD and HDSA outperform all other baselines in the overall performance. This demonstrates that it is reasonable and effective to transform hierarchical labels into label sequences and employ neural network for classification. In each layer, HD and HDSA also have advantages over other algorithms, except that CSSA has higher recall at level 1. However, Micro-F_1 and W-M-F_1 of CSSA are lower than HD and HDSA at every level. Therefore, HD and HDSA are still better than CSSA in LIWC lexicon expansion.

Second, the HDSA outperforms HD by approximately 2%, which indicates that incorporating sememe information into the decoder model is useful for LIWC lexicon expansion. It is mainly because sememes can represent different meanings in a word, and help our model to alleviate the polysemy and indistinctness problems. In other words, HDSA can expand a more comprehensive and precise LIWC lexicon with the aid of sememe information.

Third, comparing HDSA with conventional Top-Down approaches like TD k-NN and TD SVM, one can notice that while the difference in precision between them is only around 2% in Layer 1, it increases to approximately 5% or more in Layer 3. Intuitively, this demonstrates that HDSA is more capable of preventing the error from propagating through layers.

One may argue that precision is often more important than recall in lexicon expansion for the fact that accuracy is often more important than completion for a lexicon. This is favorable for our model since we can increase the threshold δ to get a more accurate lexicon. Furthermore, Micro-F_1 and W-M-F_1 scores also change when we adjust δ. Therefore, we illustrate the effect of δ with HDSA model in Figure 4.

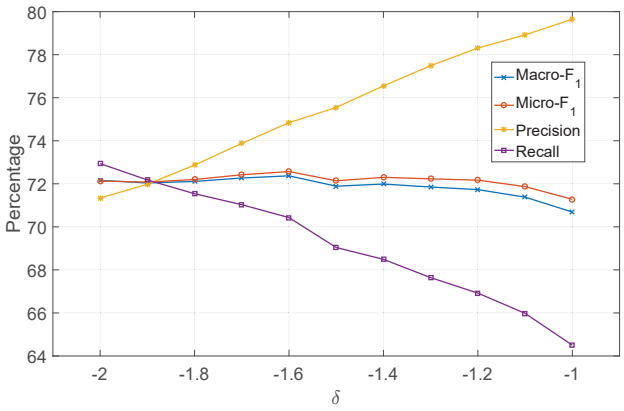

Figure 4: The effect of δ on Micro-F_1, W-M-F_1, W-M averaging Precision and Recall with HDSA model.

We can see from Figure 4 that δ has a direct impact on both precision and recall. As δ gradually increases from -2.0 to -1.0, precision also increases from 71.8% to 79.1% while recall decreases from 72.9% to 64.5%. This meets our expectation because higher δ means more strict standard and ruling out more labels than lower δ, resulting in higher precision and lower recall.

Unlike precision and recall, Micro-F_1 and W-M-F_1 do not change much as δ increases. The fluctuation range of Micro-F_1 and W-M-F_1 is about $\pm 1\%$. It is primarily because the

Word	Sememes	HD Prediction	HDSA Prediction	True Labels
恋人 (sweetheart)	交往 (associate), 人 (human), 爱恋 (love)	social←friend	social←friend, affect←posemo	social←friend, affect←posemo
今天 (today)	时间 (time), 现在 (present), 特定 (specific), 日 (day)	relativ←time	funct←TenseM←PresentM, relativ←time	funct←TenseM←PresentM, relativ←time
市镇 (town)	乡 (village), 市 (city), 地方 (place)	PersonalConcerns ←work	relativ←space	relativ←space
无望 (hopeless)	悲惨 (miserable)	cogmech←discrep	affect←negemo←sad	affect←negemo←sad
种种 (all kinds of)	多种 (various)	funct←negate	funct←quant	funct←quant
天空 (sky)	空域 (airspace)	relativ←time	relativ←space	relativ←space
联盟 (alliance)	结盟(ally), 团体(community)	PersonalConcerns ←work	social, PersonalConcerns←work	PersonalConcerns ←work
泪珠 (teardrop)	部件(part), 体液 (BodyFluid), 动物(AnimalHuman)	affect←negemo←sad	affect←negemo, bio←health	affect←negemo←sad

Table 4: Examples of words, sememes, HD prediction and HDSA prediction.

increase of precision and decrease of recall counteract each other and thus the fluctuation is small. This indicates that our model is robust against the choices of δ in a reasonable range.

Case Study

Table 4 shows some examples which HDSA predicts correctly with the favor of sememes while HD fails. It also shows some drawback where HDSA fails because of sememes. For each word, we show its relating sememes, the results of HD, the results of HDSA and the ground truth. Here we use $y_1 \leftarrow y_2$ to represent y_1 is the parent of y_2 for simplicity.

From the table, we have the following observations. First, words *sweetheart* and *today* have multiple labels hierarchically and the results of HD are only partially correct. This is mainly because of the polysemy of the words. On the contrary, with the help of sememes, such as `love` in *sweetheart* and `present` in *today*, HDSA successfully gives correct predictions, indicating that sememes are indeed useful for word polysemy.

Second, the results of HD in words *town* and *hopeless* are completely wrong. This could be caused by imprecise word embeddings. Instead, due to the extra information provided by sememes, such as `place` in *town* and `miserable` in *hopeless*, HDSA can predict them accurately.

Third, because of the indistinctness problem, HD may make mistakes when distinguishing categories. Words *all kinds of* and *sky* are two examples. On the other hand, since sememes can explicitly express meanings of words, such as `various` and `airspace`, HDSA gives errorless predictions.

Lastly, the result of HDSA in words *alliance* and *teardrop* are partially incorrect while HD gives correct predictions. This is mainly because sememes can sometimes be misleading, such as `community` in *alliance* and `BodyFuild` in *teardrop*. We will take the relations among sememes into consideration in the future work to better utilize sememe information.

We can conclude from the above observations that sememes are clearly useful for LIWC lexicon expansion and give HDSA improvements over the HD. Also, it indicates that HDSA is still not perfect and we will try to improve it in the future.

Conclusion and Future Works

In this paper, we utilize the Sequence-to-Sequence model for hierarchical classification of words to expand LIWC lexicon. To capture the exact meanings of words, we propose to use sememe information and utilize the attention mechanism to select appropriate senses of words when expanding the lexicon. In the experiments, we compare our model with other state-of-the-art methods and the results show that our model outperforms all of them. We also analyze several cases to show the effectiveness of sememes. The cases indicate that our model is more capable of understanding the meaning of words with the help of sememe attention.

In the future, we will explore the following directions:

- The sememes and words in HowNet have complicated structure and relations, and we currently ignore them in our model. We will explore to use this extra information for more precise and comprehensive LIWC lexicon.
- Currently our model needs pre-trained embeddings, and they are fixed during training. In the future, we will explore to update them during training or even not to use pre-trained embeddings.

Acknowledgments

This work is supported by the 973 Program (No. 2014CB340501), the Major Project of the National Social Science Foundation of China (No. 13&ZD190), the China Association for Science and Technology (2016QNRC001), and Tsinghua University Initiative Scientific Research Program (20151080406).

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barbedo, J. G. A., and Lopes, A. 2007. Automatic genre classification of musical signals. *EURASIP Journal on Applied Signal Processing*.

Bi, W., and Kwok, J. T. 2011. Multi-label classification on tree-and dag-structured hierarchies. In *Proceedings of the ICML*.

Bloomfield, L. 1926. A set of postulates for the science of language. *Language*.

Bravo-Marquez, F.; Frank, E.; Mohammad, S. M.; and Pfahringer, B. 2016. Determining word-emotion associations from tweets by multi-label classification.

Bravo-Marquez, F.; Frank, E.; and Pfahringer, B. 2015. Positive, negative, or neutral: Learning an expanded opinion lexicon from emoticon-annotated tweets. In *IJCAI*.

Bucci, W., and Maskit, B. 2005. Building a weighted dictionary for referential activity. *Computing attitude and affect in text*.

Bucci, W., and Maskit, B. 2007. Beneath the surface of the therapeutic interaction: The psychoanalytic method in modern dress. *Journal of the American Psychoanalytic Association*.

Cerri, R.; Barros, R. C.; and De Carvalho, A. C. 2014. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*.

Chen, Y.; Crawford, M. M.; and Ghosh, J. 2004. Integrating support vector machines in a hierarchical output space decomposition framework. In *IGARSS'04*.

Clare, A., and King, R. D. 2003. Predicting gene function in saccharomyces cerevisiae. *Bioinformatics*.

Coden, A.; Gruhl, D.; Lewis, N.; Mendes, P. N.; Nagarajan, M.; Ramakrishnan, C.; and Welch, S. 2014. Semantic lexicon expansion for concept-based aspect-aware sentiment analysis. In *Semantic Web Evaluation Challenge*.

da Silva Guimarães, N. R. P. 2016. Lexicon expansion system for domain and time oriented sentiment analysis.

Dong, Z., and Dong, Q. 2003. Hownet-a hybrid language and knowledge resource. In *Natural Language Processing and Knowledge Engineering, 2003. Proceedings*.

Fagni, T., and Sebastiani, F. 2007. On the selection of negative examples for hierarchical text categorization. In *Proceedings of the 3rd Language & Technology Conference (LTC07)*.

Fu, X.; Liu, G.; Guo, Y.; and Wang, Z. 2013. Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon. *Knowl.-Based Syst.*

Gao, R.; Hao, B.; Li, H.; Gao, Y.; and Zhu, T. 2013. Developing simplified chinese psychological linguistic analysis dictionary for microblog. In *International Conference on Brain and Health Informatics*.

Grimmer, J., and Stewart, B. M. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*.

Huang, C.-L.; Chung, C.; Hui, N.; Lin, Y.-C.; Seih, Y.-T.; Chen, W.; and Pennebaker, J. 2012. The development of the chinese linguistic inquiry and word count dictionary. *Chinese Journal of Psychology*.

Joachims, T.; Finley, T.; and Yu, C.-N. J. 2009. Cutting-plane training of structural svms. *Machine Learning*.

Kacewicz, E.; Pennebaker, J. W.; Davis, M.; Jeon, M.; and Graesser, A. C. 2014. Pronoun use reflects standings in social hierarchies. *Journal of Language and Social Psychology*.

Karn, S. K.; Waltinger, U.; and Schütze, H. 2017. End-to-end trainable attentive decoder for hierarchical entity classification. *EACL 2017*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiritchenko, S.; Matwin, S.; Nock, R.; and Famili, A. F. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Canadian Conference on AI*.

Lewis, M. P.; Simons, G. F.; Fennig, C. D.; et al. 2009. *Ethnologue: Languages of the world*.

Li, X., et al. 2008. Lexicon of common words in contemporary chinese.

Li, T. M.; Chau, M.; Yip, P. S.; and Wong, P. W. 2014. Temporal and computerized psycholinguistic analysis of the blog of a chinese adolescent suicide. *Crisis*.

Mehl, M. R.; Vazire, S.; Ramírez-Esparza, N.; Slatcher, R. B.; and Pennebaker, J. W. 2007. Are women really more talkative than men? *Science*.

Müller, A. C., and Behnke, S. 2014. pystruct - learning structured prediction in python. *Journal of Machine Learning Research*.

Newman, M. L.; Groom, C. J.; Handelman, L. D.; and Pennebaker, J. W. 2008. Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*.

Niu, Y.; Xie, R.; Liu, Z.; and Sun, M. 2017. Improved word representation learning with sememes. In *Proceedings of the ACL*.

Pennebaker, J. W.; Boyd, R. L.; Jordan, K.; and Blackburn, K. 2015. The development and psychometric properties of liwc2015. Technical report.

Pennebaker, J. W.; Booth, R. J.; and Francis, M. E. 2007. Linguistic inquiry and word count: Liwc [computer software]. *Austin, TX: liwc. net*.

Rohrbaugh, M. J.; Mehl, M. R.; Shoham, V.; Reilly, E. S.; and Ewy, G. A. 2008. Prognostic significance of spouse we talk in couples coping with heart failure. *Journal of consulting and clinical psychology*.

Schwartz, H. A.; Eichstaedt, J. C.; Kern, M. L.; Dziurzynski, L.; Ramones, S. M.; Agrawal, M.; Shah, A.; Kosinski, M.; Stillwell, D.; Seligman, M. E.; et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*.

Semeniuta, S.; Severyn, A.; and Barth, E. 2016. Recurrent dropout without memory loss. *arXiv preprint arXiv:1603.05118*.

Silla, C. N., and Freitas, A. A. 2009. Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In *SMC 2009*.

Silla Jr, C. N., and Freitas, A. A. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in NIPS*.

Yu, L.-C.; Lee, L.-H.; Hao, S.; Wang, J.; He, Y.; Hu, J.; Lai, K. R.; and Zhang, X.-j. 2016. Building chinese affective resources in valence-arousal dimensions. In *HLT-NAACL*.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *IEEE TKDE*.