SOM-NCSCM: An Efficient Neural Chinese Sentence Compression Model Enhanced with Self-Organizing Map

Kangli Zi[1,2], **Shi Wang**[1,*], **Yanan Cao**[3], **Yu Liu**[1,2], **Jicun Li**[1,2], **Cungen Cao**[1]
[1]Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zikangli19b, wangshi, liuyu19s, lijicun19s, cgcao}@ict.ac.cn
caoyanan@iie.ac.cn

Abstract

Sentence Compression (SC), which aims to shorten sentences while retaining important words that express the essential meanings, has been studied for many years in many languages, especially in English. However, improvements on Chinese SC task are still quite few due to several difficulties: scarce of parallel corpora, different segmentation granularity of Chinese sentences, and imperfect performance of syntactic analyses. Furthermore, entire neural Chinese SC models have been under-investigated so far. In this work, we construct an SC dataset of Chinese colloquial sentences from a real-life question answering system in the telecommunication domain, and then, we propose a *neural Chinese SC model* enhanced with a *Self-Organizing Map* (SOM-NCSCM), to gain a valuable insight from the data and improve the performance of the whole neural Chinese SC model in a valid manner.[1] Experimental results show that our SOM-NCSCM can significantly benefit from the deep investigation of similarity among data, and achieve a promising F1 score of 89.655 and BLEU4 score of 70.116, which also provides a baseline for further research on the Chinese SC task.

1 Introduction

Sentence Compression (SC) is an important natural language processing (NLP) task which aims to shorten sentences or texts while preserving their essential original meanings. The technique of SC can benefit several real applications, such as automatic title generation (Zhang et al., 2012; Wang et al., 2018), information extraction, opinion mining (Feng et al., 2010), machine translation (Li et al., 2020), and question answering systems.

Previous works of SC can be classified into two categories: (1) rule-based approaches (Vanetik et al., 2020), (2) machine-learning-based approaches (Filippova et al., 2015; Zhao et al., 2017). The latter can be further divided into statistic-based approaches (Knight and Marcu, 2002) and syntax-based approaches (Kamigaito and Okumura, 2020). Besides, most approaches have treated SC as a deletion-based processing that estimates the importance of each word in the sentences or texts in turn, and then decides if it should be kept or deleted in generating compressed sentences.

Although researches on SC have been conducted for many years in many languages, especially in English, improvements on Chinese SC task are still quite few as far as we know. One reason for this situation is lacking corpora of Chinese parallel data which are necessary for training and evaluating a supervised or semi-supervised Chinese SC model. To address this problem, several Chinese SC works tend to translate the English SC datasets into Chinese or use web crawler and data filtering techniques to produce high-quality data from the popular Chinese micro-blogging website or other Chinese news websites (Chen et al., 2009; Zhang et al., 2013; Hu et al., 2015). However, those datasets are mostly written in Chinese formal expressions, and not made publicly available or exactly extraction-based. Another reason is the particularity of the Chinese language itself, which not only brings about the former problem, but also causes the different segmentation granularity of Chinese sentences, without standard syntactic or grammatical instructions and vagueness of some Chinese expressions that make the judgement of compressed results non-identical but reasonable among different human evaluators.

To deal with the problems above, many Chinese SC approaches are more unsupervised-method-based using heuristic rules, which are hard to be duplicated and transferred to delete unnecessary constituents in sentences for other domains, or using statistical probabilities, like the TF-IDF, to trim

*Corresponding author.
[1]Our dataset and code will be available at: https://github.com/Zikangli/SOM-NCSCM.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 403–415
November 7–11, 2021.

the syntactic parse trees (Zhang et al., 2012, 2013; Wang et al., 2020), which lack flexibility and capacity, and demand for proper data distribution via calculating from the adopted dataset.

Since the domains of current Chinese SC datasets are limited, to better explore the advantages of entire utilization of neural network models for Chinese SC task, we (1) produce an SC dataset of Chinese colloquial sentences from a real-life question answering system in the telecommunication domain which is more natural than Chinese formal written expressions but more challenging to compress, and (2) augment a neural sequence labeling model with the combination of the output of a pre-trained neural clustering model based on a Self-Organizing Map (SOM) (Kohonen, 1982) using this labeled dataset, and several lexical features for a more effective Chinese SC method (*SOM-NCSCM*). We measure the performance of our models on the Chinese colloquial SC dataset using the F1 metric, BLEU scores (Papineni et al., 2002) and compression ratio (CR) (Napoles et al., 2011).

The contributions of this paper can be summarized as follows:

- To deal with the sparsity of Chinese SC parallel datasets, we create a Chinese colloquial SC corpora, which is the first Chinese parallel SC dataset in the telecommunication domain as far as we know. Evaluations during manually labeling and at the end demonstrate the high quality of the parallel data.
- On account of the source of data, and for better exploring the similarity among Chinese colloquial sentences, we propose a *SOM-enhanced neural Chinese SC model* (SOM-NCSCM) to gain a valuable insight from our data and improve the performance of the whole neural Chinese SC model in a simple but valid manner.
- We conduct extensive experiments to examine the effectiveness of our proposed SOM-NCSCM on our Chinese colloquial SC dataset. The results prove that our model can get promising performance.

2 Related Work

2.1 Sentence Compression (SC)

As mentioned in the introduction, the works of SC can be classified into two categories: the rule-based and machine-learning-based approaches. The rule-based approaches are mostly unsupervised, which dispense with large parallel corpora and generate compressed sentences using artificial rules (Zajic et al., 2007). The statistic-based approaches compress sentences according to calculated probabilities from large corpora, and they can also be supervised or unsupervised based on the types of training corpora (Knight and Marcu, 2002; Turner and Charniak, 2005; Malireddy et al., 2020). While the recent researches tend to operate on syntactic trees and reformulate the compression task as a tree pruning procedure (Clarke and Lapata, 2008; Filippova et al., 2015; Zhao et al., 2018; Kamigaito and Okumura, 2020).

However, the works above are mainly focusing on English SC tasks while researches on Chinese SC tasks are less common. Furthermore, most Chinese SC approaches are less commonly and fully adopting neural network models. Xu and Grishman (2009) enhanced linguistically-motivated heuristics by exploiting the event word significance and event information density via the TF-IDF weighting scheme, and then to solve Chinese news SC task. Similarly, Feng et al. (2010) applied a statistical score function for opinion-oriented Chinese SC. While Zhang et al. (2012) used Support Vector Machine (SVM) for syntax tree trimming of Chinese sentences to generate news titles from preprocessed WangYi news. Wang et al. (2020) proposed a Chinese SC algorithm based on the combination of heuristic rules and the emotional needs of different text scenes, and tested it on articles from Chinese news websites.

Moreover, according to the above researches on Chinese SC task, along with those on English SC task, two conclusions can be made. (1) The corpora of parallel training data are mostly produced from news articles in which the compressed sentences are the titles of those articles which are all in formal written language. That is far away from human daily expressions which are more casual and natural. While original Chinese parallel training corpora are relatively few, let alone those colloquial expressions in question answering situations could make those casual sentences become more complicated to compress. (2) Besides, efforts for seeking the effective ability of fully adopted neural network models for Chinese SC task have been barely done. Therefore, we will work on creating a Chinese parallel SC dataset from Chinese collo-

quial expressions in a real-life question answering system, and proposing an efficient method to deal with Chinese SC task in a neural-network manner.

2.2 Text Clustering

Text clustering is an application of cluster analysis to textual data for sample classification problems. As an unsupervised machine learning method, text clustering has a certain flexibility and high automatically processing capacity, for it does not require a training process with manually labeled categories of texts in advance.

There are many typical clustering algorithms, to name a few, such as K-means (MacQueen, 1967), Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) (Zhang et al., 1996), and Gaussian Mixture Model (GMM) (Rasmussen et al., 1999). These techniques have been applied to automatic summarization of documents, information retrieval, recommendation systems, and etc.

In this paper, we introduce a neural clustering method to enhance a neural Chinese SC model in dealing with Chinese colloquial sentences. The unsupervised clustering algorithm we utilize is SOM which fully uses Artificial Neural Network (ANN) framework (Kohonen, 1982). Besides, another reason for choosing the SOM as our clustering method is that it emphasizes mapping input data to the low-dimensional map while still preserving their topology. Moreover, it applies competitive learning in which the output neurons in the output computational layer need to compete amongst themselves to be activated and are selectively tuned to various classes of input data in the course of learning, which is different from those classic clustering algorithms and neural-network-based clustering methods that need to be provided with the number of clusters in advance, or use error-correction learning (such as the error back-propagation with gradient descent), or just generate word embeddings for those traditional clustering algorithms. As a result, only one output neuron could be activated at any one time.

3 Dataset Preparation

3.1 Data Collection and Annotation

The process of data collection and annotation is shown in Figure 1.

Data source. The data are collected from a real-life question answering system in the telecommunication domain[2]. This question answering system runs within China and most users are Chinese. The queries of users are composed of simplified Chinese natural language words and sometimes a few English words and numbers, such as "VIP", "WIFI" and "1GB".

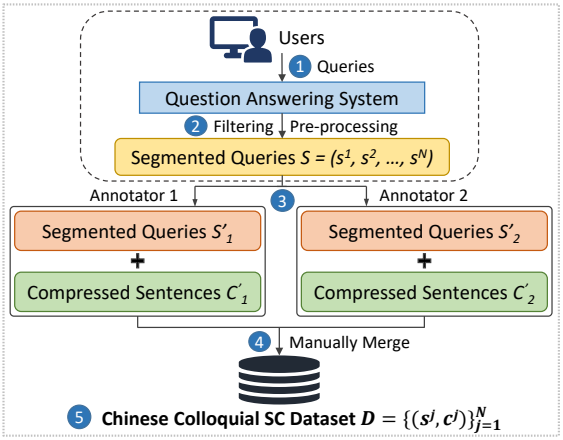

Figure 1: Diagram of the process of creating Chinese colloquial SC dataset.

Data collection. The real-life queries of different users are randomly collected from the question answering system. Besides, no user will be directly contacted, neither will their personal privacy information be stored in our dataset. So the data collected are not secrecy-related, nor do they involve users' personal privacy information. If there is little user information or domain-related sensitive information in the queries, we will filter out those queries or use tokens to mask the whole phrases express sensitive information in them. Appendix A.1 shows all special tokens we use in our dataset. In addition, our annotators will re-confirm all data in the process of data annotation[3]. It is worth noting that, we maintain the same segmentation results as the question answering system manages, for Chinese sentences should be segmented before downstream processing.

Data annotation. We ask two professional annotators to manually recheck the spelling of words, the granularity of segmentation, and then label the compressed sentences. These two annotators

[2]This is a collaborative project of our research group, and we've got and followed the consent of collecting and using the data.

[3]The data collection procedure fairly treats users' query sentences of all genders, ages, financial statuses, backgrounds, which means each query in the entire system has an equal probability of being chosen. This makes sure that our dataset is representative of the whole population.

are both expert annotators and recruited specifically for their understanding of telecommunication domain and having at least two-year work experience in handling and annotating various telecommunication-domain data from the question answering system. They're treated equally and provided with 500 sampled queries beforehand to be trained for the purpose of the Chinese SC task and to discuss with each other, before actually compressing those sentences. To ensure unbiased annotations, the comprehension of users' intent from both annotators and the question answering system's responses to queries are used as extra reference standards[4]. And the annotators should resume and modify mistakes over and over until there are no obvious conflicts between them. Finally, we obtain two sets of 3,300 compressed sentences from the two annotators, and further apply several quantitative metrics to measure those compressed results and merge them to produce the final compressed sentences.

3.2 Data Evaluation and Description

Data Evaluation. During the manually compressing process of each annotator, we set several automatic and quantitative evaluation metrics to assist their work: (1) a list of high-frequency words in the whole data, except the common stop words in Chinese; (2) a list of words left in real-time compressed sentences together with their 0/1 labels and the corresponding frequencies.[5] In the end, to assess the inter-annotator agreement, we apply the Cohen's Kappa coefficient (Cohen, 1960) and compute Cohen's unweighted *k*. The unweighted *k* gets 0.623, which reaches a substantial level (Landis and Koch, 1977). And to assess the ability to apply clustering algorithms on our dataset, we apply the Hopkins statistic (Banerjee and Dave, 2004) and consistently have a value of around 0.719∼0.726, which indicates our data has a high tendency to be clustered.

Data Description. We merge the manually-labeled compressed results of the two annotators to get a unified dataset, and the descriptions about this Chinese colloquial SC dataset are detailed in Table 1: (1) there are 3,300 queries left for annotation, after we automatically mask the personal information and filter out sentences with too many meaningless punctuations and those beyond responses from the system. The amount of the dataset is large enough to train and test a supervised SC model for Chinese SC task, especially by utilizing public pre-trained word embeddings and neural models; (2) the averaging CR is 0.709, which implies that the whole dataset consists of some short sentences that do not need to be compressed and are confusing and challenging for models to make deletion decisions.[5]

Total Number		3300 (original query, compressed sentence) pairs		
Compression Ratio (CR)		0.709		
Total Words	Origin: 2,552	Sentence Length	Origin	2 - 48 (characters)
				1 - 28 (words)
	Compression: 2,291		Compression	2 - 37 (characters)
				1 - 18 (words)

Table 1: Details of our Chinese colloquial SC dataset.

4 Models

In this section, we first provide the formalized definition of the SC task. Then, we describe the baseline model, a commonly-used neural sequence labeling model. Finally, we introduce our SOM-NCSCM to improve the performance and generality of the baseline model.

4.1 Task Definition

The formal definition of SC task is the same as that addressed by Filippova et al. (2015). That is, each original query sentence contains *n* length of word tokens $\boldsymbol{s} = (w_1, w_2, \ldots, w_n)$. Here, each $w_i \in \mathcal{V}$, where $\mathcal{V}$ is the vocabulary of our dataset. The SC task is to delete some of the words in $\boldsymbol{s}$ but remain the necessary words that express important information to produce a compressed sentence. Therefore, the corresponding compressed sentence contains *m* length of word tokens $\boldsymbol{c} = (w_1, w_2, \ldots, w_m)$ from the original and we can use a series of 0/1 labels $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ to denote the sequence of binary operations of the words in the original, where $y_i \in \{0, 1\}$. Here, $y_i = 0$ refers to a deletion operation of w_i, and $y_i = 1$ refers to a retention operation of w_i, so the total number of ones in the label sequence $\boldsymbol{y}$ is *m*.

Our Chinese colloquial SC dataset is denoted as $\mathcal{D} = \{(\boldsymbol{s}^{(j)}, \boldsymbol{c}^{(j)})\}_{j=1}^{N}$, and its corresponding deletion/retention label sequences denoted as $\mathcal{C} = \{(\boldsymbol{s}^{(j)}, \boldsymbol{y}^{(j)})\}_{j=1}^{N}$. Therefore, our Chinese SC goal is to learn a sequence labeling model using $\mathcal{C}$, so that for any Chinese query sentence $\boldsymbol{s}$, we

[4]The original query sentences and their corresponding compressed should get the same responses of the question answering system.

[5]Appendix A.2 shows some data details: the word frequency proportion of the 15 randomly-chosen words, and the number and proportion of different lengths of all segmented query sentences in our dataset.

can get its label sequence $\boldsymbol{y}$ and thus convert it into compressed sentence $\boldsymbol{c}$.

4.2 Baseline Model

As discussed earlier, there are few neural Chinese SC baseline models that can be readily trained on our Chinese colloquial SC dataset, so we choose a commonly-used neural sequence labeling model as our baseline model, which is a Bidirectional Long Short Term Memory (Bi-LSTM) network combined with a Conditional Random Field (CRF) layer, and its embedding layer is enhanced with various rich features, such as named entity (NE) and part-of-speech (POS).

Hence, for each query $\boldsymbol{s}$, the Bi-LSTM takes the joint embedding of its words and lexical features (the NE and POS features) as input $\mathbf{x} = (w_1, w_2, \ldots, w_{T_x})$ with $w_i \in R^{d_w+d_n+d_p}$, where T_x is the input length and d_w, d_n, d_p is the dimensionality of word embedding, NE embedding and POS embedding respectively. It then produces a sequence of hidden states $\boldsymbol{h} = (h_1, h_2, \ldots, h_{T_x})$ to represent its input, each of which is a concatenation of a forward and a backward LSTM representation:

$$\begin{aligned} h_i &= \left[\overrightarrow{h}_i; \overleftarrow{h}_i\right], \\ \overrightarrow{h}_i &= \text{LSTM}\left(w_i, \overrightarrow{h}_{i-1}\right), \\ \overleftarrow{h}_i &= \text{LSTM}\left(w_i, \overleftarrow{h}_{i+1}\right) \end{aligned} \tag{1}$$

where $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ are all d_h-dimensional vectors.

Then, instead of predicting label decisions independently, we pass the output of the Bi-LSTM to a CRF layer which can produce a probability distribution over the label sequence and give the best label sequence in all possible label sequences. Specifically, before passing the output of Bi-LSTM directly to the CRF, we add a dense layer and take its output as the input to the CRF layer:

$$\hat{\mathbf{h}} = \tanh\left(\mathbf{W}_d\mathbf{h} + \mathbf{b}_d\right) \tag{2}$$

where $\mathbf{W}_d$ and $\mathbf{b}_d$ are the weight and bias of the dense layer. The score of a corresponding label sequence $\boldsymbol{y}$ is computed as:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{T_x}\left(\mathbf{W}^{\top}_{c(y_{i-1},y_i)}\hat{\mathbf{h}} + \mathbf{b}_{c(y_{i-1},y_i)}\right) \tag{3}$$

and then, considering all possible label sequences, the probability of this label sequence $\boldsymbol{y}$ can be defined as:

$$p(\mathbf{y}\,|\,\hat{\mathbf{h}};\mathbf{W}_c,\mathbf{b}_c) = \frac{\prod^{T_x} e^{s(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{y}'\in\mathcal{Y}(\hat{\mathbf{h}})}\prod^{T_x} e^{s(\mathbf{x},\mathbf{y}')}} \tag{4}$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are the weight and bias corresponding to the given label pair (y_{i-1}, y_i) we are processing with.

During training, the objective of the whole model is to maximize the log-probability of the correct label sequence, and finally, we adopt the Viterbi algorithm for training the CRF layer and use the label sequence $\mathbf{y}^*$ with maximum score as the optimal label sequence:

$$\begin{aligned} L(\mathbf{W}_c, \mathbf{b}_c) &= \sum_{j=1}^{N}\log p\left(\mathbf{y}^{(j)}\,|\,\hat{\mathbf{h}}^{(j)};\mathbf{W}_c,\mathbf{b}_c\right) \\ \mathbf{y}^* &= \text{argmax}_{\mathbf{y}\in\mathcal{Y}(\hat{\mathbf{h}})}\, p(\mathbf{y}\,|\,\hat{\mathbf{h}};\mathbf{W}_c,\mathbf{b}_c) \end{aligned} \tag{5}$$

4.3 Our SOM-enhanced Neural Chinese SC Model

For minimizing manual labelling, we make no use of any syntactic feature, such as dependency parsing trees, and for better exploring the deeper implication of our data and relieving the degree of data sparsity, we decide to take the advantage of clustering methods and finally apply a pre-trained neural clustering model to improve our baseline neural Chinese SC model.

Specifically, in our model, we set up a SOM as our clustering method and build it to the kind that consists of a feed-forward structure with a single output computational feature layer where each neuron is fully connected to all the input nodes in the input layer. The architecture of our model is shown in Figure 2.

For training a SOM, we first discretize the word embeddings $\mathbf{e}^{\mathbf{w}} = \left(e^w_1, e^w_2, \ldots, e^w_{T_x}\right)$ from our original query sentences to a two-dimensional feature layer. Its output representation is defined as:

$$\mathbf{z}_s = \text{som}\left(\mathbf{e}^{\mathbf{w}};\theta_s\right) \tag{6}$$

where $som(\cdot)$ refers to the SOM processing, and θ_s denotes the trainable weights which are initialized with random values. The SOM processing will stop when the feature layer stops changing. Particularly, the normalized coordinates $\mathbf{z}_s$ of the activated neuron, representing a cluster of a query sentence, can then be converted to a mono-dimensional index.

Later, by applying the pre-trained SOM, we can assign a corresponding cluster index to each original query sentence. Then, we append an extra attention-based Bi-LSTM model (Bahdanau et al., 2015) for better enhancing the representation of a query sentence which takes the joint embedding of its words, lexical features (the NE and POS

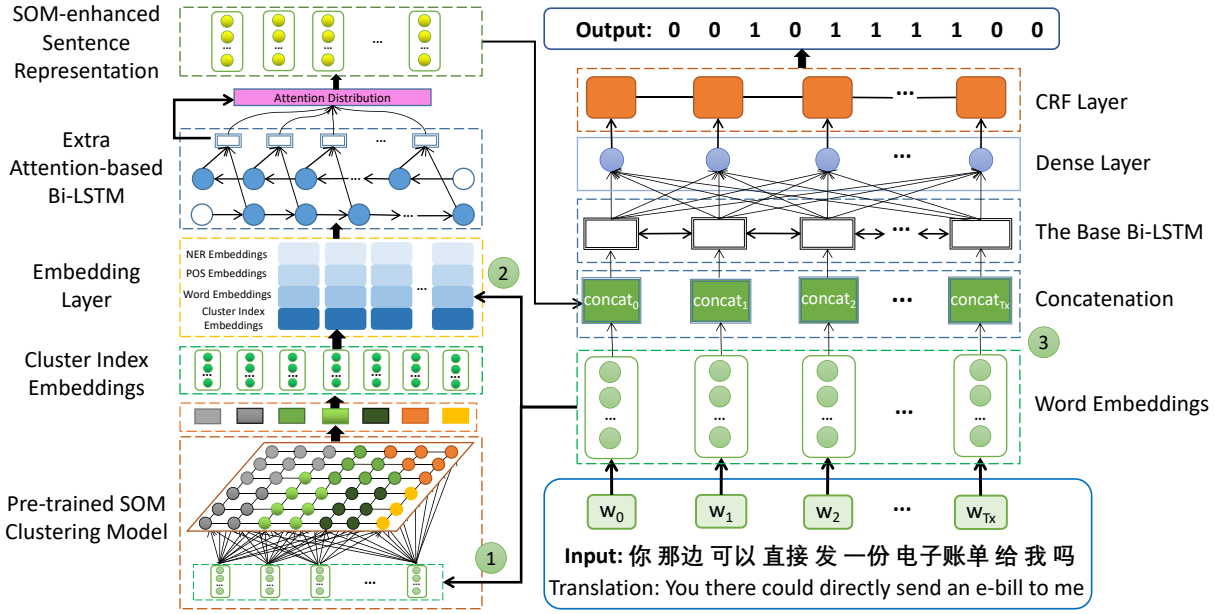

Figure 2: The framework of our SOM-NCSCM. The word embeddings are used three times as inputs: (i) in pre-training the SOM clustering model; (ii) in the extra attention-based Bi-LSTM model; (iii) in the base model.

features), and the randomly initialized cluster index feature as input $\mathbf{x}' = (w'_1, w'_2, \ldots, w'_{T_x})$ with $w'_i \in R^{d_w+d_n+d_p+d_c}$, where T_x, d_w, d_n, d_p are the same as the input to the baseline model mentioned above, and the d_c is the dimensionality of cluster index feature. The output representation of the current sentence is calculated as follows:

$$\begin{aligned} e^s_t &= \sum_i^{T_x} \alpha_{it} h'_i \\ \alpha_{it} &= \text{softmax}\left(u_{it}\right) \\ u_{it} &= v^\top \tanh\left(W_h^\top h'_i + W_s^\top h'_t\right) \end{aligned} \tag{7}$$

where $t \in [1, T_x]$, W_h, W_s, and v are all trainable parameters, the h'_i and h'_t are the outputs of the extra Bi-LSTM which are the concatenation of a forward and backward LSTM representation using the similar formula as Equation 1.

Finally, the input to the base model in our SOM-NCSCM is the concatenation of the word embeddings $\mathbf{e^w}$ and the sentence representation $\boldsymbol{e^s}$:

$$\hat{\mathbf{x}} = [\mathbf{e^w}; \boldsymbol{e^s}] \tag{8}$$

To avoid repetition, we won't describe the following compressing process which is similar to that of the baseline model again.

5 Experiments

5.1 Dataset and Experiment Settings

Dataset. We conduct experiments on our Chinese colloquial SC dataset. We shuffle the whole dataset and then split it into three parts, which produces 3,000, 150 and 150 samples for training set, development set and test set, respectively.

Implementation Details. In the experiment, we use the pre-trained Chinese word vectors with 300 dimensions to initialize the Chinese word embeddings (Li et al., 2018).[6] We use the Stanford CoreNLP to extract POS and NE features.[7] The MiniSom[8] is employed for constructing the neural clustering model, and we choose an 11×11 square map with a sigma of 4, an initial learning rate of 0.5, the Euclidean distance function to activate the map and the Gaussian function to weigh the neighborhood of nodes in the map. The representations of POS, NE and cluster index features are all randomly initialized as 32-dimensional vectors in the training stage. The depth of the LSTM layer is set to 2, while the hidden size of Bi-LSTM in both the baseline model and the extra attention-based model is 128, and the size of the dense layer is set to 64. Besides, to avoid overfitting, we use dropout before the Bi-LSTM layer and the dense layer with a dropout rate of 0.5. We set the batch size to 64 and use the optimization algorithm Adam (Kingma and Ba, 2015) with default parameters as an initial learning rate of 0.001. Our models are all trained

[6] We use the merge_sgns_bigram_char300.txt from https://github.com/Embedding/Chinese-Word-Vectors.

[7] https://github.com/stanfordnlp/CoreNLP

[8] https://github.com/JustGlowing/minisom

Models	F1	BLEU1	BLEU2	BLEU3	BLEU4	CR
Chinese BERT-based model	85.785	73.442	64.253	58.295	53.911	0.833
Baseline	88.466	82.996	76.179	71.210	67.449	0.690
w/ SOM direct	88.228	81.162	76.214	70.804	66.645	0.649
w/o NE&POS	87.872	81.250	75.155	68.773	64.124	0.712
w/o NE&POS w/ SOM direct	87.991	82.313	75.948	69.861	64.431	0.702
K-Means + NCSCM	89.061	**84.238**	77.995	72.299	67.333	0.672
GMM + NCSCM	88.704	82.893	**79.201**	73.763	68.489	0.673
SOM-NCSCM	**89.655**	84.000	78.221	**74.766**	**70.116**	0.683

Table 2: Main results on our Chinese colloquial SC dataset. Each corresponding model is repeated at least five times and we select the one with the middle value of metrics for comparison.

on a single GPU and the results are reported on the test set.

Evaluation Metrics. For automatically evaluating the performance of Chinese SC models and comparing them with other models in the future, we report the micro F1, BLEU scores and CR as the main evaluation metrics where the CR is computed as the number of left words in compressed sentences divided by the total number of words in the original query sentences.

Model Comparison. To evaluate our Chinese SC dataset, we propose the following models for comparisons.

(1) **Chinese BERT-based model.** We implement a char-level model using fine-tuned Chinese BERT-wwm model (Cui et al., 2020), and evaluate its performance on our dataset by replacing the pre-trained word embeddings in the baseline with the concatenation of BERT's outputs (without the NE and POS features). The output labels are based on the BIO scheme.

(2) **Baseline model.** This is the baseline model with NE and POS features as introduced in Section 4.2. We also evaluate its variants, such as those without NE and POS features, and those directly incorporating the cluster index features into the inputs.

(3) **NCSCM enhanced with classical clustering algorithms.** We replace the cluster index features obtained from the SOM with those from two classical clustering algorithms (K-Means and Gaussian Mixture), and evaluate their performances on our dataset, respectively. We set the parameter of the number of clusters to be 120, which is comparable to the output of the SOM model.

Our SOM-NCSCM					
SOM Size	9×9	10×10	11×11	12×12	13×13
F1	89.299	88.704	**89.655**	88.466	88.823
BLEU1	83.908	81.227	**84.000**	83.206	81.132
BLEU2	**79.156**	76.425	78.221	77.663	75.552
BLEU3	74.263	70.606	**74.766**	73.876	71.400
BLEU4	**70.547**	66.207	70.116	70.016	66.470
CR	0.688	0.649	0.683	0.680	0.710
Number of Clusters	81	100	121	144	169

Table 3: Main results of our SOM-NCSCM with different SOM sizes on our Chinese colloquial SC dataset.

(4) **Our SOM-NCSCM.** This is our proposed neural Chinese SC model as described in Section 4.3.

5.2 Main Results

Table 2 shows the main metric evaluation results of models on our test Chinese colloquial SC dataset and we have the following observations:

- Although the fine-tuned Chinese BERT-based model tends to retain more tokens/chars (its CR is relatively higher than the others), its F1 score doesn't reach a fascinating point. Moreover, during the training (fine-tuning) process, it's very easy to be over-fitting, due to the complex structure which consists of plenty of para-meters and requires large amount of dataset to fine-tune it.
- All the models employed with lexical features (NE and POS features) perform better than those without them. This verifies the effectiveness of the lexical features in expressing word functions and sentence meaning.
- With the performances of the baseline models which have been directly incorporated with cluster index features, compared to the one that only utilizes word embeddings as inputs,

Original Query Sentence	你/那边/可以/直接/发/一份/电子账单/给/我/吗	你/要/给/记录/[编号]/号/客服代表/挂/我电话/的/问题
Translation in English	You there could directly send an e-bill to me	You should record for (me) the matter that the No. [number] customer service representative hangs up on me
Gold Compressed Sentence	发/一份/电子账单/给/我 Send an e-bill to me	[编号]/号/客服代表/挂/我电话 The No. [number] customer service representative hangs up on me
Chinese BERT-based model	{边}/可以/直接/发/{份}/电子/账单/给/我 {There} could directly send {an} e-bill to me	记录/[编号]/号/客服/代表/挂/我/电话/问题 Record the matter that the No. [number] customer service representative hangs up on me
Baseline	你/那边/可以/直接/发/一份/电子账单/给 You there could directly send an e-bill to	[编号]/号/挂/我电话/问题 The matter that the No. [number] hangs up on me
K-Means + NCSCM	发/一份/电子账单/给 Send an e-bill to	挂/我电话/问题 The matter that hangs up on me
GMM + NCSCM	可以/发/一份/电子账单 Could send an e-bill	号/挂/我电话/问题 The matter that the No. hangs up on me
SOM-NCSCM	可以/发/一份/电子账单/给 Could send an e-bill to	[编号]/号/挂/我电话/问题 The matter that the No. [number] hangs up on me

Table 4: Some original query sentences along with the actual output compressed sentences from the compared models and our SOM-NCSCM. We provide aligned translations while additional words are being added in parentheses to form proper English sentences. The words in segmented Chinese sentences are separated with slashes and those exactly matching the Gold are in blue color, while words in brackets are masked personal information, words in red color are frequently omitted by those models, and characters in green braces are tokens retained by the fine-tuned Chinese BERT-based model.

it's clear that adding the cluster index features generally helps. But directly employing the cluster index features into the variants of baseline causes the performance to drop a little, compared to the standard baseline model. This makes sense because, while lexical features indicate each word in sentences, the cluster index features denote each sentence as a whole which could bring noise and cause sparsity problem as the amount of dataset is not so large to make a trade-off between fixing clustering mistakes and learning from cluster index features.

- Our NCSCM, accompanied by two classic clustering algorithms, can get attractive performance, which indicates the efficient and feasible structure of our Chinese SC model. Besides, our SOM-NCSCM, which utilizes the cluster index features from the SOM and other lexical features in an extra attention-based neural network to represent sentences, achieves the best F1 score of 89.655 and BLEU4 score of 70.116 among all baseline models. This result testifies the advantageous ability of the SOM and implies that the whole model can alleviate the effect of the shortage of parallel data while also make better use of similarity among data to solve SC task at the same time. Moreover, we will analyze several actual compressed sentences in the following section.

Model	Silhouette Coefficient Score	Calinski-Harabasz Score
K-Means	0.032	2.451
GMM	0.007	2.205
SOM-9	0.026	2.597
SOM-10	0.045	2.788
SOM-11	**0.071**	**4.066**
SOM-12	0.053	3.002*
SOM-13	0.035	2.838
Human	0.065*	2.945

Table 5: The Silhouette Coefficient scores and Calinski-Harabasz scores of all experimental models and human judgement on 50 randomly selected query sentences. (Note: the bold scores are the optimal scores among all scores, and scores with * are sub-optimal. The som-X means the SOM size = X×X.

5.3 Experiments on SOM Parameters

To better understand how different settings of SOM size in the neural clustering model impact the overall performance of our SOM-NCSCM, we conduct several ablation studies and those ablation results are shown in Table 3. With the growth of the size of SOM map, the number of clusters increases which may lead each cluster to be sparse so that it contains less than two query sentences. Besides, based on the outputs of SOM models in different SOM parameters, we manually track several original query sentences and analyze their corresponding clusters along with other original query sentences in the same clusters. Manual judgment criteria include the user demands, the sentence length, and the structure of original query sentences, such as those queries asking for e-bill, which should be less likely to be clustered with those asking about installation costs of TV.

In addition, we randomly selected 50 query sentences from our dataset, and calculate the Silhouette Coefficient scores (Rousseeuw, 1987) and Calinski-Harabasz scores (Caliński and Harabasz, 1974) to evaluate the cluster performance of all experimental models and the manual judgment.[9] Those scores are showed in Table 5. As a result, in the main experiments, we choose the SOM size = 11×11 whose outputs are more consistent with manual judgements and could get the best F1 score for illustration and comparison.[10]

5.4 Case Studies

Here, we provide two instances in our Chinese colloquial SC dataset jointly with the actual outputs from each model in Table 2 and analyze them to help guide further researches on this task (Table 4).

In both examples, we can see that, although the outputs of all the models tend to miss some words which form a grammatical structure of sentences, or keep too many optional words which don't contain key information of the sentences, they are acceptable and meaningful to some extent, in Chinese colloquial expressions. This indicates that there's some degree of gap between getting exact matched results with strict gold compressed sentences and producing acceptable outputs, and grammatical check could be added during the training or post-processing process. Besides, except for the situation of over-fitting, the fine-tuned Chinese BERT-based model keeps the most characters in its outputs where some characters (边 and 份), such as in the first example, are even incorrectly labeled with "I-1" while without the corresponding "B-1" that couldn't form the accurate and meaningful words (那边 – there and 一份 – an).

In the second example, including the baseline model, the outputs from it and our SOM-NCSCM are not "perfect", which mistakenly delete the keyword "客服代表 (customer service representative)" in the original query sentence. We further investigate the cause of this mistake and find that this word is a domain-specific proper noun which is an unknown word in the pre-trained word vectors dictionary and it even only occurs once in our dataset, let alone it's unrecognizable as a whole word phrase for the Stanford CoreNLP tool to label NE and POS tags. While in the meantime, the fine-tuned Chinese BERT-based is a char-level model and could hold the correct word tokens but in a different segmentation granularity. This phenomenon stimulates our interests in exploring how to better use the domain-specific knowledge and other neural-network techniques (e.g., copy mechanism) in our model to improve the quality of compressed sentences in the future.

[9] We use the scikit-learn toolkit to calculate those scores.

[10] More detailed information and analyses can be found in Appendix A.3.

6 Conclusion

To sum up, we construct a Chinese SC dataset, composed of Chinese colloquial sentences, from a real-life question answering system to address a major problem for supervised Chinese SC models – the lack of parallel corpora. The dataset, as far as we know, is the first Chinese parallel SC dataset in the telecommunication domain. Then, we build several fundamental baselines and propose an efficient neural Chinese SC model for introducing the neural clustering technique (the SOM) to enhance the fully neural Chinese SC task, which achieves satisfactory performances on the dataset. Those results confirm the utilization of similarity among data could benefit the SC task. We believe our Chinese SC dataset and SOM-NCSCM could provide a public and diverse Chinese SC dataset and a fully neural-network-based efficient Chinese SC model, and we also plan to continue constructing larger Chinese colloquial SC dataset and explore other neural-network techniques and semi-supervised approaches to deal with the Chinese SC task.

Acknowledgements

This work is supported by Beijing NOVA Program (Cross-discipline, Z191100001119014), the National Key Research and Development Program of China (2017YFB1002300,2017YFC1700300), National Natural Science Foundation of China (61702234). We thank all anonymous reviewers for their valuable comments.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Amit Banerjee and Rajesh N Dave. 2004. Validating clusters using the hopkins statistic. In *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, volume 1, pages 149–153 vol.1.

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Jinguang Chen, Tingting He, Zhuoming Gui, and Fang Li. 2009. Probabilistic unsupervised Chinese sentence compression. In *2009 IEEE International Conference on Granular Computing*, pages 61–66.

J. Clarke and M. Lapata. 2008. Global inference for sentence compression: An Integer Linear Programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Shi Feng, Daling Wang, Ge Yu, Binyang Li, and Kam-Fai Wong. 2010. A Chinese sentence compression method for opinion mining. In *Information Retrieval Technology*, volume 6458 of *Lecture Notes in Computer Science*, pages 320–329. Springer Berlin Heidelberg.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with LSTMs. In *EMNLP*, pages 360–368.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale Chinese short text summarization dataset. In *EMNLP*, pages 1967–1972.

Hidetaka Kamigaito and Manabu Okumura. 2020. Syntactically look-ahead attention network for sentence compression. *arXiv:2002.01145*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Teuvo Kohonen. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143. Association for Computational Linguistics.

Zuchao Li, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2020. Explicit sentence compression for neural machine translation. In *AAAI*, volume 34, pages 8311–8318.

J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, pages 281–297.

Chanakya Malireddy, Tirth Maniar, and Manish Shrivastava. 2020. SCAR: Sentence compression using autoencoders for reconstruction. In *ACL: Student Research Workshop*, pages 88–94.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Carl Edward Rasmussen et al. 1999. The infinite gaussian mixture model. In *NIPS*, volume 12, pages 554–560.

Peter J. Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *ACL*, pages 290–297.

Natalia Vanetik, Marina Litvak, Elena Churkin, and Mark Last. 2020. An unsupervised constrained optimization approach to compressive summarization. *Information Sciences*, 509:22 – 35.

Jingang Wang, Junfeng Tian, Long Qiu, Sheng Li, Jun Lang, Luo Si, and Man Lan. 2018. A multi-task learning approach for improving product title compression with user search log data. *AAAI*, 32(1).

ZhongXian Wang, XiangHui He, and XingYan Hu. 2020. Chinese sentence compression algorithm based on deep analysis of sentence hierarchy in multiple application scenarios. In *AEMCSE*, pages 61–66.

Wei Xu and Ralph Grishman. 2009. A parse-and-trim approach with information significance for Chinese sentence compression. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation - UCNLG+Sum '09*, pages 48–55.

David Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, 43(6):1549–1570.

Chunliang Zhang, Minghan Hu, Tong Xiao, Xue Jiang, Lixin Shi, and Jingbo Zhu. 2013. Chinese sentence compression: Corpus and evaluation. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 257–267. Springer Berlin Heidelberg.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: An efficient data clustering method for very large databases. In *ACM SIGMOD International Conference on Management of Data*, volume 1, page 103–114.

Yonglei Zhang, Cheng Peng, and Hongling Wang. 2012. Research on chinese sentence compression for the title generation. In *Workshop on Chinese Lexical Semantics*, pages 22–31. Springer, Berlin, Heidelberg.

Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. 2018. A language model based evaluator for sentence compression. In *ACL (Volume 2: Short Papers)*, pages 170–175.

Yang Zhao, Hajime Senuma, Xiaoyu Shen, and Akiko Aizawa. 2017. Gated neural network for sentence compression using linguistic knowledge. In *Natural Language Processing and Information Systems*, volume 10260, pages 480–491. Springer International Publishing, Cham.

A Appendices

A.1 Description of Special Tokens

In order to protect users' personal information and guard against breaches of the sensitive information in the telecommunication domain, we use five tokens to mask such information. And if such information in queries contains more than one segmented Chinese word, we will still use one token to mask that whole piece of word phrase. These tokens will not impede the use and comprehension of query sentences. Table 6 shows a detailed description of those special tokens, including examples in the form of paired original query sentences and manually compressed sentences.

A.2 Data Details

A.2.1 Word Frequency Proportion of the Randomly-chosen Words in the Dataset

After the manually annotating process, we randomly choose 15 words in our dataset, including telecommunication-related words and common stop words, and calculate their word frequency in the cases of original and compressed sentences. Figure 3 shows the heat map of word frequency proportion of those words in our dataset. (1) In the original query sentences, the focus on domain-related words are distracted by those stop words, for users usually use colloquial expressions and polite words in the real-life question answering system, which means their queries are not concise enough. (2) After compression, those stop words without semantic meanings in the original sentences and those that won't affect the syntactic structure of the compressed sentences, will be deleted. As a result, the compressed sentences will be shorter and more succinct than the originals that express the demands of users immediately, which also achieves the goal of SC task to dispense important information and give that to the downstream tasks more effectively.

A.2.2 Distribution of Sentence Lengths of Our Dataset

All query sentences of real-life users are randomly chosen from the question answering system, and we calculate the number and proportion of different lengths of all 3,300 segmented query sentences in our dataset, in the cases of original and compressed parts. Figure 4 shows the distribution of sentence lengths of our dataset. In the original part, nearly 85% of the sentences contains 4 – 11 words, while in the compressed part, they can be shortened into 3 – 8 words. There are too short sentences (e.g. containing fewer than 4 words) in the original part, which make the exception of SC task that may not be shortened anymore, while those long sentences (e.g. containing more than 15 words) will be obviously tightened. Containing different lengths of original sentences can ensure the diversity of our data and challenge the ability of SC models.

A.3 Cluster Results of the Randomly-selected Query Sentences in the Dataset

To analyse the performance of the clustering algorithms with different parameters in a clearer way, we randomly select 50 query sentences from our dataset, and use the GMM, K-means and pre-trained SOM models in different SOM sizes to produce the corresponding cluster results. Those cluster results are shown in Figure 5.[11] Several observations can be made as follows.

(1) Look directly at the data distribution, and we can see there is a tendency to cluster those

[11] We use the Principal Component Analysis (PCA) implemented in the scikit-learn toolkit to reduce each data point into two-dimensional coordinates.

Token	Description	Example		Number (Percentage)
业务名词 Domain-related noun	This token masks those word phrases express telecommunication-related information which are sensitive to be made public.	OQS	[业务名词]/有/推出/什么/惠民/政策/吗 Do [domain-related noun] introduce any policies to benefit the people	455/3300 (13.636%)
		GCS	[业务名词]/推出/什么/惠民/政策 [Domain-related noun] introduces any policies to benefit the people	
名词 Noun	This token masks those word phrases express names of cellphone applications or products which are not related to telecommunication domain.	OQS	这个卡/我/可以/开/[名词]/视频/的/会员/吗 (Using) this phone number, can I open a VIP account of [noun] Video	64/3300 (1.939%)
		GCS	这个卡/可以/开/[名词]/视频/会员 (Using) this phone number can open a VIP account of [noun] Video	
电话号码 Phone number	This token masks specific service phone numbers or presonal cellpohone numbers when users already directly input them into query sentences.	OQS	[电话号码]/打/过/太/复杂/没有/人/接 (This) [phone number] was called, (it's) so complicated, nobody answered	Service phone number: 18/3300 (0.545%) Personal cellphone number: 4/3300 (0.121%)
		GCS	[电话号码]/打/过/没有/人/接 (This) [phone number] was called, nobody answered	
编号 Number	This token masks products' or servers' serial numbers, or other information or materials' codes (e.g. application error code).	OQS	电视/[编号]/是/什么/障碍 The television (shows) [number], what's the trouble	14/3300(0.424%)
		GCS	电视/[编号]/障碍 The [number] trouble of television	
地址 Address	This token masks information of Chinese provinces or addresses in query sentences.	OQS	安装费/是/多少/我/在/[地址] How much do installation cost, I'm in [address]	69/3300 (2.091%)
		GCS	安装费/多少/[地址] How much do installation cost in [address]	

Table 6: Description of special tokens used in our dataset. We provide aligned translations in examples while additional words are being added in parentheses to form proper English sentences. (OQS represents the Original Query Sentence; GCS represents the Gold Compressed Sentence.)

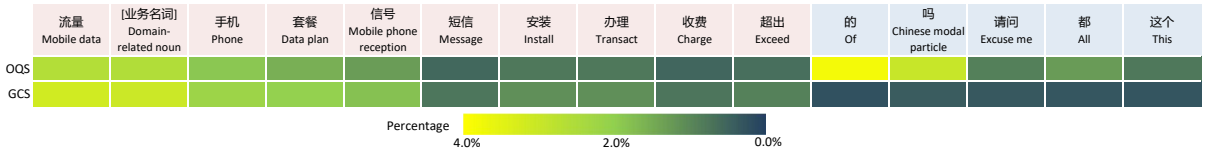

Figure 3: The word frequency proportion of the 15 randomly-chosen words in our dataset. Words in pink background are telecommunication-related domain words, and words in blue background are stop words. We provide aligned translations in English or descriptions of words' usage of these 15 words in Chinese. (OQS represents the Original Query Sentence; GCS represents the Gold Compressed Sentence.)

sentences, and the cluster results of manual judgement also prove this tendency.

(2) The GMM and K-Means have more than twice as many clusters as the SOM model of size = 11×11 and the manual judgement do. They are sort of randomly assigning clusters to those 50 query sentences.

(3) Each SOM model can obtain good cluster results of those sentences, where the cluster results of the SOM size = 11×11 are much closer to those of manual judgement, and the separation of each cluster in its results is clearer than that of other results in SOM models of different sizes.

(4) There are some intersections and correlation among the data, such as query sentences located in the lower left and right corner of the sub-figure respectively. When we analyse those query sentences in detail, we notice that there are several query sentences without clear user demands or specific domain-related words, which are difficult to classify them. And there are also some query sentences manually classified into more fine-grained clusters than they are in other cluster algorithms. For better classifying them, the telecommunication-related knowledge should be involved and the specific situation in which the query sentences are spoken should be considered. These kinds of information are important in dealing with real-life semantic analysis. We will continue our studies on the Chinese SC task in the future.

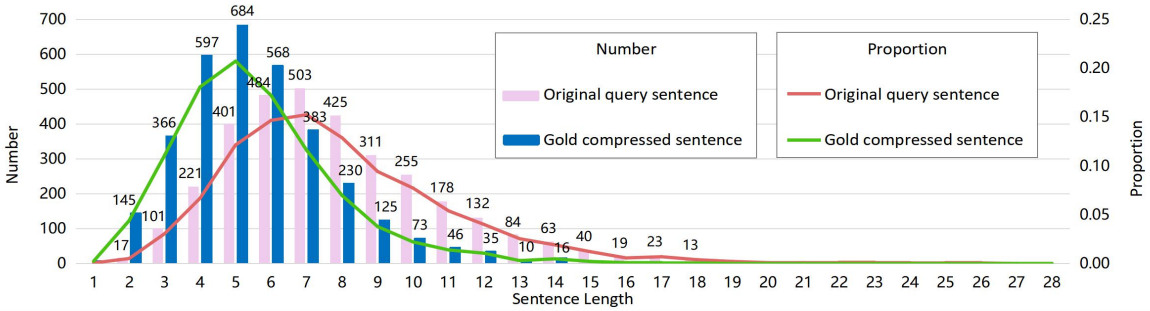

Figure 4: The number and proportion of different lengths of all segmented query sentences in our dataset.

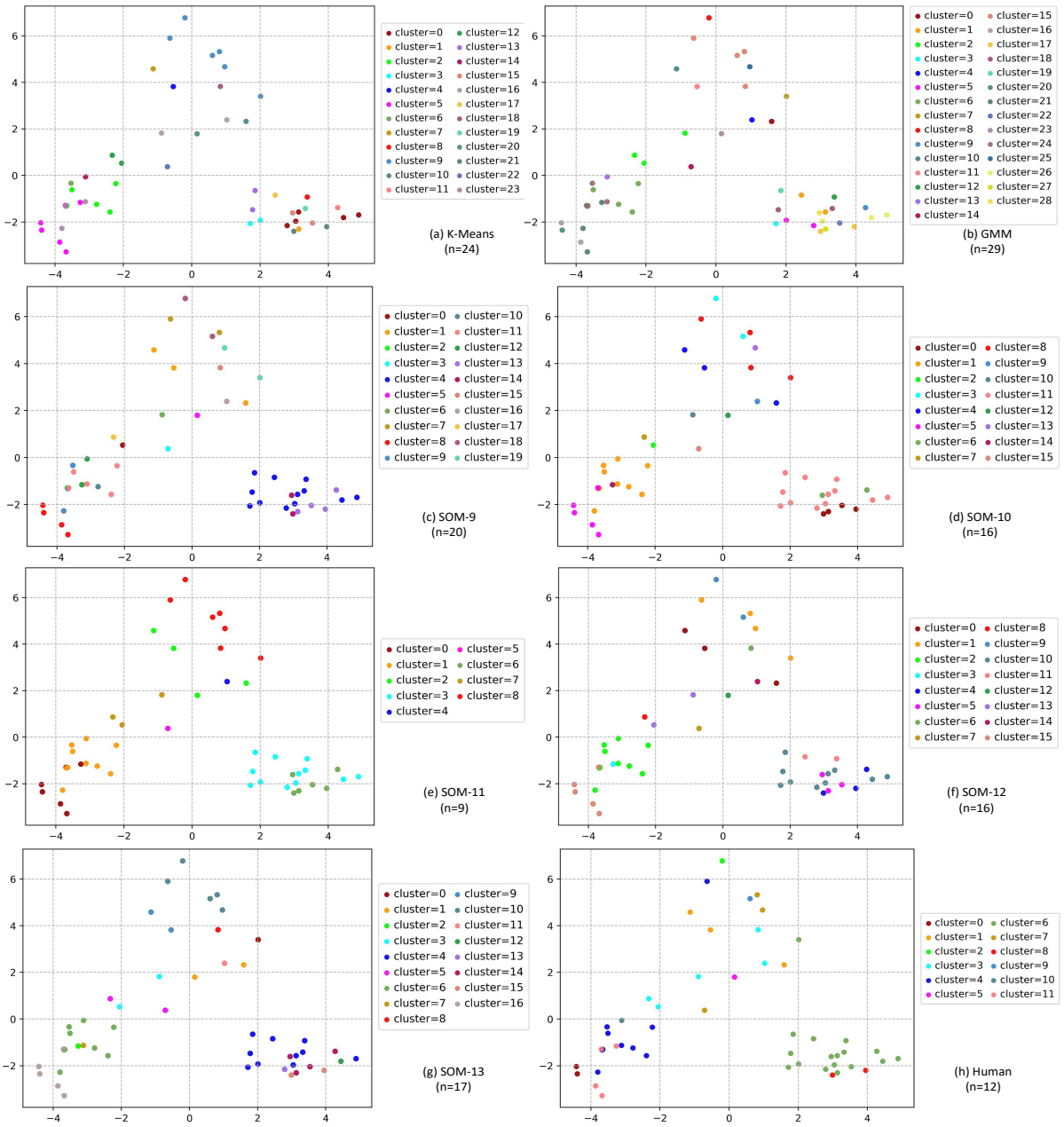

Figure 5: The cluster results of the 50 randomly selected query sentences in our dataset. In the title of sub-figures, the n=Y in the parentheses means that the total number of clusters is Y, and the som-X means the SOM size = X×X. (Note: the cluster indexes in each figure are serial numbers unrelated to others.