Completion and Parsing Chinese Sentences Using Cogent Confabulation

Zhe Li, Qinru Qiu
Dept. of Electrical Engineering & Computer Science
Syracuse University
Syracuse, NY 13244, USA
{zli89, qiqiu}@syr.edu

***Abstract*—Among different languages' sentence completion and parsing, Chinese is of great difficulty. Chinese words are not naturally separated by delimiters, which imposes extra challenge. Cogent confabulation based sentence completion has been proposed for English. It fills in missing words in an English sentence while maintains the semantic and syntactic consistency. In this work, we improve the cogent confabulation model and apply it to sentence completion in Chinese. Incorporating trained knowledge in parts-of-speech tagging and Chinese word compound segmentation, the model does not only fill missing words in a sentence but also performs linguistic analysis of the sentence with a high accuracy. We further investigate the optimization of the model and trade-offs between accuracy and training/recall complexity. Experimental results show that the optimized model improves recall accuracy by 9% and reduces training and recall time by 18.6% and 53.7% respectively.**

Keywords—Chinese sentence completion; parts-of-speech tagging; word segmentation; mutual information; cogent confabulation

I. INTRODUCTION

As an important part of text recognition, sentence completion and prediction, which stands for the capability of filling missing words in an incomplete sentence, has attracted much attention. The first step of sentence completion is syntactic parsing of the input text. Among different languages, Chinese is a great challenge due to its linguistically isolating. Each Chinese character generally corresponds to exactly one morpheme and multiple semantic meanings. Moreover, there has been a strong tendency in the Chinese language family over the last 2000 years for single morpheme words to develop into compounds of two or more morphemes [8], which makes Chinese language linguistically more flexible and complex. All of the above makes Chinese sentence completion extremely difficult.

In our previous research [1] [2] [3] [4], a cogent confabulation based sentence completion framework is developed. A sentence is represented by a set of lexicons corresponding to its words, word pairs, and part-of-speech tags. The conditional probability between neighboring lexicons are learned from training corpus. During recall, the missing information (including unknown word and part-of-speech tags for both unknown and given words) is selected that maximizes the likelihood of observed information (i.e. those words already given in the input sentence). Due to the difference between linguistic structures, this framework has to be modified for Chinese sentences. First of all, each Chinese character, which is represented as a 3-byte UTF-8 code, is analogy to an English word. In the rest of the paper, we use character and word interchangeably, as they are the same in Chinese. Secondly, the part-of-speech tagging of Chinese is usually associated with each multi-character compound. Correct segmentation is essential to syntactic parsing of the sentence.

We improve previous cogent confabulation model and apply it to Chinese sentence completion. Besides integrating *parts-of-speech (POS)* tagging that identifies the function of each word, in the Chinese sentence confabulation, segmentation label for multi-character compound is added, which identifies word compound consisting of 1~4 Chinese characters.

This work focuses on developing, optimizing and evaluating a confabulation model for Chinese sentence completion with high accuracy. It has three major contributions:

1) We extend the original sentence confabulation model to consider linguistic properties of Chinese language. Segmentation labels and beginning of sentence markers are specifically added to the model. *Knowledge link*s *(KL)* are shared to reduce complexity and improve performance as well. Experiment results shows that the extended Chinese sentence confabulation model achieve 76.9% sentence recall accuracy with reduced memory and computing complexity.

2) We analyze the mutual information between source and target lexicons of each knowledge link in the confabulation model and assign weight to these knowledge links accordingly. Compared to the original model, the model with weighted knowledge link has 9% higher recall accuracy.

3) The mutual information of KLs is also exploited to find the best training set size, which gives the best tradeoffs between training effort and recall accuracy.

The rest of this paper is organized as follows: A brief introduction of background is provided in Section II. In section III, the modeling and operation of Chinese sentence confabulation is introduced. The comparison of different configuration models of Chinese sentence confabulation and experimental results are presented in Section IV. Section V gives the conclusions.

This work is partially supported by the National Science Foundation under Grants CCF-1337300, and Air Force Research Laboratory under contract FA8750-11-1-0266

978-1-4799-4549-8/14/$31.00 ©2014 IEEE

II. BACKGROUND

A. Cogent confabulation

Inspired by human cognitive process, cogent confabulation [7] mimics human information processing including Hebbian learning, correlation of conceptual symbols and recall action of brain. The model describes the target using a set of orthogonal features that are mapped to a set of lexicons. The observed value of the feature is considered as a random variable in a discrete space represented by a set of symbols. The operation involves two important steps: training and recall. In training process, posterior probabilities between observations of two lexicons are collected and referred as the knowledge links. The collection of all knowledge links in the model forms its *knowledge base* (*KB*).

If we consider symbols as neurons and lexicons as categories, then the model consists neurons (i.e. symbols) in different categories (i.e. lexicons). When a symbol is observed, then the corresponding neuron fires. The neurons belonging to the same lexicon are connected via inhibitory synapses, as they are exclusive to each other, while those belonging to different lexicons can be connected via excitatory synapses. The strength of an excitatory synapse from sender (s) to receptor (t) is defined as $ln[P(s|t)/p_0]$ where $P(s|t)$ is the probability that s is observed given the condition that t is observed, and p0 is a small constant to make the result positive. This definition agrees with the Hebbian theory, which specifies that the synaptic strength increases when two neurons constantly firing together.

The input of the recall process is a noisy observation of the target. In this observation, certain features are observed with great ambiguity, therefore multiple symbols are assigned to the corresponding lexicons. The goal of the recall process is to resolve the ambiguity and select the set of symbols for maximum likelihood using the statistical information obtained during the training process. This is achieved using a procedure similar to the integrate-and-fire mechanism in biological neural system. Each neuron receives an excitation, which is the weighted sum of its incoming excitatory synapses. Among neurons in the same lexicon, those that are least excited will be suppressed and the rest will fire and become excitatory input of other neurons. Their firing strengths are normalized and proportional to their excitation levels. As neurons gradually being suppressed, eventually only the neuron that has the highest excitation remains firing in each lexicon and the ambiguity is thus resolved.

A computational model for cogent confabulation is proposed in [7]. Let a set of symbols associated to lexicon A is denoted as S_A. A KL from lexicon A to B is a $M \times N$ matrix where *source lexicon A* has M symbols and *target lexicon B* has N symbol. Each row in KL represents a *source symbol* in A and each column represents a *target symbol* in B. The (i, j)th entry of the matrix represents the strength of the synapse, called *contribution*, between the source symbol s_i and the target symbol t_j. It is quantified as the conditional probability $P(s_i|t_j)$, where s_i is the ith symbol in lexicon A and t_j is the jth symbol in lexicon B. During recall, the excitation levels of all ambiguous symbols are evaluated. Let l denote a lexicon, F_l denote the set of lexicons that have knowledge links going into lexicon l, and S_l denote the set of symbols that belong to lexicon l. The excitation level of a symbol t in lexicon l can be calculated as

$$I(t) = \sum_{k \in F_l} \sum_{s \in S_k} I(s) \left[\ln \left(\frac{P(S|t)}{p_0} \right) + B \right], t \in S_l \quad (1)$$

The function $I(s)$ is the excitation level of the source symbol s. The parameter p_0 is the smallest meaningful value of $P(s_i|t_j)$. The parameter B is a positive global constant called the *bandgap*. The purpose of introducing B in the function is to ensure that a symbol receiving N active knowledge links will always have a higher excitation level than a symbol receiving $(N-1)$ active knowledge links, regardless of the strength of the knowledge links.

B. Processing the training text

Our training text is segmented and tagged using Standford Part-of-speech (POS) tagger [5]. It is one of the most matured Natural Language Processing software based on probabilistic tagging systems. First, the Chinese training sentences are segmented using Stanford Chinese word Segmenter, which is based on a linear-chain *conditional random field (CRF)* model. The tool partitions sentence into compound words consisting of single or multiple Chinese characters. And then Stanford POS Tagger takes segmented sentence as input and assigns a part-of-speech tag to each compound. Stanford POS Tagger for Chinese Language exploits 33 word level Chinese tags specified by the Penn Treebank Tagging System [6]. TABLE I. lists some examples of these Tags.

The information of POS tags and segments will be built into the knowledge base during training. However, the POS tagger cannot be used to process sentences with missing words. Therefore during recall, we cannot use POS tagger for syntactic analysis. Our solution is to rely on the confabulation model to recall the segments and tags during the same time when the missing words are filled in. The basic idea is to assume that all tags and segment partitions are possible at the beginning, and gradually eliminate the ambiguity during the recall process. This approach is feasible since the number of tags and possible segment partitions is limited. Our experimental results show that considering tags and segmentations at the same time helps to improve the accuracy of sentence completion.

TABLE I. PENN TREEBANK TAG LIST

Tag	Function	Examples
VA	Predicative Adjective	很(very),雪白(snow white)…
VC	Copula	是(be),为(be),不是(not be)…
VE	有(have) as the main verb	有(have), 没有(have),无(not have)…
VV	Other verb	想(want to),走(walk),喜欢(like)…
NR	Proper Nouns(location, newspaper…)	北京(Beijing),纽约时报(New York Times)…
NT	Temporal Nouns	一月(Janurary),汉朝(Han Dynasty)…
NN	All other Nouns	书(book),房子(house)…
PN	Pronoun	我(I),你(you),这(this)…
…	…	…

III. Chinese Sentence Confabulation

The Chinese sentence confabulation is an extension of the framework proposed in [3], which is designed for English text.

A. Basic confabulation framework

Inheriting from original sentence confabulation framework [3], we assume that the maximum length of a sentence is 20 words and sentence with more than 20 words will be truncated. We pad the sentence that has less than 20 words with special character ‘’ to represent the end of a sentence. Anything beyond the end of sentence will be ignored during training and recall.

Original Sentence confabulation framework has two levels of lexicons – word and word pair. Lexicons 0 to 19 correspond to single English word at location 0 to 19 in a sentence. Lexicons 20 to 38 correspond to 19 word pairs combining word from lexicon 0~19 and its right adjacent neighbor. Each lexicon stores tremendous number of symbols (words or word pairs) that appears in the corresponding location.

In original framework, a KL is created between any two lexicons. In training process, we build all KL matrices to form knowledge base. And during recall, observed symbols will be set active in each lexicon. When there is no ambiguity in observation, only one symbol in a lexicon will be set active. Multiple symbols in the same lexicon will be set active as a result of ambiguous observation. They are referred as *candidates*. When a lexicon is not observable, all possible symbols will be set active to indicate the highest ambiguity. The excitation level of each candidate in the lexicon with ambiguity will be calculated and the symbols that is least excited will be suppressed. This procedure repeats until there is only one symbol left in each lexicon.

B. Chinese Sentence confabulation model

Each Chinese character is encoded using 3 bytes of UTF-8 code. As mentioned before, we regard each Chinese character as a “word” and they occupy the word level lexicons in the confabulation framework. Modern Chinese language is based on word compound, which consists of 1~4 single Chinese characters. These word compounds are not delimited, however, they can be found with the help of tools, such as the Stanford POS tagger. We label each Chinese character based on its position in a word compound, and refer this as *segmentation label*. For example, in a two character word compound 书籍 (book), 书(book) is located at the first position of the two character word compound, therefore, it is marked as 1IN2, and 籍(book) is marked as 2IN2. In this work, ten segmentation labels are used. They are: 1IN1, 1IN2, 2IN2, 1IN3, 2IN3, 3IN3, 1IN4, 2IN4, 3IN4, and 4IN4. Please note that segmentation label is only needed in Chinese sentence confabulation. This is a major difference between Chinese and western languages. In Section IV we will show the necessity of including segmentation label in the confabulation model.

In the improved confabulation model, new lexicons are created for tags and segmentation labels. Moreover, instead of having lexicons for two adjacent words, we create lexicons for three adjacent words in order to adapt to semantic compounds of multiple Chinese characters. Therefore, lexicons in the new confabulation model can be divided into four *levels*: lexicons 0~19 correspond to single Chinese word; lexicons 20~37 correspond to Chinese word triplets; lexicons 38~57 correspond to POS tags and lexicons 58~77 correspond to segmentation labels.

The original sentence confabulation framework has a knowledge link between any two lexicons. Therefore, the size of knowledge base increases exponentially with the number of lexicons. In this way, $78 \times 77 = 6006$ KLs will be generated for the Chinese sentence confabulation model, which takes tremendous resources.

To reduce the complexity of our computational model, two actions are jointly taken. First is to share KL matrix between lexicons that have the same relative position in sentence. For example, the distance from lexicon 0 to lexicon 1 is the same as the distance from lexicon 1 to lexicon 2, so the KLs between 0~1 and 1~2 are merged and shared.

The second action is to only create KLs between lexicons within N-neighborhood in the same lexicon level or across lexicon levels. In [10], experimental results show that considering words with low correlation in speech recognition making the performance poor. Empirically, 5-neighborhood is a best trade-off for accuracy and complexity. Therefore, we only generate knowledge links between two lexicons whose horizontal distance is within -5 to 5. We refer to the new sentence confabulation model with these two changes as *circular model* as the knowledge links are circulated among lexicons.

Segmented and tagged training text is used during training. Characters, tags and segment labels are placed in corresponding lexicons. KLs are established not only between two lexicons in the same level, but also between lexicons in different levels, as long as their distance is less than 5. However, there is no KL between tag and segmentation label lexicons, because tags and segments are derivatives of the Chinese characters, and Stanford tools are not able to ensure 100% accuracy in tagging. Keeping KL between tag and segment lexicons will introduce noise in the confabulation procedure.

A test sentence with missing characters will be given during recall. For those lexicons that are partially observable, a set of candidates that compliant with the partial observation is activated. If a lexicon is completely unobservable, then all possible symbols are activated as potential candidates. Since the test sentence originally is provided without tags and segmentation labels, the confabulation model automatically activates all tags and segmentation labels as possible candidates for each tag and segmentation label lexicon respectively.

C. Training and Recall functions

Given the confabulation model, the training and recall procedures are developed.

The training process establishes knowledge base on tagged and segmented text. Taking following sentence “国王#NN (The king) 有#VE (has) 两#CD 个#M (two) 儿子#NN (sons)” as example, the corresponding tagged and segmented training text is as follows, and the confabulation model constructed based on the training text is given in Fig. 1.

Fig. 1. Lexion Structure of confabulation model

As shown in Fig. 1, a special symbol "^" is assigned to the first lexicon in each level. Those words that frequently appear at the beginning of a sentence will have strong link with this special symbol. The indication of beginning of sentence is especially important for circular model, because its knowledge base only contains relative position information. The beginning of sentence symbol acts as anchors that provide absolute position information.

We can also see from Fig. 1 that the sentence is extended to 20 characters that are symbols assigned to lexicons 0 to 19 respectively. Those 20 characters will generate 18 three-word triplets and be assigned to lexicons 20~37, 20 tags and 20 segmentation labels will enter lexicons 38~57 and lexicons 58~77 respectively. Knowledge links between lexicons will be established as explained in Section III.B. At the end of training, the system will calculate the symbol to symbol conditional probability to fill in the KL matrix entry. For example, P("国"|"王") will be stored as an entry in the KL connecting lexicons 1 and 2, and P("国"|"NN") will be stored as an entry in the KL connecting lexicons 1 and 39.

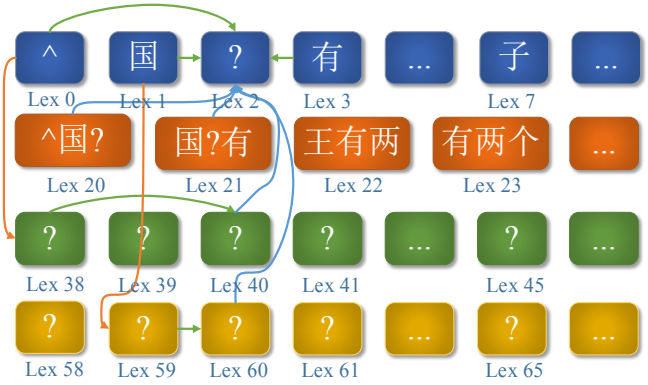

Fig. 2. Lexion Structure of confabulation model (Any arrow is from source lexicon to target lexicon. Orange arrows represents Knowledge Links from observable lexicons to unobservable or partially observable lexicons; Green arrows represents Knowledge Links between lexicons in same level; Blue arrows represents Knowledge Links from unobservable or partially observable lexicons to observable lexicons)

During recall, sentences with missing characters will be given. Taking the same sentence in Fig. 1 as example, Fig. 2 gives a simple explanation how the model works. Assume that the third character "王(king)" is partially observable, and the ambiguous observation gives two candidates: "王(king)" and "工(labor)". Symbols in lexicons are activated according to the observation. Hence lexicon 2 has two symbols "王(king)", "工(labor)" activated. And since no tags and segmentation labels are provided for the test sentence, all tags and segmentation labels are activated in tag and segmentation label lexicons. The lexicons with only one candidate are regarded as *known lexicons* and others are regarded as *unknown lexicons*. Through KLs, active symbols in source lexicons will excite candidate symbols in target lexicons. Each candidate's excitation level is calculated based on (1). The least excited one is eliminated from candidate list and others are set to be active. It is noted that no matter a source lexicon is known or not, as long as its candidates are set to be active, the active symbols will always excite the symbols in unknown lexicons. In this example, "^" in lexicon 0 will excite tag candidates in lexicon 38, and active symbols in lexicon 38 will then excite tag candidates in lexicon 40, while the active symbols in lexicon 40 excite candidates, "王(king)", "工(labor)" respectively in unknown lexicon 2. This procedure iterates so that unknown character will be determined gradually by eliminating weak candidates in unknown tag lexicons, segmentation lexicons and word triplet lexicons. Finally only one candidate is left in each lexicon and the candidate will be chosen as the most likely result and "王(king)" is recalled for the missing character.

D. Knowledge Link Weighting

In the basic confabulation model, the excitation level of a candidate is the sum of contributions from active symbols in other lexicons. Intuitively, however, different source lexicons do not contribute equally to a target lexicon. For example, the lexicon right next to an unknown word obviously gives more information in determining the unknown word than the lexicon that is five words away. This motivates us to weight KL's contribution during recall.

The basic idea is to weight the contribution of each KL based on the *Mutual information* (*MI*) [9] between its source and target lexicons. Mutual information of two random variables is a measure of variables' mutual independence. In our work, mutual information is calculated as

$$I(A;B) = \sum_{b\in B}\sum_{a\in A} p(a,b)log\left(\frac{p(a,b)}{p(a)p(b)}\right) \qquad (2)$$

where A is the source lexicon and a represents symbols in A; B is the target lexicon and b represents symbols in B. $p(a,b)$ is the joint probability of symbol a and b; $p(a)$ and $p(b)$ are the margin probability of symbol a and b respectively. $I(A;B)$ is nonnegative. The value of $I(A;B)$ will increase when the correlation of symbols in lexicon A and B get stronger. Because each knowledge link has its source and target lexicons, in the rest of the paper when we say the MI of a KL we refer to the MI of the source and target lexicons of that KL.

In Section IV, detailed analysis of mutual information is presented and compared among different KLs. Based on the comparison different weighting scheme is explored to improve performance. Another useful function of mutual information is to guide the training process. As the training progresses, by monitoring the change of the mutual information of each knowledge link, we can see how much knowledge is gained and hence decide the effectiveness of the training. More discussion will be provided in the next section.

IV. Experimental Results

In this section, we compare the performance of different models and show how the analysis of mutual information can

help to improve the efficiency of the confabulation modeling and recall.

We train the Chinese confabulation model with a corpus of 10 sets of collected fairy and folk tales. We choose Chinese version of worldwide fairy tales such as Hans Christian Andersen's Fairytales, Grimm's Fairy Tales and also Chinese folk tales, because those works use vivid and common language, which will lead to a statistically meaningful knowledge base. The training set includes 364709 sentences and 3232600 words, and is chunked into 1328 small files with equal size. The test document includes 91 sentences extracted from elementary school textbook on Chinese language art. Each test sentence has 1~4 randomly picked missing Chinese words. For each missing word, 2~5 possible candidates will be given. Accuracy is measured as the rate of successfully confabulated sentences, which must be identical to the original sentences.

A. Necessity of incorporating segmentation labels and circular Knowledge Base

The first thing we want to show is the importance of including segmentation labels in the confabulation model. In this experiment, all knowledge links have the equal weight. We compare the recall accuracy of confabulation models with and without segmentation labels. The result is shown in Fig. 3. As we can see, adding segmentation label improves recall accuracy by 4.4%.

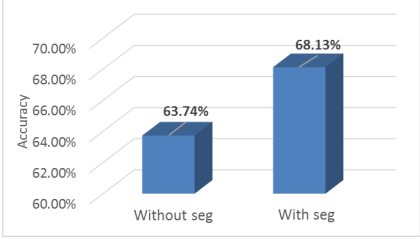

Fig. 3. Recall accuray of sentence confabulation model with/without segmentation label

Another experiment compares the training time, recall time and accuracy between non-circular model and circular model. TABLE II. shows that non-circular model takes about four times training effort more than the circular model, and 17.5% more recall time, but gives 13% lower recall accuracy.

TABLE II. COMPARISON OF NON-CIRCULAR AND CIRCULAR MODEL

	Non-circular	Circular	Improvement(%)
Training time(s)	489180	144540	70.45%
Recall time(s)	6317.22	5207.83	17.56%
accuracy	54.95%	68.13%	13.18%

B. Analysis of mutual information

In the second set of experiments, we demonstrate how the change of mutual information (MI) relates to the effectiveness of the training process. We continuously monitor the mutual information of each KL as the training process progresses. The 1328 files of the training corpus is processed one by one. The MI of each KL is calculated each time after a training file is processed.

The line chart in Fig. 4 shows the change of MI for four selected KLs as the number of processed training files increases. The blue line gives the MI of KL*0*, which connects two word lexicons of immediate neighbor. We can see that as more files are trained; the MI of KL*0* gets smaller. The grey and yellow lines in the figure give the MI of KL72 and KL94 respectively. They are the knowledge links between a single word lexicon and its corresponding tag lexicon. The MI of these KLs fluctuate within a very small range at the beginning of training. When more training files are processed, they converge to a stable value. This is because every Chinese character has its specific semantic and syntactic function and the relation between tags (or segmentation labels) and words are relatively fixed. Very few new character-tag or character-segment relationship will be learned after certain amount of training. In other words, the knowledge base becomes saturated at certain point.

The orange line gives the MI of KL50 that connects between single word lexicon and its corresponding word triplet lexicon. Our results show that there is a strong correlation between a word and its corresponding word triplet. This means a character always co-occur with a limited number of word triplets. Similar to the MI of KL0, 72 and 94, when more training files are processed, the MI of KL50 increases and then gradually saturates to a stable value.

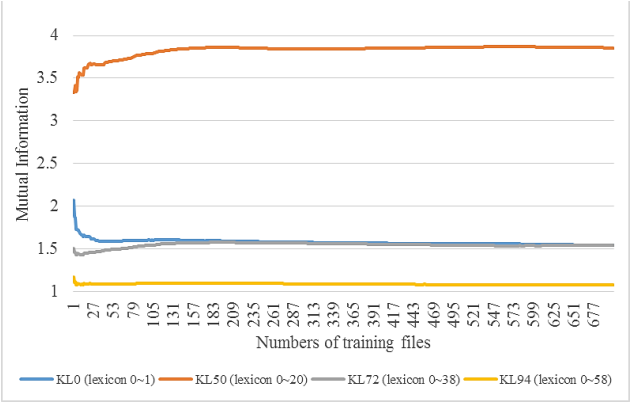

Fig. 4. Mutual Information trend chart for 4 kinds of Knowledge Links

The convergence of MI of KLs indicates that adding more training files will not necessarily increase the learned knowledge. At certain point, the knowledge acquiring speed slows down and further learning will not be as effective as before. It is the time that we should either stop the training or switch to another set of training text that has significantly different style.

Our experimental data show that most Knowledge Links' mutual information will reach ±5% and ±3% of its stable value after the model is trained with 30 and 100 files respectively. We take the knowledge base generated at different stages of training and apply them to sentence completion test. Their recall accuracy is given in Fig. 5. The X-axis gives the number of training files used to generate the knowledge base and the Y-axis give the recall accuracy. The graph shows that when training set size exceeds 300, the recall accuracy has stabilized at around 68%, and when the training set size reaches 170, the recall accuracy is already close to its peak. However, if the training set size is too small, the recall quality is not acceptable. This result agrees with Fig. 4, which shows that the MI of

knowledge links starts to converge after 170 training files and becomes very stable after 300 training files. Based on the above discovery, we set the saturation threshold of the training set size at 300. Our previous work [3] shows that the training time is linearly proportional to the size of training data. Limiting the training set size to the saturation threshold can sharply reduce training time with very little sacrifice of accuracy.

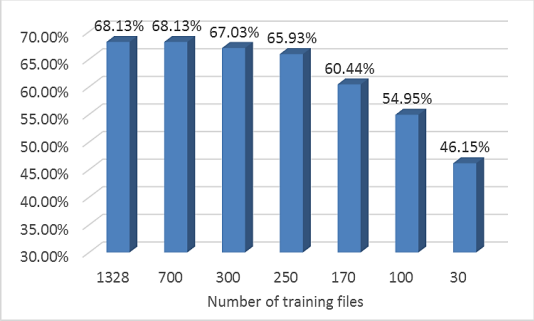

Fig. 5. Recall accuracy of different training set size

C. *Quantified Knowledge Link Weighting scheme*

The goal of next experiment is to find a systematic way to assign KL weight. Previous works [4] have shown that weighing the contribution of KLs based on their significance can improve recall accuracy, however, their weight is assigned only in an ad-hoc way. We believe that the significance of a KL can be measured by the mutual information between its source and target lexicons and therefore the MI of a KL should decide its weight.

Fig. 6 shows the mutual information of all knowledge links. Based on their connections, the KLs are divided into 9 groups. The group division is described in TABLE III. and labeled in Fig. 6 underneath the X-axis.

TABLE III. KNOWLEDGE LINK GROUP DIVISION

KL group	KL IDs	Connection
(a)	0~9	Between word lexicons
(b)	10~19	Between word triplet lexicons
(c)	20~29	Between tag lexicons
(d)	30~39	Between segmentation label lexicons
(e)	40~61	Between word and word triplet lexicons
(f)	62~83	Between word and tag lexicons
(g)	84~105	Between word and segmantation lavel lexicons
(h)	106~127	Between word triplet and tag lexicons
(i)	128~149	Between word triplet and segmentation label lexicons

We can see from Fig. 6 that, from left to right, the MI of the KLs in the same group are clustered together. And as the distance between the source and target lexicon of the KL increases, the MI of the KL decreases. For example, KL0 and KL8 belong to the same group, therefore, they have similar MI. However, since KL0 connects between two immediate neighboring lexicons while KL8 connects between two word lexicons that are 4 words apart from each other, the MI of KL0 is slightly greater than KL8. This agrees with our intuitions that adjacent characters have stronger correlations. Second, the KLs connecting to word triplet lexicons always give more information than others, therefore they should be weighed as the biggest during the recall.

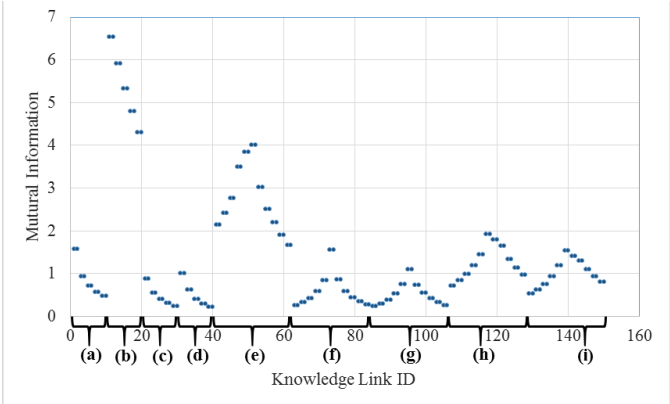

Fig. 6. Knowledge Links' mutual information

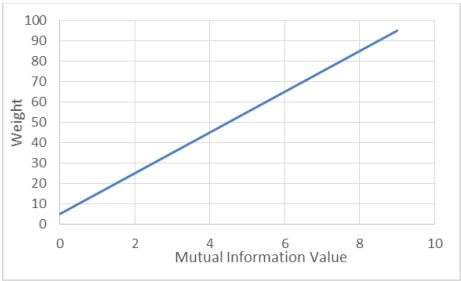

Fig. 7. Knowledge Links' weight

We assign the weight of a KL as a linear function of its mutual information as shown in Fig. 7. We then compare the recall accuracy of confabulation models with and without weighted KL. Fig. 8 shows the recall accuracy of the two sets of confabulation models. For each set of models, the Bandgap is varied from 1 to 1000. As we can see, when bandgap value is 10 or less, assigning weight to KL provides little improvement. However, when the bandgap value exceeds 100, assigning weight to KLs brings visible benefits; it improves accuracy by more than 4%. We also observe that, without weighted KL, changing the bandgap value has almost no impact on the recall accuracy. However, with weighted KL, increasing the bandgap value from 1 to 10 and 100 can increase recall accuracy from 68.13% to 69.23% and 72.53% respectively. The recall accuracy becomes saturated after the bandgap exceeds 100. Thus for the rest of the experiments, we fix the bandgap value to be 100.

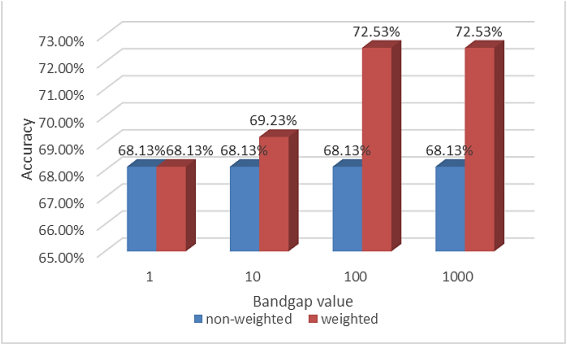

Fig. 8. Recall Accuracy of basic confabulation model of different bandgap value with/without weighting

D. *Confabulation model optimization*

Modern Chinese language is based on word compounds that consists of two or three single character words. Considering only word triplets in the confabulation model will lose information of two-word compound. Thus we add one more level of lexicons for adjacent word pairs. This brings the confabulation lexicon structure to five levels: words, word pairs, word triplets, tags, and segmentation labels. We then repeat the previous experiments to assign KL weights and evaluate the recall accuracy. Fig. 9 shows two sets of recall accuracy. The blue bars give the recall accuracy of the original 4-level confabulation model and the red bars give the recall accuracy of the new 5-level model. Both models are evaluated with and without KL weight and with two different bandgap values. The results show that adding one more layer of lexicon does not improve the recall accuracy when the KLs are not weighted and the improvement is limited if bandgap is small. However, it does make visible differences when KLs are properly weighted and bandgap is set large enough. The overall recall accuracy can be 76.9%, which is about 9% higher than the basic model without weighted KL.

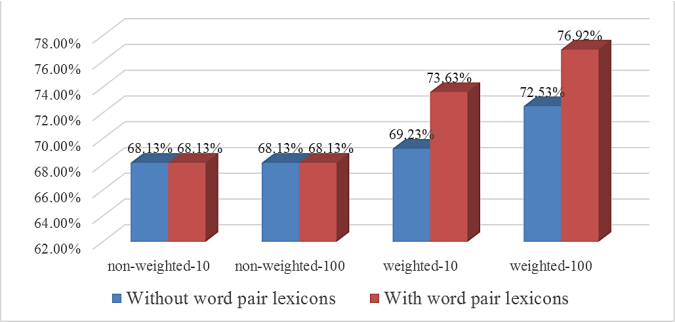

Fig. 9. Recall Accuracy of confabulation model with/without word pair lexicons (non-weighted-10 represents recall accuracy with bandgap value of 10 and without weighting scheme; non-weighted-100 represents recall accuracy with bandgap value of 100 and without weighting scheme; Weighted-10 represents recall accuracy with bandgap value of 10 and weighting scheme; Weighted-100 represents recall accuracy with bandgap value of 100 and weighting scheme)

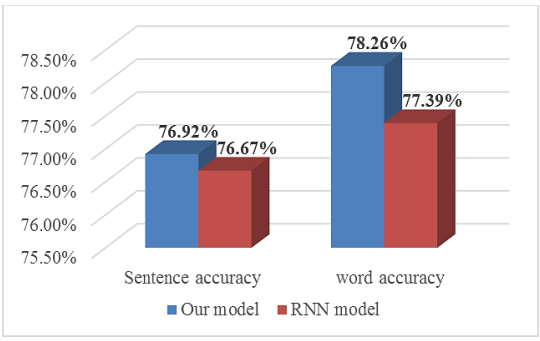

Fig. 10. Recall accuracy between different models(sentence accuracy is evaluated by the amount of sentences recalled identically to original sentences; word accuracy is evaluated by the amount of missing Chinese charaters recalled identically to the original)

As a reference, we compare the confabulation model with a recurrent neural network (RNN) model [11]. Please note that the RNN model identifies the missing word from the list of candidates by evaluating the probability of the sentence that they could make. Therefore, it has to create a sentence for each combination of the candidates and calculate its probability. The complexity of the RNN is an exponential function of the number of missing words, while the complexity of confabulation model is a linear function of the number of missing words. Fig. 10 compares the recall accuracy of the RNN model and confabulation model. It shows that these two has comparable recall accuracy and the confabulation model has slightly better word recall accuracy.

One of the advantages of using the confabulation model is that it performs syntactic parsing at the same time of sentence completion. It does not only fills in the missing characters, but also finds out the tags and segmentation labels for all words in the sentence. Fig. 11 gives the tag and segmentation label recall accuracy. Overall, 82.6% of the words are tagged correctly with POS tagging and 86.2% of the words are correctly labeled with their segmentation information. We can see that even those unknown characters are tagged and labeled with quite high accuracy. Their tagging accuracy is 75.6% and segmentation accuracy is 84.3%.

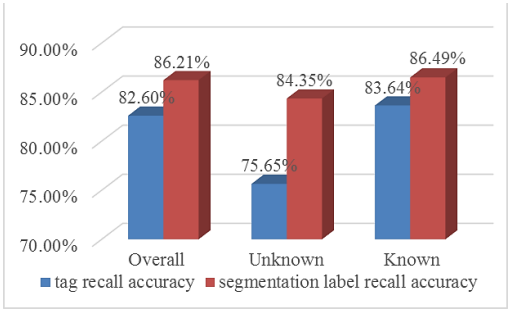

Fig. 11. Recall accuracy of tags and segmentation labels(Overall denotes accuracy for all characters in sentences; Unknown denotes accuracy for unknown missing characters, Known denotes accuracy for known characters)

Observing the mutual information, we realize that the 4th and 5th neighbors of a lexicon provide far less information than other closer neighbors. The MI of these KLs is approximately 20%~60% of the average of other KLs. This motivates us to use KLs only up to the 3rd-neighborhood. This simplification does not only reduce KB size but also save training and recall time. Fig. 12 compares the training and recall time of the original model (5-neighbor) and simplified model (3-neighbor). Experiments show that these two models give the same recall accuracy, which is 76.9%. However, the training time is decreased by 18.6% and the recall time is decreased by 53.7%. This is because the KB size is reduced from 226 to 142 by removing KLs connecting between lexicons and their 4th and 5th neighbors. These data show that, the mutual information of KLs does not only help us to assign weight to KLs for better recall accuracy, but also facilitate the decision on removing KLs with small contribution for lower model complexity without significantly sacrificing the accuracy.

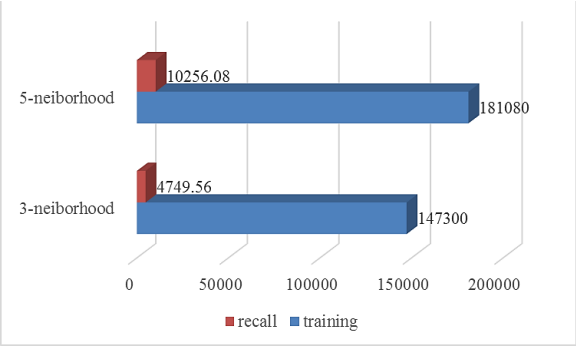

Fig. 12. Training time and recall time of different KL structure

All above results are based on the restriction that only sentences identical to the original are considered as correct. Many recalled sentences are actually syntactically correct and semantically close to the original sentence. For example, for the original sentence: 妈妈是一个很温(柔)的人(Mother is a gentle person), our recalled sentence is 妈妈是一个很温(和)的人(Mother is a gentle and mild person), in which 温柔(gentle) and 温和(gentle and mild) are synonyms. Even though the recalled sentence is not identical to the original sentence, it has very close meaning. If we treat all grammatically correct recalled sentences as successful recall, the accuracy will increase to 80%.

TABLE IV. EXAMPLES OF CONFABULATED SENTECES

Original	王明负责**(be in charge of)**检查卫生工作
Basic	王明责任**(responsibility)**检查卫生工作
Optimized	王明负责**(be in charge of)**检查卫生工作
Original	锤炼得更坚强(temper it **to be** stronger)
Basic	锤炼的更坚强(temper**ed** stronger)
Optimized	锤炼得更坚强(temper it **to be** stronger)
Original	口号特别震撼人心(Slogan **excites** people's mind)
Basic	口号特别振撼人心(Slogan **shakes** people's mind)
Optimized	口号特别震撼人心(Slogan **excites** people's mind)

Finally, TABLE IV. lists some examples of recalled sentences. The rows labeled as "Original" give the correct sentences; the rows labeled as "Basic" give the recall sentence from the original Chinese confabulation model with only word triplet lexicons and without KL weights; and the rows labeled as "Optimized" give the recall results from the optimized model, which has both word triplet and word pair lexicons as well as MI directed KL weights. The text in bold highlights the difference between the recall results. We can see that the optimized model improves the recall results semantically and syntactically.

V. CONCLUSION AND FUTURE WORKS

We proposed a Chinese sentence confabulation model by refining and modifying the English sentence confabulation model. The proposed model exploits semantic information including POS tags and segmentation labels, as well as optimized method such as circular knowledge storage, marking the start of sentence and N-neighborhood lexicon link, to successfully complete Chinese sentences with missing characters. Based on the mutual information analysis, a saturation threshold is set to the size of training set. This can sharply reduce the training time with little sacrifice of accuracy. We also found that MI directed KL weights could amplify the effect of other optimization actions, such as increasing the bandgap value and adding word pair lexicons. All together they can improve the recall accuracy by 9%. Finally the MI analysis helps us to simplify the model and reduces the overall training and recall time by 18.6% and 53.7% respectively.

ACKNOWLEDGMENT

This work is partially supported by the National Science Foundation under Grants CCF-1337300, and Air Force Research Laboratory under contract FA8750-11-1-0266.

Any Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

REFERENCES

[1] Q. Qiu, Q. Wu, and R. W. Linderman, "Unified Perception- Prediction Model for Context Aware Text Recognition on a Heterogeneous Many-Core Platform," International Joint Conference on Neural Networks, July, 2011.

[2] Q. Qiu, Q. Wu, M. Bishop, R. Pino, and R. W. Linderman, "A Parallel Neuromorphic Text Recognition System and Its Implementation on a Heterogeneous High Performance Computing Cluster," IEEE Transactions on Computers, vol. 62, pp 886-899, May,2013.

[3] Q. Qiu, Q. Wu, D. J. Burns, M. J. Moore, R. E. Pino, M. Bishop, and R. W. Linderman, "Confabulation Based Sentence Completion for Machine Reading," IEEE Symposium Series on Computational Intelligence, April, 2011.

[4] F.Yang, Q. Qiu, M. Bishop and Q. Wu, "Tag-assisted Sentence Confabulation for Intelligent Text Recognition," IEEE Symposium on Computational Intelligence for Security and Defense Applications, July, 2012.

[5] The Stanford Natural Language Processing Group, "Chinese Natural Language Processing and Speech Processing," URL: http://nlp.stanford .edu/projects/chinese-nlp. shtml.

[6] Fei Xia, "The Part-Of-Speech Tagging Guidelines for the Penn Chinese Treebank," URL: http://www.cis.upenn.edu/~chinese/posguide.3rd.ch. pdf.

[7] R. Hecht-Nielsen, "Confabulation Theory: The Mechanism of Thought," Springer, August 2007.

[8] Bill Tong, "Linguistic Features of the Chinese Language Family", URL: http://www.oakton.edu/user/4/billtong/chinaclass/Language/linguistics.h tm.

[9] Z. R. Yang, M. Zwolinski, "Mutual information theory for adaptive mixture models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, PP 396-403, April, 2001

[10] F. Jelinek, "Statistical methods for speech recognition," Proc. of the IEEE, vol. 64, pp 532-536, April,1976.

[11] T. Mikolov, S. Kombrink, L. Burget, J.H. Cernocky, Sanjeev Khudanpur, "Extensions of recurrent neural network language model," In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 5528–5531, May,2011.