Tag-assisted Sentence Confabulation for Intelligent Text Recognition*

Fan Yang[1], Qinru Qiu[1], Morgan Bishop[2], and Qing Wu[2]

[1]Dept. of Electrical Engineering & Computer Science
Syracuse University
Syracuse, NY 13244, USA

[2]Air Force Research Laboratory
Information Directorate, RITC
525 Brooks Road, Rome, NY 13441, USA

Abstract — **Autonomous and intelligent recognition of printed or hand-written text image is one of the key features to achieve situational awareness. A neuromorphic model based intelligent text recognition (ITR) system has been developed in our previous work, which recognizes texts based on word level and sentence level context represented by statistical information of characters and words. While quite effective, sometimes the existing ITR system still generates results that are grammatically incorrect because it ignores semantic and syntactic properties of sentences. In this work, we improve the accuracy of the existing ITR system by incorporating parts-of-speech tagging into the text recognition procedure. Our experimental results show that the tag-assisted text recognition improves sentence level success rate by 33% in average.**

Keywords – cogent confabulation, text recognition, parts-of-speech tagging

I. Introduction

Autonomous and intelligent recognition of printed or hand-written text image is one of the key features to achieve situational awareness. Although generally effective, conventional Optical Character Recognition (OCR) tools or pattern recognition techniques usually have difficulties in recognizing images that are noisy, or even incomplete due to the damages to the printing material, or obscured by marks or stamps. However, such tasks are not too difficult for humans as we predict the missing information by associating it with its context.

Many human cognitive processes involve two interleaved steps, sensing and information association. Together, they provide higher accuracy. In our previous work [1][2][11], a proof-of-concept prototype of context-aware Intelligence Text Recognition (ITR) system is developed. The ITR system is inspired by the human cognitive process. Instead of relying on complicated signal processing, it combines large number of simple, fuzzy and independent pattern classification models with powerful information association function. The lower layer of the ITR system performs pattern matching of the input image using a simple non-linear autoassociative neural network model called *Brain-State-in-a-Box* (*BSB*) [3]. It matches the input image with the stored alphabet. Each BSB model is analogous to a cortical column in the primary sensory area that performs the preliminary detection. Sometimes, multiple matching patterns may be found for one input character image. The upper layer of the ITRS performs information association using the *cogent confabulation* model [4]. It enhances those BSB outputs that have strong correlations in the context of word and sentence and suppresses those BSB outputs that are weakly related. In this way, it selects those characters that form meaningful words and sentences. Each confabulation model is analogous to a cortical column in the sensory association area that associates the primary detections to form high level cognition.

One of the major limitations of the current ITR system lies in its sentence confabulation function. Current sentence confabulation model fills in missing words (or narrow down ambiguous words) simply based on the word level and phrase level probabilities extracted from the training text. It ignores semantic and syntactic properties of sentences. We believe that linguistic knowledge could be used to improve the accuracy of sentence confabulation and generate more meaningful outputs.

In this work, we overcome this limitation by integrating *parts-of-speech* (*POS*) tagging with sentence confabulation. Part-of-speech tagging is a powerful Natural Language Processing tool for categorizing useful information. It is usually used to identify the function of words in a known text in order to build relational database [12] or distinguish different pronunciations for speech recognition [14]. Due to the simplicity of the cogent confabulation model, the integration with POS tagging can be achieved naturally. When used in the ITR system for text image recognition, the tag-assisted sentence confabulation improves sentence level success by 33% in average.

The remainder of the paper is organized as follows. A brief introduction of background in cogent confabulation and POS tagging is provided in Section 2. In Section 3 we introduce the modeling and operation of tag-assisted sentence confabulation. The overall ITR system with POS tagging is also described. The experimental results and discussions are presented in Section 4. Section 5 summarizes the work.

II. Background

A. Cogent confabultion

Cogent confabulation [4] is an emerging computational model that mimics the Hebbian learning, the information storage and inter-relation of symbolic concepts, and the recall operations

* Received and approved for public release by AFRL on 04/27/2012, case number 88ABW-2012-2529.

978-1-4673-1417-6/12/$31.00 ©2012 IEEE

of the human brain. Based on the theory, the cognitive information process consists of two steps: learning and recall. During the learning step, the knowledge links are established and strengthened as symbols are co-activated. During recall, a neuron receives excitations from other activated neurons. A “winner-takes-all” strategy takes place within each lexicon. Only the neurons (in a lexicon) that represent the winning symbol will be activated and the winner neurons will activate other neurons through knowledge links. At the same time, those neurons that did not win in this procedure will be suppressed.

A computational model for cogent confabulation is proposed in [4]. Based on this model, a *lexicon* is a collection of symbols. A *knowledge link* (*KL*) from lexicon *A* to *B* is a matrix with the row representing a source symbol in *A* and the column representing a target symbol in *B*. The (*i*, *j*)th entry of the matrix represents the strength of the synapse between the source symbol s_i and the target symbol t_j. It is quantified as the conditional probability $P(s_i \mid t_j)$. The collection of all knowledge links is called a *knowledge base* (*KB*). The knowledge bases are obtained during the learning procedure. During recall, the excitation level of all symbols in each lexicon is 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:

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

The function $I(s)$ is the excitation level of the source symbol s. Due to the “winner-takes-all” policy, the value of $I(s)$ is either “1” or “0”. The parameter p_0 is the smallest meaningful value of $P(s_i \mid t_j)$. The parameter B is a very large positive 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. Stanford parts-of-speech tagging

Part-of-speech (POS) tagging [5][6] is a matured technique developed for natural language processing. One of the most widely used probabilistic tagging systems is the Stanford POS Tagger [8]. It is based on the 36 word level tags specified by the Penn Treebank Tagging system. Table 1 lists some examples of these tags. During the training procedure, it scans the manually tagged training text to extract features, which is the tagging (t) of a word and the context (h) of the word to be tagged (i.e. one word before and after it.) The condition probably $p(t|h)$ is then calculated for maximum entropy.

For testing, a sentence without tags is given, the Stanford POS Tagger use the training data to calculate the entropy of the sentence with different tag sequences using the following equation:[6]

$$E = \sum_{h\in X, t\in T} \tilde{p}(h)p(t|h)f(h,t)$$

$\tilde{p}(h)$ is the empirical probability of the sequence of tags for the sentence. $p(t|h)$ is the conditional probability of the tag, and $f(h,t)$ is a constrain function used to improve the accuracy of special cases. T is the set of all possible tags while X is the set of all possible tag sequences available from the training data. The maximum entropy tag sequence is selected as the most likely one, and the tags are assigned to each word.

Tag	Function	Example
CC	Coordinating conjunction	and, or, but…
CD	Cardinal number	one, two, three, …
DT	Determiner	the, this, any,…
EX	Existential there	there,
IN	Preposition or subordinating conjunction	of, for, with,…
JJ	Adjective	worthy, clean, sick,…
NN	Noun, singular or mass	kettle, curiosity, ….
NNS	Noun, plural	infants, noses, …
VB	Verb, base form	tell, eat, …
VBD	Verb, past tebse	told, began, …
…	…	…

Table 1 Examples of Penn Treebank Tags

In addition to probabilistic model such as the Stanford tagger, some work incorporates rule based technique as well. The authors of reference [7] use conditional probability to establish confidence scores for rule-based and statistical driven POS tag confabulation. When a discrepancy between the models occurs, the one with higher confidence level is chosen. Their study shows significant tag accuracy improvement when there is a suitable rule to distinguish between different candidates from the statistical model. However, when no rules are identified, the Text-to-Speech tagging generates more error than a pure probability model.

From our perspective, the ITR system is designed to recognize text purely based on knowledge (i.e. statistics) extracted from standard corpora. Rule-based tagging limits the flexibility of the design and introduces significant overhead that may not yield sufficient accuracy improvement to offset the throughput reduction.

III. Tag-Assisted Sentence Confabulation

A. Original sentence confabulation framework

Similar to the original sentence confabulation framework [8] we assume that the maximum length of a sentence is 20 words. Any sentence that is longer than 20 words will be truncated. We also assume that the empty space is a word. Any sentence that is shorter than 20 words will be padded with empty spaces.

The original sentence confabulation framework consists of two levels of lexicons. Lexicons 0 through 19 belong to the first level. Each level 1 lexicon associates to a single word in the sentence. The *i*th lexicon represents the *i*th word. Lexicons 20~38 belong to the second level. Each level 2 lexicon associates to a pair of adjacent words. The lexicon labeled (20+*i*) represents the pair of words in the (*i*+1)th and (*i*+2)th location. Associated to each lexicon is a collection of symbols. A symbol is a word or a pair of words that appears in the corresponding location. We use S_A to denote the set of symbols associated to lexicon A.

A knowledge link (KL) from lexicon A to B is a $M \times N$ matrix, where M and N are the cardinalities of symbol sets S_A and S_B. The i,jth entry of the knowledge link gives the conditional probability $P(i|j)$, where $i \in S_A$, and $j \in S_B$. Symbols i and j are referred as *source symbol* and *target symbol*.

For our sentence completion system, between any two lexicons there is a knowledge link. If we consider the lexicons as vertices and knowledge links as directed edges between the vertices, then they form a complete graph.

B. Sentence confabulation framework with POS tagging

With the addition of tags, a new level of lexicons is added. Lexicon 39~58 are the POS tags for word lexicons 0~19. The structure of knowledge links is exactly the same as in the original confabulation model.

During training, the reference text is passed through Stanford POS tagger first to generate their respective tags. Knowledge links are established between word lexicons and tag lexicons, but not between word-pair lexicons and tag lexicons. This is because the word pair knowledge links are derivatives of the word knowledge links; therefore they are not needed to build knowledge links with tags.

Since a sentence without tags is given for testing, the confabulation model automatically assumes all tags are possible candidates for all tag lexicons. The system calculates excitation level for all candidates during each iteration and eliminates the least excited one. This elimination method allows multiple candidates to compete throughout the confabulation process and provides more cognitive capacity.

The concept of multiple tag candidates racing has also been proposed in reference [9]. The authors show that if a single tag is chosen in each decision iteration, the tag error rate is compounded. They use the data provided in [10] to show that the accuracy of Penn Treebank tag is about 92%. For a sentence with 15 words, the probability of fully correct tag confabulation drops down to $(0.92)^{15} = 28.6\%$. By allowing multiple candidates and learning based statistical model, the full sentence tag accuracy can be improved to 79.5%.

C. Training and recall functions

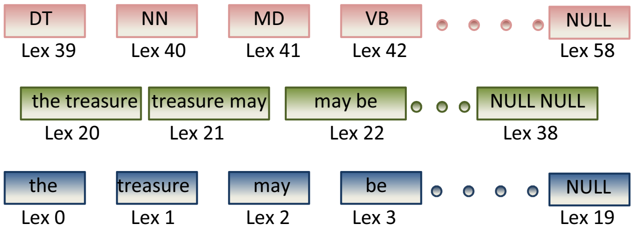


Figure 1 Training Function Lexicon Structure

Just like the original sentence confabulation model, the tag-assisted version is also divided into two parts, Training and Recall. The Training function uses reference text with tags to build the knowledge base, while the Recall function use the knowledge base to confabulate incomplete sentences with no tags.

Figure 1 shows an example of a given training sentence and its corresponding lexicon structure. The sentence is "the treasure may be hard to find". The tags are:

```
the_DT treasure_NN may_MD be_VB hard_JJ to_TO find_VB
```

In order to extend it to 20 words, we pad 14 empty words and tags to the end of the sentence. Each word will then be enter into lexicon 0~19 respectively as symbols, and each word pair will be enter into lexicon 20~38. Then the tags following each word will be entered into lexicon 39~58. The system will adjust the value of all knowledge links between lexicons to learn from the sentence and tag. For example, the *KL* from lexicon 0 to lexicon 1 will be adjusted by increasing the conditional probability $P("the"|"treasure")$. The *KL* from lexicon 0 and lexicon 39 will also be adjusted by increasing the conditional probability $P("the"|"DT")$. Obviously, if the words and tags have frequent co-occurrence, their corresponding entry in the knowledge link will have a high value.

Once all training texts are processed, the training process is complete and all final knowledge links are available for the Recall function.

Figure 2 is a very simple illustration of the recall function that uses the confabulation model to complete a test sentence with an unknown word and tags.

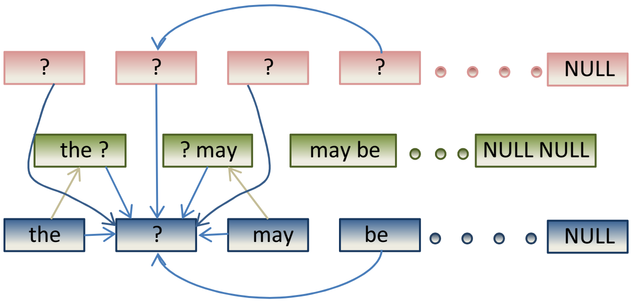


Figure 2 Tag-assisted Sentence Confabulation

For the sake of illustration, the testing sentence is the same as the training sentence in Figure 1, with the word "treasure" missing and without pre-processed POS tags. Each square still represent lexicons at different levels and question marks indicate pieces of missing information. As we can see, given a sentence with one missing word, the associated word pair lexicon are also unknowns.

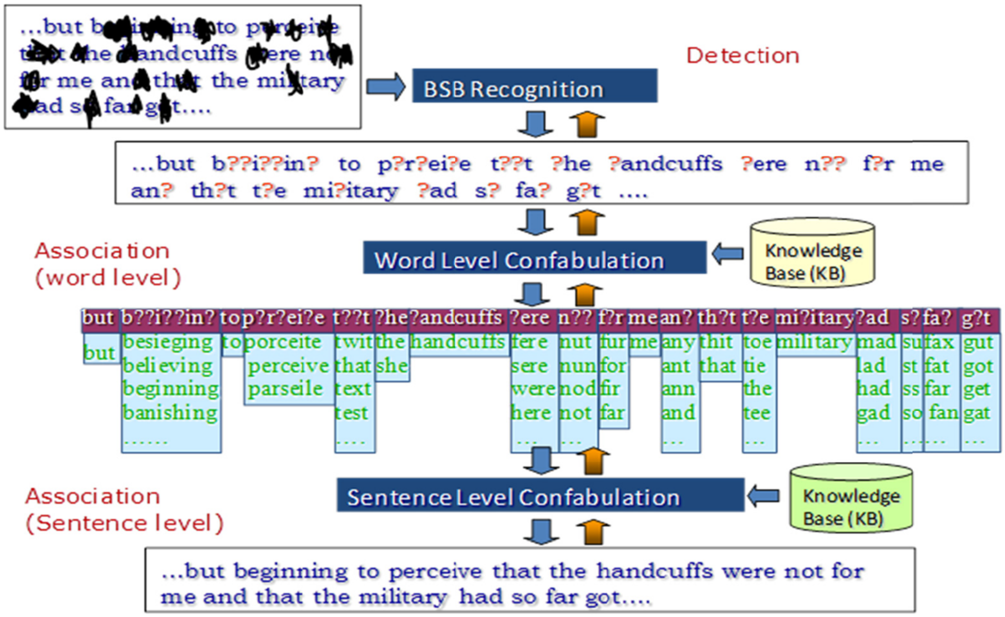
Given a lexicon that has missing information, there is either a set of given candidates or all possible symbols associated to this lexicon are considered as candidates. In Figure 2, arrows are knowledge links from source lexicon to target lexicon. Arrows of different colors indicate that the knowledge links are used to excite lexicons on different levels. For example, all blue arrows are knowledge links used to excite the unknown word lexicon; while all red arrows are knowledge links used to excite the tag lexicon directly above the unknown word. The active symbols in the source lexicon will excite candidate symbols in the target lexicon, and the excitation level is determined by the corresponding value of the knowledge link. As shown in the figure, the confabulation model calculates the excitation level of all candidates to confabulate the unknown word in lexicon 1. It eliminates the least excited one and set others as active. Consequently, the model also needs to calculate the excitation levels of the initially unknown tags and word pairs. This procedure iterates until only one candidate is left in each lexicon. This candidate usually has the highest excitation level and will be chosen as the most likely result.

In the basic confabulation model, the total excitation level of a candidate is the sum of all contributions from other lexicons. However, intuitively not all the lexicons should contribute to an unknown candidate equally. For example, knowledge links from adjacent words are much more important than knowledge links from far away words in determining an unknown word. In the experimental results section, we will explore different KL weight schemes to find their impact to the performance of the recall function.

D. Tag-assisted text recognition

The tag-assisted sentence confabulation is integrated with the aforementioned ITR system to include OCR and word confabulation. This allows us to test the effectiveness of the sentence confabulation in a realistic environment. The inputs are scanned images of text. The output is the recognized text itself.

The ITR system is divided into 3 layers as shown in Figure 3. The input of the system is the text image. The first layer is character recognition software based on BSB models. It tries to recall the input image with stored image of the English alphabet. In this work, a race model is adopted. The model assumes that the convergence speed of the BSB indicates the similarity between patterns. For a given input image, we consider all patterns that converge within 50 iterations as potential candidates that may match the input image. All potential candidates will be reported as the BSB results. Using the racing model, multiple matching patterns will be found if there is noise in the image or the image is partially damaged. For example, a horizontal scratch in the letter "T" will make it look like the letter "F". In this case we have ambiguous information.



Detection
BSB Recognition
…but b??i??in? to p?r?ei?e t??t ?he ?andcuffs ?ere n?? f?r me an? th?t t?e mi?itary ?ad s? fa? g?t ….
Association (word level)
Word Level Confabulation
Knowledge Base (KB)
but b??i??in? to p?r?ei?e t??t ?he ?andcuffs ?ere n?? f?r me an? th?t t?e mi?itary ?ad s? fa? g?t
but
besieging believing beginning banishing ……
to
porceite perceive parseile
twit that text test ….
the she
handcuffs
fere sere were here …
nut nun nod not …
fur for fir far
me
any ant ann and …
thit that
toe tie the tee …
military
mad lad had gad …
su st ss so …
fax fat far fan …
gut got get gat …
Association (Sentence level)
Sentence Level Confabulation
Knowledge Base (KB)
…but beginning to perceive that the handcuffs were not for me and that the military had so far got….

Figure 3 Information processing flow of the ITR system

The ambiguity can be removed by considering the word level and sentence level context, which is achieved in the second and third layer where word and sentence recognitions are performed using cogent confabulation models. The models fill in the missing characters in a word and missing words in a sentence. The three layers work cooperatively. The BSB layer performs the word recognition and it sends the potential letter candidates to the word level confabulation. The word recognition layer forms possible word candidates based on those letter candidates and sends this information to the sentence recognition layer.

IV. Experimental Results

As mentioned in Section III.D, knowledge link weight can greatly influence the quality of the sentence confabulation. In the experiments, we will first test the impact of different weights of knowledge links in order to search for the optimum weight scheme. Then we will compare the tag-assisted confabulation with untagged confabulation to evaluate the effectiveness of incorporating POS tagging.

The ITR system is trained with a training corpus consisting of 73 folk tales, and the testing document is an untrained text in the same category. The testing document has 523 sentences, and the success rate in this section is always measured as number of correctly confabulated sentences over the number of total sentences. A sentence is considered correct only if it is identical to the sentence in the original text.

A. Knowledge link structure and weight testing

To test the effectiveness of tag-assisted confabulation, we randomly introduce 3-pixel wide horizontal strikes to 10% characters of a scanned text image. The BSB character recognition is often unable to identify the correct character and give ambiguous results. Then it will be the responsibility of the word and sentence level confabulation to remove the ambiguity.

In the first experiments, we vary the number of tag lexicons that have knowledge links with each word lexicon. The number is denoted as *N*. For an *N*-tag model, each word lexicon is connected to *N* tag lexicons. The *i*th word lexicon connects to the *i*th tag lexicon and its (*N*-1)/2 neighbors. For example, for a 5-tag model, to calculate the excitation level of an unknown word lexicon, we only consider its direct tag lexicon and two nearby tag lexicons on each side of the direct tag.

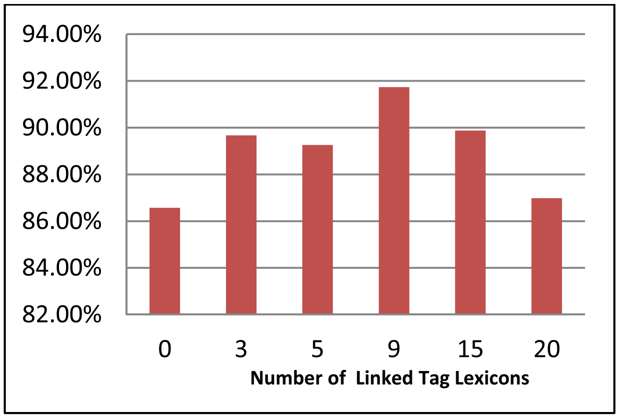


Figure 4 Results of *KL* structures test

We compare the recall accuracy of different confabulation models by varying the number of linked tag lexicons from 0 to 20. The results are given in Figure 4. In this experiment, 0-tag means the confabulation does not use tag at all, 20-tag lexicons means the confabulation use all 20 tags for each unknown words. As we can see, while using too few POS tags leads to relatively poor accuracy, using too many tags is equally bad. This is because far away tags do not contribute as much information to determine an unknown word as its direct tag does. Due to the lack of deterministic relation, these remote tags will even increase noise in the confabulation procedure. Based on our experiments, the optimum number of linked tags is 9. This setting will be used in all following tests.

Next we test the weight of some primary knowledge links. We speculate that the knowledge links between adjacent word lexicons and adjacent tag lexicons carries more information than others. And hence should play a more important role in confabulation than other knowledge links. In addition, the knowledge link between word and its direct tag should also be much stronger than others. It is our hypothesis that, scaling up the excitation value of these primary knowledge links will yield better confabulation results.

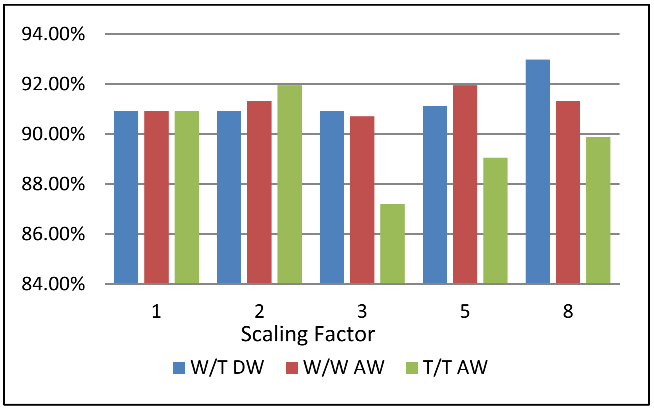


Figure 5 Results of *KL* Weight Test

In the second experiment, we selectively scale up the excitation value of each one of the above mentioned primary knowledge links. The scaling factor *M* is varied from 1 to 8. All other knowledge links have a scaling factor of 1. Figure 5 shows the success rate of various weights for the three primary knowledge links. In the figure, W/TDW is used to represent the knowledge links between word and its direct tag; W/WAW is used to represent the knowledge links between adjacent words; T/TAW is used to represent the knowledge links between adjacent tag lexicons. In all these tests, we use 9 link-tag lexicons. The results show that setting the scaling factor of the KLs between adjacent tags (i.e. T/TAW) greater than 2 will degrade the system performance, while the scaling factor of KLs between words and their direct tags should be set to very high.

We select the scaling factor with the highest success rate for each knowledge link category and form our optimum weight scheme.

B. Evaluate the performance of tag-assisted confabulation

Using the knowledge structure and weight discovered in previous experiments, we configure the ITRS to evaluate the effectiveness of incorporating POS tag in text recognition. The tag-assisted confabulation method is compared with no-tag confabulation at various noise levels. The noise level percentage means the ratio of characters in text with a 3-pixel wide horizontal strike. Note that the size of original character is 15x15 pixels, a 3-pixel wide strike is almost equivalent to 20% distortion.

Figure 6 shows that no-tag sentence confabulation quickly collapse as noise level increases. This is because each test sentence contains on average 28 characters and we only consider the sentence correct if all of its characters are correct. The noise level at character level is compounded into character and word level ambiguity. Without semantic information, which provides an overall structure for each sentence, the success rate is expected to drop exponentially as noise level increase.

Tag-assisted confabulation shows clear improvements over no-tag confabulation at all noise levels. The improvement is minor at low noise level, but significant at high noise level. Overall, tag-assisted confabulation improves success rate by 33% in average.

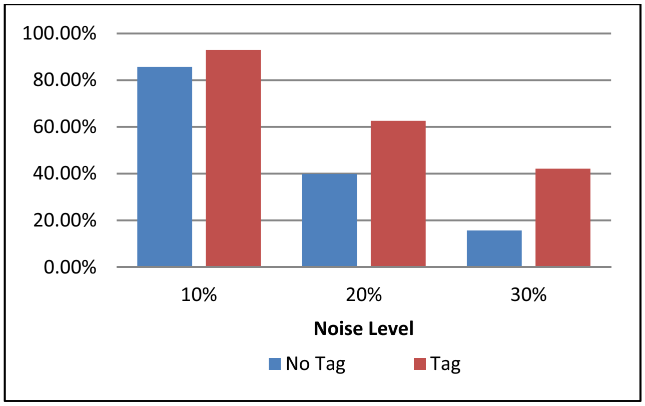


Figure 6 Accuracy comparison between tag-assisted and no-tag sentence confabulation

Some of the sentences recognized by ITR system with and without tag are listed in Table 2. The text in bold highlights the difference between the confabulation results with and without tag. As we can see, the integration with POS tag greatly improves the sentence structure syntactically and semantically.

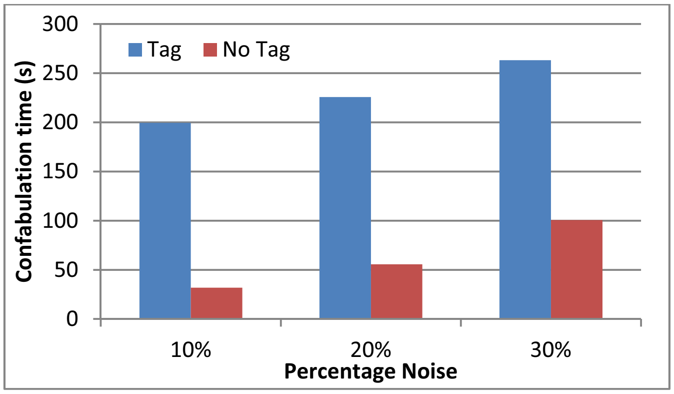


Figure 7 Runtime comparison between tag-assisted and non-tag confabulation

The tag-assisted sentence confabulation achieves great improvement in accuracy at the cost of increased computation complexity. Figure 7 shows the computation time of tag-assisted and non-tag confabulation as the percentage noise level varies from 10% to 30%. Although the tag-assisted confabulation is consistently slower than no-tag confabulation, the difference is decreasing as the noise level increases. At 10% noise level, tag-assisted confabulation is about 6.7 times slower than no-tag confabulation, while the number is reduced to 2.5 at 30% noise level. This is because, the tag-assisted confabulation consider all existing tags as potential candidate. This is a significant overhead at low noise level. However, as the noise level increases, the ambiguity of characters and words increases, but the ambiguity of tags does not increase. Therefore the overhead becomes less significant.

Table 2 Examples of confabulated sentence

Original	and they returned as they ***came***
No-tag	and they returned as they ***come***
Tagged	and they returned as they ***came***
Original	then ***cassim grew*** so envious that he could not sleep
NO-tag	then ***cassia grow*** so envious that he could not sleep
Tagged	then ***cassim grew*** so envious that he could not sleep
Original	***whom*** ali ***baba took*** to be their captain
NO-tag	***whim*** ali ***baby look*** to be their captain
Tagged	***whom*** ali ***baba took*** to be their captain
Original	you pretend to be poor ***and*** yet you measure ***gold***
NO-tag	you pretend to be poor ***end*** yet you measure ***fold***
Tagged	you pretend to be poor ***and*** yet you measure ***gold***
Original	which was ***full*** of ***oil***
NO-tag	which was ***cult*** of ***iii***
Tagged	which was ***full*** of ***oil***
Original	***ten mules loaded*** with great chests
NO-tag	***ken mules lauded*** with great chests
Tagged	***ten jules loaded*** with great chests
Original	***we are certainly*** discovered
NO-tag	***me fro certainty*** discovered
Tagged	***we are certainly*** discovered

V. Conclusions and Future Works

We have introduced the modeling, training and recall techniques of tag-assisted sentence confabulation. The proposed technique incorporates semantic information with the confabulation model and it generates more sentences that are grammatically correct. As shown in our result section, the tag-assisted confabulation is especially effective at high noise level. The increase in success rate ranges from 10% to 55%. This is a very essential add-on to provide sematic information to lexicon based algorithms in text recognition applications demanding high accuracy.

On the other hand, the main drawback of implementing tag lexicons is longer execution time. In our experiment depending on the noise level, no-tag confabulation on average processes roughly 5 unknown lexicons and 22 knowledge links for each lexicon, while tag-assisted confabulation on average processes 25 unknown lexicons and 20 knowledge links for each lexicon. This overhead can be reduced by parallel processing. Applications that demand high throughput will have to evaluate the proposed confabulation method depending on the hardware available.

Another weakness for the tag-assisted confabulation model is its dependency on context information at sentence level. This prohibits tag confabulations to perform well for short sentences due to less available information. One possible solution to this problem is to consider context at higher level. For example, use information from sentences before and after current one. This will be the direction of our future research.

Acknowledgment of Support and Disclaimer

This work is funded by the 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," *Proc. Of 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," to appear in *IEEE Transactions on Computers*.

[3] J. A. Anderson, "An Introduction to Neural Networks," *The MIT Press*, 1995.

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

[5] Toutanova, K.; Manning, C. D. "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger," *Proc. Of the SIGDAT conference on Empirical methods in natural language processing and very large corpora*, 2000.

[6] Ratnaparkhi, A. "A Maximum Entropy Model for Part-of-Speech Tagging," *Proc. of the Empirical Methods in Natural Language Processing*, pp. 133-142, 1996.

[7] Ming Sun; Bellegarda, J.R.; , "Improved pos tagging for text-to-speech synthesis," *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* , vol., no., pp.5384-5387, 22-27 May 2011

[8] "Stanford Log-linear Part-Of-Speech Tagger," The Stanford Natural Language Processing Group, URL: http://nlp.stanford.edu/software/tagger.shtml, Accessed October, 2011.

[9] Faili, H.; , "Building deep dependency structure from partial parses," *Computer Conference, 2009. CSICC 2009. 14th International CSI* , vol., no., pp.247-252, 20-21 Oct. 2009 doi: 10.1109/CSICC.2009.5349409

[10] S. Bangalore, "Complexity of Lexical Descriptions and its Relevance to Partial Parsing", PhD thesis, Department of Computer and Information Sciences, University of Pennsylvania, 1997.

[11] Qinru 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," *Proc. of IEEE Symposium Series on Computational Intelligence*, April, 2011.

[12] Z. Liu, Y. Wang, "A Novel Method of Chinese Web Information Extraction and Applications", WASE International Conference on Information Engineering, 2009.

[13] M. Bulut, S. Lee, S. Narayanan, "A Statistical Approach for Modeling Prosody Features using POS Tags for Emotional Speech Synthesis", IEEE International Conference on Acoustics, Speech, and Signal Processing, 2007.

[14] Schlueter, R.; Nussbaum-Thom, M.; Ney, H.; , "Does the Cost Function Matter in Bayes Decision Rule?," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.34, no.2, pp.292-301, Feb. 2012, doi: 10.1109/TPAMI.2011.163