The Role of Shared Attention in Human-Computer Conversation

Hideki Kozima and Akira Ito
Communications Research Laboratory[1]

Abstract

This paper describes our on-going project on human-computer and human-computer-human conversation. We here emphasize the role of "shared attention" in verbal and non-verbal communications. Shared attention spotlights things and events being mentioned in the conversation and makes the discourse coherent about the same topic. Being inspired by communication disorders in autism, we assume that shared attention plays an indispensable role in naturally interactive and communicative conversation, since its function is observed in infants at pre-verbal stage and its malfunction is observed in infants and children with autism. Focusing especially on "shared visual attention", that is simply "looking at the same object", we are developing a computer system that can create and maintain shared visual attention with humans by monitoring their gaze-direction. We are planning experiments on human interaction with this system in order to evaluate and elaborate our model of shared attention in conversation.

1. Introduction

Human-computer conversation is one of the most challenging targets of computational linguistics. Researchers have developed a number of computational devices for analyzing and generating speech, sentences, and discourse. Integrating these devices, however, no one achieved natural human-computer conversation like that of HAL 9000.

We emphasize here that "shared attention" (Baron-Cohen 1995) have been missing in the former studies. Shared attention, that is attention shared by speakers and hearers, conveys a clue to what has "relevance" (Sperber 1986) to the current context. It plays a dominant role not only in encoding and decoding referring expressions but also in making coherent discourse "being about what we are paying attention to". (Note that psychologists often use the term "joint attention" instead of "shared attention".)

This paper describes our on-going research on the role of shared attention in verbal and non-verbal communications. We currently deal with "shared visual attention" (Baron-Cohen 1995), that is simply "looking at the same objects", as one of the most fundamental devices of pre-verbal communication of infants as well as verbal communication of adults.

The following section briefly describes the nature of shared attention in human

[1] Iwaoka 588-2, Iwaoka-cho, Nishi-ku, Kobe 651-24, Japan. ({xkozima,ai}@crl.go.jp)

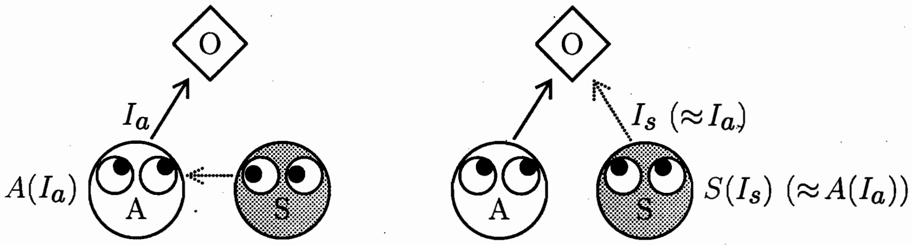


Fig. 1 Shared attention and estimation of others' mental states.

communication. Shared attention enables us to estimate others' intentions and emotions behind their explicit behavior. Section 3 introduces the attention-sharing system being developed. The system is intended to create and maintain shared visual attention by monitoring people's gaze-direction. Section 4 gives our preliminary conclusion and draws our plan for future research.

2. Shared Attention in Communication

Human communication is an activity of creating "shared world" with others. Shared world consists of things and events that are physically or mentally manipulatable from speakers and hearers, and it works as a field for exchanging information with others and understanding others' mental states.

Shared attention, especially shared visual attention, plays an indispensable role in creating shared world with others, since one's attentional target is closely related to his or her belief and desire (Frith 1991, Baron-Cohen 1995). Figure 1 illustrates how shared visual attention is created. First, Self (e.g. an infant) captures the gaze-direction of Agent (e.g. his or her caretaker). Then, Self searches in the direction and identifies Object to which Agent is paying attention.

Shared visual attention is one of the developmentally fundamental devices for communication. Visual attention-sharing is observed in infants at the pre-verbal stage: its development starts before 6 months old and completed around 18 months old (Butterworth 1991). In addition, it is also observed in some species of non-human primates, e.g. chimpanzees and orangutans (Itakura 1996).

Most infants and children with autism can not create shared visual attention with others; being instructed by an experimenter, however, they can do it (Baron-Cohen 1995). This means they are unaware that one's gaze-direction implies his or her attentional target. Unawareness of others' attention results in "mindblindness" (Baron-Cohen 1995), that is a disability of estimating others' mental states in terms of shared attention, which causes autism's typical disorders in verbal and non-verbal communications.

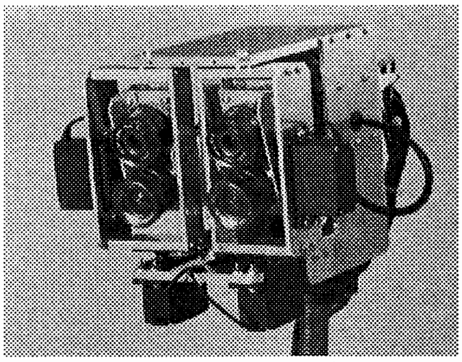


Fig. 2 Attention-sharing system.

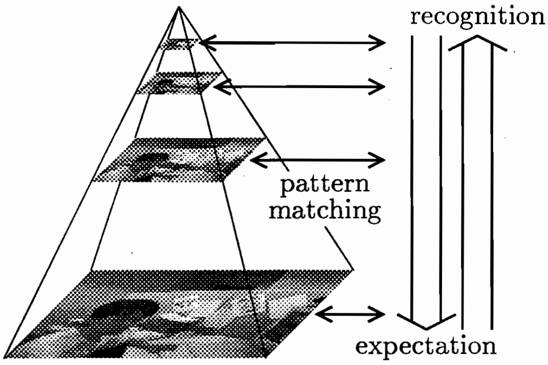


Fig. 3 Hierarchical image processing.

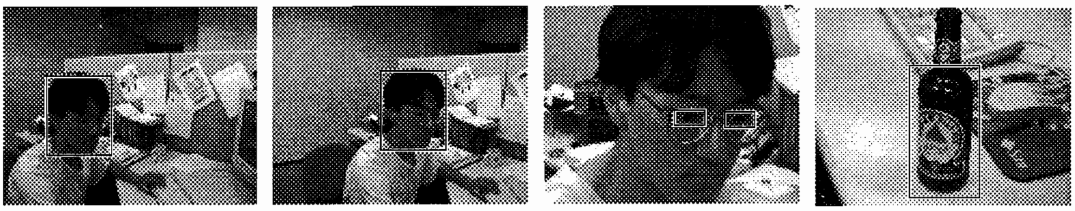
The role of shared attention in human-computer conversation lies in (1) capturing "relevance" and (2) estimating "others' mental states". First, shared attention enables us to select a target object relevant to the current context. Communication of intentions is often achieved by just pointing to a relevant object (e.g. pointing to a clock when you want to make someone hurry). Though it can tell us only physical targets in our sight, shared visual attention provides developmental basis for higher-level attention-sharing.

Secondly, shared attention enables us to estimate "others' mental states": once a hearer selected a relevant object in terms of shared attention, the hearer can co-observe speaker's sensory-input (what Agent is perceiving from the target object) and the subsequent mental states (speaker's intentions and emotions). As illustrated in Fig. 1 again, Self can estimate Agent's sensory-input I_a being perceived from Object O, then Self can simulate Agent's mental states $A(I_a)$ by applying Self's mind $S(\cdot)$ to the pseudo-input I_s thus co-observed. Since shared attention guarantees $I_a \approx I_s$ and our innate and/or cultural bias expects $A(\cdot) \approx S(\cdot)$, the simulation result $S(I_s)$ becomes a good approximation of $A(I_a)$.

3. The Attention-Sharing System

We are developing a computer system that can share visual attention with people in terms of monitoring their gaze-direction (Tanenhaus 1996). The system, though it is still under development, is intended as a computational model of the cognitive module for visual Attention-sharing that will be incorporated into our system of human-computer conversation.

The attention-sharing system consists of a robot head with anthropomorphic shape shown in Fig. 2, and a standard workstation. The robot head has four CCD monochrome cameras (left/right × zoom/wide) and four servo motors to drive the directions of the left/right "eyes" at the speed of human saccade. The images taken by these cameras are sent to the workstation for a gaze-monitoring procedure. For real-time gaze-monitoring,



(1) detect a face (2) saccade (3) capture gaze (4) identify the target

Fig. 4 Gaze-monitoring process.

we employed "hierarchical image processing" illustrated in Fig. 3.

The gaze-monitoring procedure consists of the following computational stages on "images" and "relevance". (See also Fig. 4.)

1. Detect a face (under varying pose and size) in a complex scene.
2. Saccade to the face and switch to the zoom cameras for precise face images.
3. Detect eyes and capture the gaze-direction in terms of the position of pupils. If it is impossible, capture face-direction instead.
4. Search for an object in the gaze-direction. If something relevant to the current context is found, identify it as a target object.

We have developed a prototype of the robot head, its control module, and the real-time face/eyes detection procedure; we are now working on capturing gaze-direction and target selection. The search/identify stage requires to evaluate objects' relevance to the context. This is because human gaze-monitoring is not so precise — though we have not done the evaluation of the precision — that they would rely on semantic and pragmatic clues like relevance.

4. Conclusion and Future Research

We outlined our on-going research on the role of shared attention. Human communication is an activity of sharing one's mental state with others. Shared attention plays an indispensable role in (1) selecting a target object relevant to the current context and (2) estimating others' mental states produced by the target object.

We are developing an attention-sharing system which can create and maintain shared visual attention with people in terms of monitoring their gaze-direction. We have achieved the real-time face/eyes detection mainly by a bottom-up approach; we found that capturing gaze-direction and selecting a target require a top-down approach, namely evaluation of objects' relevance to the current context.

Our short-term goal is to complete the gaze-monitoring process, evaluate it in human

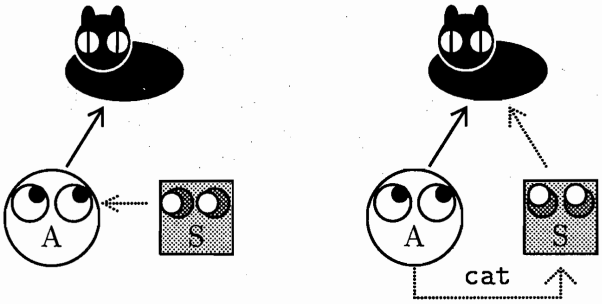


Fig. 5 Language acquisition by attention-sharing.

experiments, and elaborate our model of shared attention. Also we are planning an experiment on evaluating human gaze-monitoring precision. This will reveal how humans rely on top-down semantic and pragmatic expectations in gaze-monitoring.

We have two long-term goals. One is to incorporate the gaze-monitoring system into human-computer conversation systems. For this, the system has to extract linguistic context from the discourse in order to evaluate objects' relevance to the context. The other is to construct a model of infants' acquisition of the symbolic system of the first language. A symbolic system is a set of arbitrary associations between expressions and meaning; it articulates things and events in the world into categories and gives phonological representations to the categories. Attention-sharing with caretakers, as is illustrated in Fig. 5, will enable infants to learn associations between representations (e.g. caretakers' utterances) and meanings (e.g. estimated caretakers' mental states produced by target objects).

References

Baron-Cohen, S.: *Mindblindness: An Essay on Autism and Theory of Mind*, MIT Press, 1995.

Butterworth, G. and Jarrett, N.: What minds have in common in space: spatial mechanisms serving joint visual attention in infancy, *British Journal of Developmental Psychology*, Vol.9, pp.55-72, 1991.

Frith, U.: *Autism: Explaining the Enigma*, Blackwell, 1989.

Itakura, S.: An exploratory study of gaze-monitoring in nonhuman primates, *Japanese Psychological Research*, Vol.38, pp.174–180, 1996.

Sperber, D. and Wilson, D.: *Relevance: Communication and Cognition*, Blackwell, 1986.

Tanenhaus, M. K. and Spivey-Knowlton, M. J.: Eye-tracking, *Language and Cognitive Processes*, Vol.11, pp.584–588, 1996.

Chinese Word Segmentation and Part-of-Speech Tagging in One Step[*]

Tom B.Y. Lai, Maosong Sun, Benjamin K. Tsou, S. Caesar Lun
cttomlai@cityu.edu.hk, rlvt6@cityu.edu.hk, rlbtsou@cpccux0.cityu.edu.hk, ctslun@cityu.edu.hk
City University of Hong Kong

Abstract

In Chinese natural language processing, word segmentation and part-of-speech tagging is generally carried out as two separate steps. Earlier, the authors introduced a tag-based Markov-model approach to word segmentation. As the tags are of a syntactic nature, this is effectively doing word segmentation and part-of-speech tagging simultaneously. We have used a best-first algorithm with empirical results showing the search for the best solution to be efficient for inputs of reasonable length. In this paper, we will see that the job can be done using an $O(n^2)$ algorithm. In our experiments, we actually had the algorithm reduced to $O(n)$ by setting a maximum number of character for words in Chinese to a constant. We also show that performing word segmentation and part-of-speech tagging in one step will bring about improvement in accurracy.

1. Introduction

Chinese word segmentation (Chen 1992) can be done using a number of approaches (Liang 1987, He, 1991, Fan 1988, Sproat 1990 & 1996, Yeh 1991, Chang 1991, Lua 1994, Wu 1995) including the maximal-match principle, rule-based approaches and probability-based approaches. Lai 1991 suggests doing Chinese word segmentation by optimizing the product of successive tag bigram probabilities. The tags used (Sun 1992) are of a syntactic nature. Experiment results (Lai 1992) show that using the A* search algorithm is efficient for inputs of reasonable length (up to 30 characters). Bai 1995 uses tag-based bigrams to resolve ambiguities after segmenting the input. Sun 1995 uses mutual information instead of bigram probability.

[*] Supported by Research Grant #9040049, University Grant Council, Hong Kong

Markov-model approaches (Bahl 1983) have been used successfully in part-of-speech tagging (Marshall 1983, . DeRose 1988, Kupiec 1992, Chang 1993a).

In Chang 1993b, the best N outputs of a segementation module are passed to a tagging module. The two modules, operating sequentially, contribute to a score function that is used to yield the best segmentation and tagging scheme.

Lai 1991 and 1992 apply Markov Model techniques in Chinese word segmentation by using tags. As the tags are of a syntactic nature, this is effectively doing word segmentation and part-of-speech tagging at the same time. This is a genuine one-step approach. There is a well-understood linear-time dynamic programming algorithm for Markov-model-based approaches (Bahl 1983 and, e.g., DeRose 1988). However, the fact that a Chinese word can consist of a variable number of characters makes it impossible for this algorithm to be used in our approach. The A* search used in Lai 1992 is inefficient theoretically. But for inputs of less than 30 characters, space- and time-complexity are linear empirically. In this paper, we will see that genuine simultaneous word segmentation and part-of-speech tagging can nevertheless be done using an efficient dynamic programming algorithm.

2. Segmentation and part-of-speech tagging in one-step

2.1 When word segmentation and part-of-speech tagging are carried out one after another, errors in the two steps multiply. But if the two processes are integrated, then their interaction may help improve the combined accuracy. Consider:

dong1 ji4 shi4 cong2 tou2 nian2 yuan2 yue4 kai1 shi3 di2/de (1a)

shi4	cong2_tou2	nian2	yuan2_yue4	(1b)
V	Adv	TimeN	TimeN	
copula	start afresh	year	first month of year	
shi4	cong2	tou2_nian2	yuan2_yue4	(1c)
V	P	TimeN	TimeN	
copula	from	last year	first month of year	

Input (1a) may be segmented either into (1b) or (1c). If segmentation is carried out independently before part-of-speech tagging, (1b) will probably be preferred as tou2_nian2 in (1c) is rather infrequent in Chinese text. The final tag sequence of V-Adv-TimeN-TimeN, though rather unlikely by itself, will be produced. On the other hand, if part-of-speech tagging is carried out at the same time as word segmentation, then the fact that the tag sequence V-P-

TimeN-TimeN is more likely may be enough to offset the balance to allow the correct segmentation scheme (1c) to come out as the winner.

2.2 Word segmentation and part-of-speech tagging can be carried out simultaneously using a tag-based Markov model (Lai 1992). Consider:

yu3 zhong1 guo2 you3 guan1 lian2 (2)
yu3 / zhong1 guo2 / you3 guan1 / lian2 (2a)
yu3 / zhong1 guo2 / you3 / guan1 lian2 (2b)
yu3 / zhong1 guo2 / you3 / guan1 / lian2 (2c)
yu3 / zhong1 / guo2 you3 / guan1 lian2 (2d)
yu3 / zhong1 / guo2 you3 / guan1 / lian2 (2e)
yu3 / zhong1 / guo2 / you3 guan1 / lian2 (2f)
yu3 / zhong1 / guo2 / you3 / guan1 / lian2 (2g)

Input sentence (2) can be segmented into 2(a) to (2g). Words in (2a) to (2g) above may have the following tags: tag(yu3) = {jom, pom}, tag(zhong1 guo2) = {spd}, tag(zhong1) = {fom, spm, vnm}, tag(you3 guan1) = {qd, vnd}, tag(you3) = {vy}, tag(lian2) = {vnm, cnr, bom, pom}, tag(guo2 you3) = {aod}, tag(guo2) = {nam}, tag(guan1 lian2) = {vnd}, tag(guan1) = {vnm, ncm}.

As a word can have more than one tag, each segmentation scheme in (2a) to (2g) will correspond to a number tag sequences. The correct segmentation (2b), corresponding to tag sequence

pom /spd / vy / vnd (2b*),

is found by maximizing the product of successive tag bigrams. (Lai 1992 for details.)

A closer look at the tags reveals that they are essentially of a syntactic nature. For example, *pom* and *spd* in (2b*) are monosyllabic preposition and poly-syllabic proper noun respectively. With the correctly segmentated character string (2b), we also obtain the part-of-speech tagging information in (2b*). We are thus effectively performing segmentation and part-of-speech tagging at the same time.

One problem with this approach is that syntactic class information has to be encoded in the lexicon. This is expensive in terms of resources. However, it should be noted that such information is required for part-of-speech tagging anyway.

Another problem is computational efficiency. The well-understood linear-time algorithm for Markov-model based part-of-speech tagging (e.g. DeRose 1988) cannot be used. The best-first search algorithm in Lai 1992 is exponential, though experimental results show that it is efficient in practical situations (with the input containing up to 30 characters). Addressing this issue, we have designed an O(n^2) dynamic programming algorithm (described elsewhere) for finding the best segmentation-tagging alternative. By setting the maximum number of characters in a word to a constant, this algorithm can be further reduced to O(n).

3. Experimental results

Using a 486 PC (66 Mhertz, 8 M), we have performed experiments (using bigrams) on 729 sentences with a total of 10734 character tokens. The O(n^2) algorithm was reduced to O(n) by setting the maximum number of characters per word to 7. For the sake of clarity, define:

FP: all segmentation possibilities taken into account, each of which further expanded to a corresponding tag lattice;

MM: maximal-match used, so only one segmentation candidate and one tag lattice;

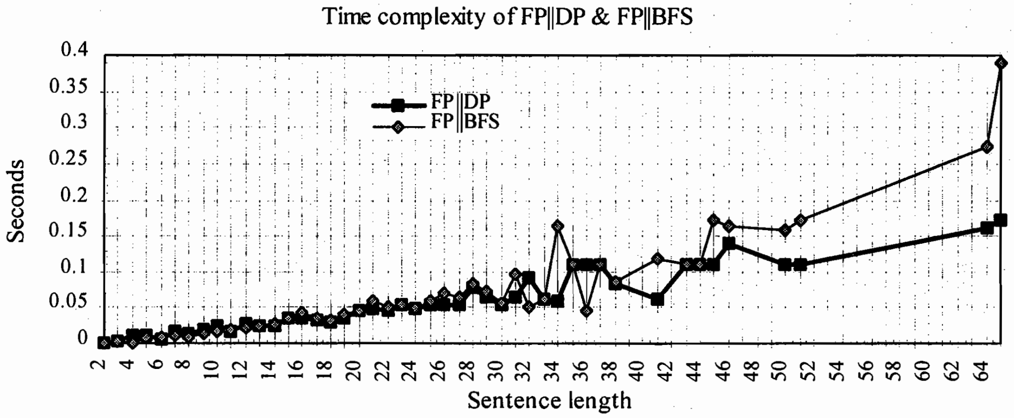
DP: dynamic programming used to find the most likely path through the tag lattice;

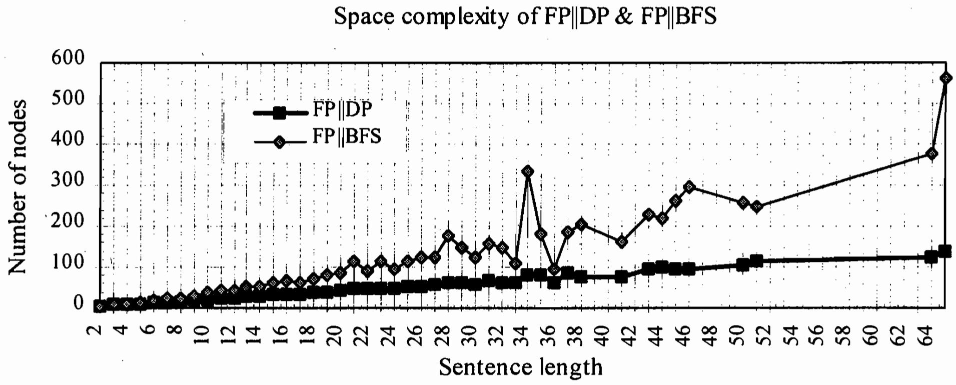
BFS: best-first search used to find the most likely path through the tag lattice;

x||y: procedures x and y carried out simultaneously;

x+y: procedures x and y carried out successively

3.1 Comparing FP||DP and FP||BFS





Sentences: 729 Character tokens: 10734

	Total time (seconds.)	Average time per character (seconds)	Total no. of nodes	Average no. of nodes per character	Total no. of arcs	Average no. of arcs per character
FP||DP	22.74	0.00	22520	2.10	21791	2.03
FP||BFS	23.24	0.00	44766	4.17	44037	4.10

(1) the time and space graphs for FP||DP are approximately linear;

(2) the number of nodes/arcs created by FP||DP is about half of that created by FP||BFS;

(3) time-efficiency improvement is signifcant for input more than 30 characters long.

3.2 Comparing FP||DP and MM+DP.

While the efficiency (and linearity) of our algorithm is established above, it is to be expected that finding the best segmentation-tagging acheme in a one step involves a larger search space than finding the best segmentation alternative and the best tagging scheme thereof in two separate steps. MM+DP, for example, should be less expensive than FP||DP. However, our results show that FP||DP does not compare too unfavourably with MM+DP.

Sentences: 729 Character tokens: 10734

	Total time (seconds.)	Average time per character (seconds)	Total no. of nodes	Average no. of nodes per character	Total no. of arcs	Average no. of arcs per character
FP||DP	22.74	0.00	22520	2.10	21791	2.03
MM+DP	9.70	0.00	10617	0.99	9888	0.92

FP||DP is just a little more than twice more expensive than MM+DP in terms of both space and time. If MM were replaced with a procedure that returned more than one segmentation scheme for the DP tagging component to work on, the combined procedure would also be more expensive than MM+DP. FP||DP is thus efficient compared to sequentially combined segmentation and tagging.

3.3 Effectiveness of FP||DP: accuracy (precision) improvements

	Correctly segmented sentence+DP	MM+DP	FP||DP
Accuracy of word segmentation (A)	100.00%	98.35%	99.66%
Accuracy of POS tagging (B)	95.06%	93.13%	94.66%
Estimation of the total performance(A*B)	95.06%	91.65%	94.34%

The left-most column gives the upper bounds (no segmentation errors). Compared with MM+DP, FP||DP has a 1.31% improvement in segmentation, a 1.47% improvement in POS tagging, and a 2.69% improvement in the combined process. This shows that doing segmentation and part-of-speech tagging simultaneously is indeed more effective than performing the two tasks in two separate steps.

4. Conclusion

We have shown that segmenting a sentence and marking parts of speech of the words identified simultaneously will improve both segmentation accuracy and part-of-speech tagging accuracy. Using our tag-based Markov-model approach, this can be done effectively, with an $O(n^2)$ algorithm.

References

Bahl, L.R., F. Jelinek and R.L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol *PAMI*-5, No. 2, March 1993, pp. 179-190.

Bai, S.H., "An Integrated Model of Chinese Word Segmentation and Part-of-Speech Tagging (in Chinese)," in *Advances and Applications on Computational Linguistics (Selected Papers* *from the 3rd National Conference on Computational Linguistics, Shanghai, Nov. 5-7. 1995)*, Tsinghua University Press, 1995, pp. 56-61.

Chen, K.J. and S.H. Liu, "Word Identification for Mandarin Chinese Sentences," *COLING9-92*, Nantes, 23-28 Aug., 1992, pp. 101-107.

Chang, C.H. and C.D. Chen, "HMM-based Part-of-Speech Tagging for Chinese Corpora," *Proc. Workshop on Very Large Corpora: Academic and Industrial Perspectives*, Columbus, Ohio, June 1993, pp. 40-47. (1993a)

Chang, C.H. and C.D. Chen, "A Study on Integrating Chinese Word Segmentation and Part-of-Speech Tagging," *Communictions of COLIPS*, Vol. 3, No. 1, 1993, pp. 69-77. (1993b)

Chang, J.S., J,I. Chang and S.D. Chen, "A Method of Contstraint Satisfaction and Statistical Optimization for Chinese Word Segmentation," *Proc. 1991 ROCLING*, Kenting, Taiwan, August 1991.

DeRose, S.J., "Grammatical Category Disambiguation by Statistical Optimization," *Computational Linguistics*, Vol. 14, No. 1, Winter 1988, pp. 31-39.

Fan, C.K. and W.H. Tsai, "Automatic Word Identification in Chinese Sentences by the Relaxation Technigue," *Computer Processing of Chinese and Oriental Languages*, Vol.4, No.1, November 1988, pp. 33-56.

He, K.K., Xu, H. and B. Sun, "Design Principles of a Expert System for Word Segmentation for Written Chinese Text (in Chinese)," *Journal of Chinese Information Processing*, 5-2, 1991.

Lai, T.B.Y., S.C. Lun, C. F. Sun and M.S. Sun, "A Maximal Match Chinese Text Segmentation Algorithm Using Mainly Tags for Resolution of Ambiguities (in Chinese)." *Proc. of ROC Computational Linguistics Conference*, Kenting, Taiwan, August 1991, pp. 135-146.

Lai, T.B.Y., S.C. Lun, C.F. Sun and M.S. Sun, "A Tagging-based First-Order Markov Model Approach to Automatic Word Identification for Chinese Sentences," *Proc. 1992 International Conference on Computer Processing of Chinese and Oriental Languages*, Tampas, Fl, 15-18 Dec., 1992, pp. 17-23.

Liang, N.Y, "Automatic Segmentation of Chinese Words and the Related Theory." Proc. 1987 *International Conference on Chinese Information Processing*, Beijing, 1987, pp. 454-9.

Lua, K.T. and G.W. Gan, "An Application of Information Theory in Chinese Word

Segmentation," *Computer Processing of Chinese & Oriental Langauges*, Vol. 8, No.1, June 1994, pp. 115-124.

Marshall, I., "Choice of Grammatical Word-Class Without Global Syntactic Analysis: Tagging Words in the LOB Corpus." *Computers in the Humanities*, Vol. 17, 1983, pp. 139-150.

Sproat, R. and C. Shih, "A Statistical Method for Finding Word Boundaries in Chinese Text," *Computer Processing of Chinese and Oriental Languages*, Vol. 4, No. 4, March 1990, pp. 336-351.

Sproat, R., C. Shih, W. Gale and N. Chang, "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese," *Computational Linguistics*, 22(3), 1996, pp. 377-404.

Sun, M.S., T.B.Y. Lai, S.C. Lun and C.F. Sun. "A Tagset for Automatic Chinese Text Segmentation," First International Conference on Chinese Linguistics, Singapore, June 1992. Printed in *Working Papers in Languages and Linguistics*, No. 5, City University of Hong Kong, April 1993, pp. 127-134.

Sun, M.S., T.B.Y. Lai, T.B.Y., S.C. Lun and C.F. Sun, "Some Issues in the Statistical Approach to Chinese Word Indentification," *Proc. 1992 International Conference on Chinese Information Processing*, Oct. 26-28, 1992, Beijing, pp. 246-253.

Sun, M.S. and B.K. Tsou, "Ambiguity Resolution in Chinese Word Segmentation," *Proceedings of the 10th Pacific Asia Conference*, Dec. 27-28, 1995, Hong Kong, pp. 121-126.

Wu, D.K, "Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Parallel Corpora," *Proceedings of IJCA-95 (Fourteenth International Joint Conference on Artificial Intelligence)*, Montreal, 1995, pp. 1328-1334.

Yeh, C.L. and H.J. Lee, "Rule-based Word Identification for Mandarin Chinese Sentences - A Unification Approach," *Computer Processing of Chinese and Oriental Langauges*, Vol 5, No. 2, March 1991, pp. 97-118.

Corpus-Based Chinese Text Summarization System

Jun-Jie Li and Key-Sun Choi
CSLab, Center for AI Research, Korea Advanced Institute of Science and Technology
Taejon, Republic of Korea
Tel: +82-42-869-5565, Fax:+82-42-869-8700
E-mail:{jklee,kschoi}@world.kaist.ac.kr

Abstract

A Chinese Text Summarization system is developed, which is based on the surface information of context as well as the corpus based word segmentation and keyword identification. Unknown words identification is the most difficult topic on Chinese Word Segmentation. The context information is utilized here to resolve the unknown words and ambiguous segmentation problem by integrating word frequency and word length to dynamically weight the word weight, the theory and experiments show that this approach is superior than traditional dictionary based matching approach and pure word frequency-based statistical approach. The segmentation precision is 98% for real text. The keyword identification is not only based on word frequency but also word length, salient sentence determination is solved by using word weights, sentence length, number of clauses, numeric word and unknown words etc.., less relying on sentence position and surface cues. The evaluation measures of summary is studied and experimental results are provided.

1. Introduction

Text Summarization System is to identify and select the central content or user inquired content from the given original texts to form the summarized output with the sentences identical to the original input text or new generated. There are three kinds of approaches on developing Text Summarization System: the first one is based on the surface-clues of the current context such as the word frequency (Luhn, 1958), sentence position, word clue or indication , title sentence(Watanabe,1996), word association or rhetorical relations(Ono et al.,1994) and linear heuristic sentence weighting function (Zechner, 1996).Its advantages are simple and domain unconstrained, its shortcoming is inaccuracy in sentence abstracting due to the uncertain valve of word frequency for key words, varied distribution of important sentences and heuristic function itself. The second one is based on the knowledge-based natural language processing techniques, such as Script-based summarization system for given texts with multilingual output(Tait, 1985), CD-based domain constrained abstracting system with incomplete syntactic and semantic analysis (Dejong, 1979), rule-based summarization system with forward and backward scanning schema (Danilo,1982) etc.. Its advantages are more accurate and in depth language analysis and generation. Its shortcomings are domain constrained and difficulty in knowledge base maintenance. The third one is the corpus based methods (Li, 1995; Li and Choi, 1997). The corpus based sentence segmentation, non-linear sentence weighting function , collocation computation based word and sentence importance analysis and efficient raw corpus and text indexing method, give this method a prospective future.

In section 2, a full text indexing method called Natural Hierarchical Network (NHN) is illustrated. In section 3, the word segmentation algorithm is introduced, the text summarization system is illustrated in 4, the experimental results are given in the section 5, and finally the conclusion is given in section 6.

2. Natural Hierarchical Network

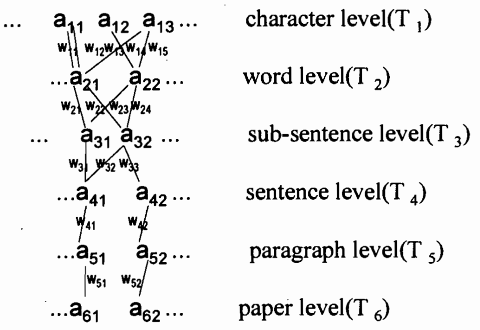


Fig.1 Description of Natural Hierarchical Network

In ordcr to make full use of context and text corpus information such as character and/or word frequency and collocation.We design a new full text indexing method, called Natural Hierarchical Network (NHN) shown as Fig. 1.

The meaning of NHN is that every language unit(character,word, sentence, paragraph) have a vector to corresponds to its every occurrence in texts, and in turn, the texts(or raw corpus) can be indexed and represented by all the occurrences of elements in a certain level m (say character level) represented by the vectors as above. Certainly, we can omit some levels to make the vector shorter, however that will lose some useful information of text structure and language usage.

In practice, according to the sentence and sub-sentence ending symbols(i. e., punctuation such as . , : ? !) and the format and markers of text(such as writing rules or custom of paragraph, chapter , title and subtitle), the input text can be automatically converted into a series of vectors as (pp, pa, sn, ss, wd, ct, c_i), where pp, pa, sn, ss, wd and ct are respectively represent the sequential numbers of paper, paragraph, sentence, sub-sentence, word and character that character c_i appears in.

3. Word Segmentation

Chinese Word Segmentation(Li, 1995; Chen and Lee, 1996) is an ever-green topic due to the unknown word identification and ambiguity resolution problem. We provides an dynamic word weighting function which calculates the frequencies of words(exactly speaking, strings) in context(or textual corpus) in the run time, segmentation algorithm is designed as follows:

```
Algorithm Segment(s)
{
/* given a string s(initially is the whole sentence), segment it into some words by greedy algorithm.*/
  Step 1. Computing weights of all the sub-strings in s.
  Step 2. Pick up the string with the greatest weight, say s, which is the current
      abstracted word and store it. If s equals to s ,then exit, otherwise go to step 3.
  Step 3. Segment(s-s),s-s is the left strings.
}
Algorithm Weighting(S)
{
/*Given a input string S= $c_1 c_2 \cdots c_n$, in order to compute weights of all strings in s, it is necessary to build a collocation matrix A, where A(i, j) represents the frequency of string from character i to j, then the weight of that string can be calculated by the weighting function $W(c_i c_{i+1} \cdots c_j) = F(c_i c_{i+1} \cdots c_j) \times (j-i+1)^c$, where $F(c_i c_{i+1} \cdots c_j)$ is the frequency of string $c_i c_{i+1} \cdots c_j$ , and its length is (j-i+1), c is a constant power of length, c>1. In practice when c equals 3, the segmented words are more probable to be correct;*/
  Step 1. Search the data base to find out NHN set $T_i$ of each $c_i$,  i= 1,..n.
Step 2. For (j=1;j<=n-1;j++)
Step 3. For (i=1;i<=j-1;i++){
Step 4.  $T_{ij}= T_{i,j-1} \wedge T_j$ ;// to compute collocation of column j in matrix A
Step 5.  A(i,j-1) = $W( c_i c_{i+1}\cdots c_j)$ = $| T_{i,j-1}| \times( j-i)^c$; //to weight $c_i c_{i+1}\cdots c_{j-1}$.
}
```

where, T_{ij} is the NHN set of $c_i c_{i+1}\cdots c_j$, $T_{ij} = T_{i,j-1} \wedge T_j = ((((T_i \wedge T_{i+1}) \wedge T_{i+2} \wedge ...) \wedge T_j$ "$\wedge$" means collocation computation. For example,

let $(pp_1, pa_1, sn_1, ss_1, wd_1, ct_1) \in T_i$, $(pp_2, pa_2, sn_2, ss_2, wd_2, ct_2) \in T_{i+1}$, if$((pp_1= pp_2)and(pa_1= pa_2)and(sn_1 = sn_2)and(ss_1= ss_2)and(wd_1= wd_2)$ and$(ct_1+1= ct_2))$, then it means that c_i and c_{i+1} collocate once with c_i appearing to the left side of c_{i+1}, let $(pp_2, pa_2, sn_2, ss_2, wd_2, ct_2) \in T_i \wedge T_{i+1} .= T_{i,i+1}$

In practice, multiple segmentation technique is utilized. The first scanning of segmentation is to identify one character words , besides the numeric words(such as 1,2,三) and count unit words(such as 個) are also preprocessed by utilizing a numeric word list and a unit word list as well as matching based segmentation. Therefore in the first segmentation, the dictionary based approach is used based on a very small function word dictionary instead of a very big and complicated dictionary.

The segmentation on the second time is continue to process the unidentified strings based on the computation of the string frequency within context to find the frequently occurred(e. g. 2 or 3 times occurred) unknown words and solving ambiguous segmentation.

On the third time, segmentation is based on the string frequencies calculated in corpus to segment common words and make some low frequency unknown words isolated because, generally, the two words on the left and right of the unknown word is most likely to be common words which can be identified.

5. Text Summarization

5.1 Key Words Identification

A more efficient word weighting function is developed which is based on word frequency and word length, where the word frequency is refer to the frequencies calculated both in context and corpus, because key words is context related, the importance of context information and its utilization should be studied, besides the word length information is highlighted in weighting words, because key word is generally proper nouns which have a longer length than function words and other unimportant shorter content words, therefore longer words should be assigned higher weight, however pure frequency based method can not do that. The word weighting function is designed as follows:

$$T(w)=\frac{F_1(w)}{F_2(w)} \times L(w)^c$$

where $F_1(w)$ is the frequency of w in context, $F_2(w)$ is the frequency of w in corpus, $L(w)$ is the length of w , c is a constant power of length, in practice c=3.

5.2 Sentence Weighting Function

The important sentences generally illustrate themes or topics of contexts in a condensed and conclusive way, such as title and subtitle sentences, topic sentences and other conclusive sentences. The characteristics of important sentences are generally to contain more important words (or key words) and have a shorter sentence length and few number of sub-sentences. Therefore, the sentence weighting function is designed as follows:

$$P(s)=\frac{T(w_1)+T(w_2)+\cdots+T(w_n)}{K\times L(s)\times N(s)}$$

where s is a sentence and w_i is a word of s, $T(w_i)$, i=1,..n, is the weight of w_i, $L(s)$ is the length of sentence s, $N(s)$ is the number of sub-sentences in s.

According to this function the shorter sentences with more important words will be given higher weight, so title and subtitle sentences, topic sentences and most of the conclusive sentences will have more chance to obtain higher weights than the other unimportant sentences.

Traditional approaches often use sentence position such as first and last sentence of a paragraph will statically assign a higher weight, however this assuption is not always true and salient sentences will often appeared in the middle of the paragraph and will not properly weighted. Besides, surface cues such as conjunctoins are often used to identify important sentence as well as rhetorical structures or referential links, which is powerful in analyzing inter- and intro sentence relations and guiding the salient sentence selection. However, the lack of /or superficial/or wrongly used surface cues often effect or mislead the rhetorical analysis, it should be noted that different styles of text such as literiture, editorial, technical paper, poetry and so on rely on and use surface cues in different weight and way.

There are also other interesting factors such as digital numbers(to answer how many/how much), time words(to answer when) unknown words such as name of people, organizations(to answer who/whom), place(to answer where) and this kind of factors is often user- oriented and text style related, and if used properly, it will make good effect. For example, if the user not only concern about the central content of the text but also concern the time or people involved, then the time and name of people should be designed to give a higher weight so that the sentences contain those words will have chance to be assigned higher weight.

5.3 Abstract Generation and Evaluation

The abstracts(or summaries) are generated by selecting the important sentences with higher sentence weight from the input text, and keeping their original sequential orders in the text.

It is not a settled problem on evaluating the quality of summarization. There are several criterias including recall, precision, brevity and ease in general, where recall means the percentage of central concepts or important sentences captured by an automatically produced summary, precision refers to the percentage of relevent concepts or sentences in this summary. Besides the above four criterias, there are other useful criterias such as Wh(when, where, what, why)&how(how, how many/how much), those criteria is useful when user are particularly interested in those contents or they acturally are the central contents of the text.

5. Experiments

5.1 Word Segmentation

The corpus used for word segmentation is a collection of texts without restrictions on domain, style and length.

A segmentation tests under 290,000-character sized textual corpus shows that the correctness rate of identified words ,denoted as PI, which is calculated by $PI=1-\frac{b}{w}$, where b is the number of wrong segmented words and w is the number of total words, is about 98% .

An comparison of 30000 words dictionary based Backwards Maximum Matching(BMM) algorithm and our textual corpus based segmentation algorithm, shows that the number of wrong segmented words of BMM is 4 times of that of our method for open test.

5.2 Text Summarization

We have tested more than 50 texts in different domains and styles including editorials, technical papers, literiture papers, pose and news etc.. The results show that the better summarization results are generally domain unconstrained but influnced by writing styles. For example, if repitition is the main techniques to express emphasis and central concepts, then those kind of texts including editorials, technical papers and news will summarized properly, while pose or literiture texts will have worse summarization results.

6. Conclusion

we have illustrated the detailed algorithms of Chinese Text summarization system and shown some experimental results. The key techniques include the NHN(Natural Hierarchical Network) raw corpus(and text) indexing method, the word length and frequency based word weighting function and word segmentation algorithm. Keyword identification and salient sentence determination. The algorithms and ideas of our system are also quite meaningful for Korean and Japanese unknown words identification, and have been applied in Corpus-based Chinese -Korean Machine Translation(Li and Choi,1997a), Chinese Automatic Abstracting System(Li, 1995), Chinese-Korean Automatic Translation System(Li and Choi,1997b).

Acknowledgment

This research was supported by China National High -Tech Development Plan 863 and Center for Artificial Intelligence Research in Korea Advanced Institute of Science and Technology.

References

Jun-Jie Li and Kai-Zhu Wang, "Study and Implementation of Non-dictionary Chinese Segmentation," in NLPRS'95, Seoul, Korea, Dec.,1995, pp.266-271.

Hsin-Hsi Chen and Jen-Chang Lee, " Identification and Classification of Proper Nouns in Chinese Texts," in COLING'96, Aug., 1996, Vol.1 pp.222-229.

Jun-Jie Li and Key-Sun Choi, "Design and Implementation of An Example-Based Chinese-Korean Machine Translation System," in ICCPOL'97, Apr. 1997,Hong Kong.

Jun-Jie Li and Key-Sun Choi, "Corpus-Based Chinese-Korean Abstracting Translation System," in IJCAI'97, Nagoya, Japan, Aug. 1997.

Hideo Watanabe, " A Method for Abstracting Newspaper Articles by Using Surface Clues," in COLING96:947-979, Copenhagen, Denmark, Aug. 5-9, 1996.

Klaus Zechner, "Fast Generation of Abstracts from General Domain Text Corpora By Extracting Relevant Sentences," in COLING96:986-989, Copenhagen, Denmark, Aug.5-9, 1996.

Luhn,H.P., "The Automatic Creation of Literature Abstracts," IBM Journal of Research and Development, Vol.2, No.2:159-165.

G. Dejong, "Prediction and Substantiation : Two Processes That Comprise Understanding," in Proceedings of IJCAI-79.

J.I. Tait, "Generating Summaries Using a Script-Based Language Analyzer," in Progress of Artificial Intelligence, 1985.

Danilo Fum. et al., " Forward and Backward Reasoning in Automatic Abstracting," COLING82.

Ono et al., "Abstract Generation Based on Rhetorical Structure Extraction," in COLING94, Vol.1:344-348.

A Study on the Portability of a Grammatical Inference System

Hsue-Hueh Shih[1] and Steve Young

Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, England

1 Abstract

This paper presents a study on the portability of our grammatical inference system called CAGC (Computer Assisted Grammar Construction). The CAGC system has been developed [1] to generate broad-coverage grammars for large natural language corpora. It utilises both an extended Inside-Outside algorithm [2] and an automatic phrase bracketing (AUTO) technique [3], which is designed to provide the extended algorithm with constituent information during learning. The system is firstly trained and tested on the Wall Street Journal (WSJ) corpus, and then ,for the study of its portability, it is moved onto the Brown Corpus to infer a Brown grammar. The experimental results shown in this paper demonstrate that the CAGC inference technique as well as the initial grammar used in the system are transferable to the new corpus.

2 Introduction to Grammatical Inference

Grammar is a crucial component in most natural language processing systems because it bounds the range of constructions which can be handled. However, the conventional method of manual grammar construction is labour intensive, time consuming and often leads to errors caused by unwanted rule interactions. In addition, manually-developed grammars often rely on the assumption that all input sentences are well-formed. Consequently, these grammars have limited coverage on naturally occurring corpora. To go beyond this traditional approach, more practical and robust techniques for grammar construction become necessary.

With the increasing availability of large naturally occurring text corpora in machine readable form, it has become possible to infer linguistic knowledge directly from regularities that appear in sentence samples. The application of techniques for inferring syntactic information in such a way is termed Grammatical Inference (GI). Among recently developed GI techniques, the Inside-Outside algorithm [4] shows potential for the inference of stochastic context-free grammars. However, its practical use in Natural Language Processing is limited by both its high computational complexity and there being no guarantee of convergence to a local optimum which is linguistically motivated.

Recent improvements to this technique have included supervised training [2] to accelerate the inference process and the use of an Explicit-Implicit technique [5] employing a hybrid initial grammar to bias the inference process towards linguistically meaningful solutions. Nevertheless, supervised training re-introduces the problem of labour intensiveness, since the required treebank must be manually annotated. Alternatives must be sought to alleviate this manual load as well as provide useful constituent information for training. The CAGC system, whose portability is examined and will be shown later in this paper, integrates a heuristic-based surface bracketing with the Explicit-Implicit technique to complement the inference process.

[1] Hsue-Hueh Shih now works in the Department of Foreign Languages and Literature, National Sun Yat-sen University, Taiwan. E-mail: hsuehueh@mail.nsysu.edu.tw

The Overview of the CAGC system is given in the next section, which is followed by a system evaluation on the WSJ in Section 4 and the portability study using Brown Corpus in Section 5. Conclusions are drawn in Section 6.

3 Overview of the CAGC system

The CAGC system takes advantages of both heuristic and stochastic approaches. Heuristic knowledge provides powerful and important constraints to the system, whereas stochastic information deals with situations which are too complex or too trivial for heuristic rules to handle. A block diagram of the system is shown in Figure 1.

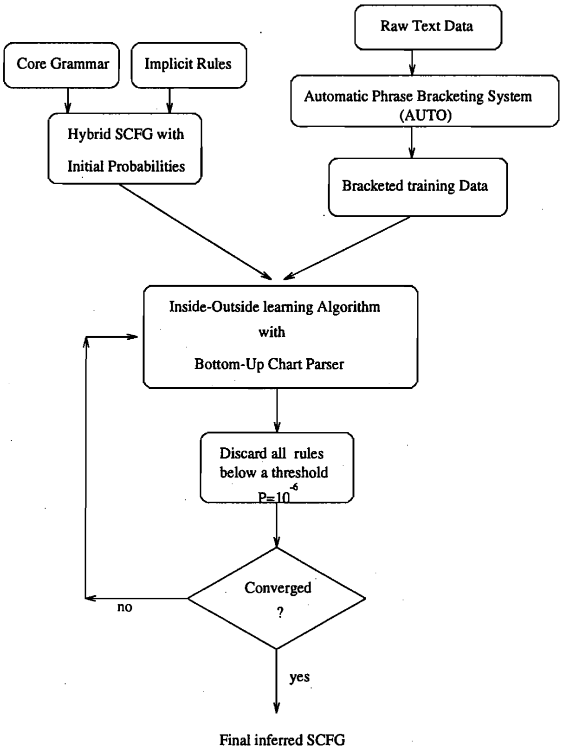

Raw Text Data
Core Grammar
Implicit Rules
Automatic Phrase Bracketing System (AUTO)
Hybrid SCFG with Initial Probabilities
Bracketed training Data
Inside-Outside learning Algorithm with Bottom-Up Chart Parser
Discard all rules below a threshold P=10^-6
Converged ?
no
yes
Final inferred SCFG

Figure 1: A Block Diagram of the CAGC System

The first part of the system falls into two stages: construction of an initial SCFG and phrase-bracketing of the raw text data. In the second part of the system, a grammar is inferred by utilising the Inside-Outside algorithm to re-estimate the initial SCFG from the bracketed text data. The initial SCFG is derived from the hand-written core grammar (explicit part), which forms a skeleton of the SCFG, and a set of CF rules (implicit part) which consists of all possible rules that do not appear in the core grammar but are nevertheless linguistically plausible. The explicit and implicit rules are then integrated into a hybrid SCFG along with an appropriate set of initial probabilities. Details of the grammar development and the calculation of the initial probabilities are described in [3]. The AUTO bracketing technique utilises heuristic knowledge to bracket the raw text data in a way which integrates top-down and bottom-up approaches. The training set augmented by this derived constituent information provides the additional constraints to the grammar re-estimation process in the second part of the CAGC system.

In the second part of the CAGC system, the Inside-Outside learning procedure, incorporating a bottom-up chart parser [6], iteratively re-estimates the probabilities of the production rules. The updated probabilities are calculated according to the weighted frequency counts of the rules used in parses licenced by the grammar and generated at the previous iteration. At the end of each iteration, the rules with probabilities falling below a pre-defined threshold are discarded. The re-estimation process continues until either the change in the total log probability between iterations is less than a minimum or the number of iterations reaches a maximum. The final inferred grammar is generated when either criteria is met.

4 System Evaluation on the WSJ Corpus

1500 training and 500 test data were chosen from the Wall Street Journal(WSJ) text corpus. There is no explicit limitation on their length, and the average length of data sentences is around 13 words. Instead of lexical entries, parts-of-speech (POSs) are used in our experiments to reduce computation. Original 48 WSJ POSs were manually subcategorized into 59 in order to capture more detailed syntactic information. Detailed subcategorization is stated in [3].

Table 1 shows the performance of the inferred WSJ grammar, when compared with an inferred grammar supervised by Penn treebank. It records the number of SCF rules which survived after training, the number of test sentences which can be parsed by the grammar, and the performance on three metrics that are often used to evaluate NLP systems [7]. Recall is the percentage of standard bracketings (in Penn treebank) present in our experimental output of the same sentence. This metric indicates the closeness between the evaluated grammar and the Penn treebank. Precision is the percentage of the bracketings in our output present in the Penn treebank sentence. A crossing error is defined as the partial overlap between a bracket pair (one generated from our experiment and the other from the treebank) and Crossings are the average number of crossing errors in a sentence.

Grammar Types	**PENN_Trained**	**AUTO_Trained**
Rules Remaining After Training	21.29% (6029/14736)	18.54% (2733/14736)
Sent. Parsed	97.80% (489)	97.20% (486)
Recall	84.65%	84.66%
Precision	64.06%	62.50%
Crossings	1.92	2.14

Table 1: Performance of WSJ Grammars Trained on PENN treebank or AUTO-bracketed data

Figures in Table 1 demonstrate that the CAGC inference technique is able to generate a high coverage grammar with good accuracy in phrase bracketing, and AUTO is capable of providing useful and competitive bracketing information during training phase.

As the data used in the experiment were manually tagged, it is desirable to integrate an automatic tagger into the CAGC system, so that the system no longer requires any pre-tagged data for its training. For this reason, the Acquilex tagger [8] was trained on a subset of the WSJ corpus, and then integrated into the CAGC system as a front-end. Table 2 shows the performance of the CAGC system using Acquilex-tagged data. Note that this experiment was carried out on increased training (4000) and test (1500) sets, which results in a better performance, when compared with the corresponding figures in Table 1 before the tagger is employed. From Table 2, one can see that the performance of the inferred grammar degrades as the tagger is introduced into the system. This degradation is due to the 7% error rate of the tagger.

System	Manually-tagged	Acquilex-tagged
Recall	86.56%	83.06%
Precision	64.25%	61.79%
Crossings	1.93	2.31

Table 2: The CAGC System Performance Using a Tagger as the Front-end

5 Portability Evaluation on the Brown Corpus

The portability of the CAGC system is investigated using the Brown corpus. Similar sizes of 4000 training and 1500 test data were collected for this experiment. These data are given consistent POSs by the Acquilex tagger. The hybrid initial grammar is directly transfered from the WSJ task. The CAGC system re-estimates the parameters of the grammar iteratively, according to the Brown training data which is AUTO bracketed in advance. The final inferred Brown grammar is generated and then used to analyse the test data. Table 3 shows the performance of the inferred Brown grammar when compared with that in the WSJ task.

Inferred Grammar	WSJ	Brown
Recall	83.06%	79.04%
Precision	61.79%	57.64%
Crossings	2.31	3.10

Table 3: Performance of the Inferred Brown Grammar on 1500 Test data

As can be seen, the overall performance on the three metrics degrades in the Brown task. Recall and Precision are both down 4%, whereas Crossings increase to 3 errors for a sentence. In order to account for this degradation, two additional experiments on the accuracy of Acquilex and AUTO are carried out (the details of these experiments can be seen in [9]). The first experiment on the tagging performance of the Acquilex tagger shows that the tagging accuracy decreases from 93% in the WSJ to 91% in the Brown tasks. This is because the tagger was trained on WSJ, and therefore the proportion of the unknown words to the tagger was larger in the Brown data. This situation can be easily improved by using a larger set of data from different copora as training material for the tagger.

The second experiment on the bracketing accuracy of AUTO shows there is a 6% decrease in both Recall and Precision metrics and Crossing errors increases 0.6 for a sentence. As AUTO works on the POS sequences, it is believed that this is caused partly by the decreasing accuracy of the tagger and partly by the fact that it is designed originally for the WSJ task. AUTO will need to be re-tuned to meet the requirement of task-independency.

From the experiments shown above, it is felt that the 4% decrease in the overall performance of the inferred grammar is mainly caused by the decreasing accuracy in both Acquilex and AUTO. Therefore, making them more task-independent becomes a key issue on improving the portability of the CAGC system. Nevertheless, the hybrid initial grammar is believed to be transferable to the new corpus, since its core part is designed to capture important general syntactic structures in English grammar and its implicit part will be shaped to target the corpus-dependent structures.

6 Conclusions

Portability is a significant issue and usually involves a large amount of manual work in most grammar-based systems. A grammar designed for one corpus may not properly apply to another corpus and ,therefore, modifying the grammar manually is often required when moving from one application to another. In this paper, the CAGC system shows its potential in alleviating this problem. From the experimental results shown, it is believed that inference technique and the initial hybrid grammar are transferable to the new corpus, and the portability of the system can be improved if two of the CAGC compoents, the Acquilex tagger and the AUTO phrase bracketing technique, are made more task-independent.

References

[1] H-H. Shih, S.J. Young, and N.P. Waegner. An inference approach to grammar construction. *Computer Speech and Language*, 9:235–256, 1995.

[2] F. Pereira and Y. Schabes. Inside-Outside re-estimation for partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, June 1992.

[3] H-H. Shih and S.J. Young. A system for computer assisted grammar construction. Technical Report TR.170, Engineering Department, Cambridge University, England, June 1994.

[4] J.K. Baker. Trainable grammar for speech recognition. In *Speech Communication Papers for the 97th Meeting of the acoustical Society of America (D. Klatt and J. Wolf, eds)*, pages 547–550, 1979.

[5] E. Briscoe and N. Waegner. Robust stochastic parsing using the inside-outside algorithm. In *AAAI Symposium on Statistic Applications to Natural Language*, June 1992.

[6] G. Gazdar and C. Mellish. *Natural Language Processing in PROLOG.* Addison-Wesley, 1989.

[7] H.S. Thompson. Parseval workshop. In *ELSNews Vol.1(2)*, 1992.

[8] D. Elworthly. Part-of-speech tagging and phrasal tagging. Technical Report Acquilex-II Working Paper 10, Computer Laboratory, Cambridge University, England, 1993.

[9] H-H. Shih. *Computer Assisted Grammar Construction.* PhD Thesis. Engineering Department, Cambridge University, England, 1995.

Fast Lexical Post-Processing on Cursive Script Recognition

Marco A. Torres, Susumu Kuroyanagi and Akira Iwata
Nagoya Institute of Technology
Department of Electrical and Computer Engineering[1]

Abstract

This paper presents a novel, fast approach to the problem of lexicon reduction. It is based on the concept of direct-addressing a table in which there is a slot of 1 bit for each component of 4 characters contained on the lexicon. In one of our experiments, the time needed for post-processing was reduced from $31sec$ to only $134ms$, without any loss on recognition accuracy. The structure proposed is very flexible, and can be used as a front end for contextual post-processing on Handwritten or Handprinted Recognition. It is ideal to be implemented on parallel computers or hardware.

1 Introduction

A Lexicon based text recognition system (handprinted or handwritten), receives as an input an image representing a word, and produces at the output the word on the lexicon that best matches that image (see fig. 1). It works as follows: The input image representing the word to be recognized is segmented and, features extracted from each segment are used by a Character Classifier to produce Character Candidates, which are combined to produce String Candidates, that are matched with a list of valid words (Lexicon); those strings not in the Lexicon are rejected (for example on fig. 1 the input image "test", produced 81 String Candidates, from which only two of them are valid words). Further processing could be applied to the remaining Word Candidates to select the most feasible to represent the input image on that particular context.

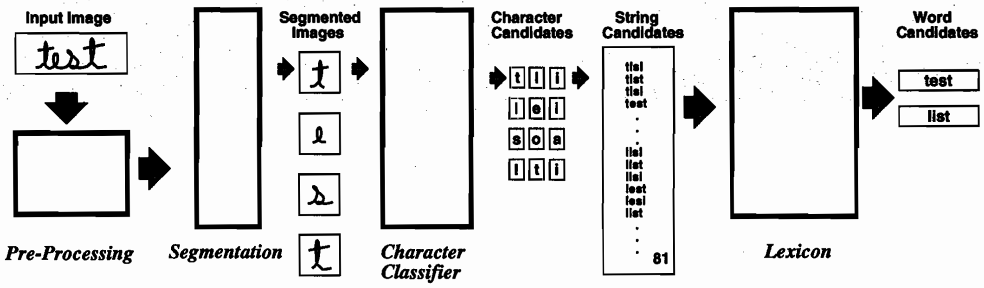


Figure 1: A Lexicon Based Text Recognition System

The time needed to analyze all the String Candidates (t_{lex}) is a function of the number of strings ($N_{strings}$) and the time needed to search in the lexicon structure for each individual string (t_{search}), $t_{lex} = t_{search} \times N_{strings}$. On the other hand, $N_{strings}$ is a function of the number of segments ($N_{segments}$) in which the input image was divided and, the

[1] Gokiso-cho, Showa-ku, Nagoya 466, Japan. e-mail: marco@mars.elcom.nitech.ac.jp

number of character candidates (N_{chars}) that are considered, $N_{strings} = (N_{chars})^{N_{segments}}$. While the list of String Candidates can be very large, it is mostly composed by not valid combinations of characters. This is specially important on script recognition where different segmentation points can be proposed and more String Candidates are produced. Then, while fast search methods are highly desirable, the most important point for efficient lexical post-processing systems is their ability to rapidly discard those not valid combinations of characters. This process is called **lexicon reduction** or **lexicon filtering**[1].

This paper examines the effectiveness of a new, fast method that reduces drastically the post-processing time by reducing the number of String Candidates. The Lexicon Reduction Method is based on pre-processing the Character Candidates before creating the actual list of String Candidates. To do this, a fast method using look-up tables (we called them Prune Tables) is proposed.

2 Description of Method

Two main goals must be considered on designing an efficient lexicon reducer: i) speed is a critical issue, especially for large lexicons, and ii) a very high percentage of the input strings must be discarded [1].

The list of String Candidates is obtained by combining the Character Candidates produced by the Classifier, for example, for an input image that was divided on 8 segments, taking 3 character candidates by segment, $3^8 = 6,561$ String Candidates will be evaluated. But, if we divide it in two groups, only $3^4 + 3^4 = 162$ components must be evaluated. Accepted components, are recombined to form the reduced list of String Candidates which will be presented to the lexicon structure on searching for valid words (see fig. 2).

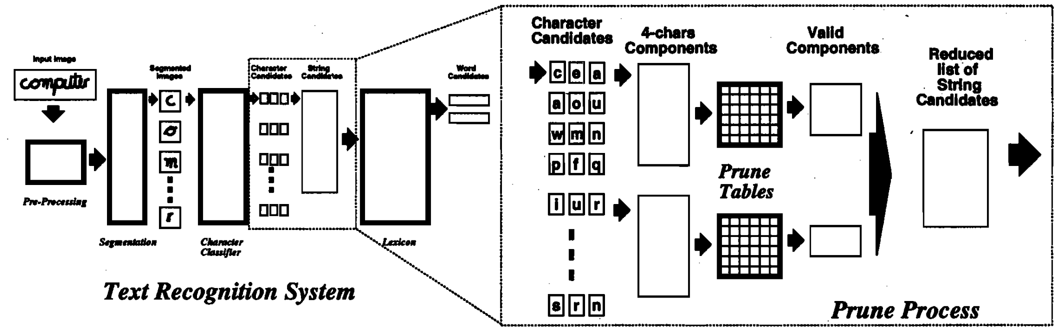


Figure 2: Including the Prune Process

If we represent $N_{strings}$ by its components:

$$N_{strings} = (N_{chars})^{\frac{N_{segments}}{2} + \frac{N_{segments}}{2}} = \underbrace{(N_{chars})^{\frac{N_{segments}}{2}}}_{component} \times \underbrace{(N_{chars})^{\frac{N_{segments}}{2}}}_{component} \tag{1}$$

Then, instead of create the entire list of String Candidates, we evaluate their components, e.g. verify if they exists on the lexicon structure or not. It is done by direct addressing [2] a table that contains one slot of 1 bit for each component in the lexicon. The size of the table is 2^r bits (where r is the size on bits of a component). Then, if we represent each of the 26 characters of English with 5 bits: from 00001 for letter **a** to

11010 for letter **z**, a table for 4 characters components will need 2^{20} locations of 1 bit (128KB), and for 5 characters, 2^{25}(4MB). Wells et al. [3], reported that using a long list of words, 85.4% of 2-grams are allowable, 37.1% of 3-grams would be accepted, and in the case of 4-gram only 5.5% are accepted. A good balance between a high rejection rate and memory usage could be obtained by using 4 characters components.

To produce the Prune Tables, first the lexicon is divided in four parts: 1) containing words of 1 to 4 characters, 2) words from 5 to 8 characters, 3) words from 9 to 12 characters and 4) words from 13 to 16 characters. Then each word is left justified, filling with NULL characters the gap at the right. Next, all the characters are represented on 5 bits. On this way, all the "words" from 1 to 4 characters, are represented by 20 bits (4 characters × 5 bits); then using the value of each word as the address on the Prune Table, the bit so pointed is set (bit = 1). Words from 5 to 8 characters are represented by 40 bits, then 2 Prune Tables are needed. On the same way for 9 to 12 characters, 3 tables are created, and for 13 to 16 characters, 4 tables. (see fig. 3)

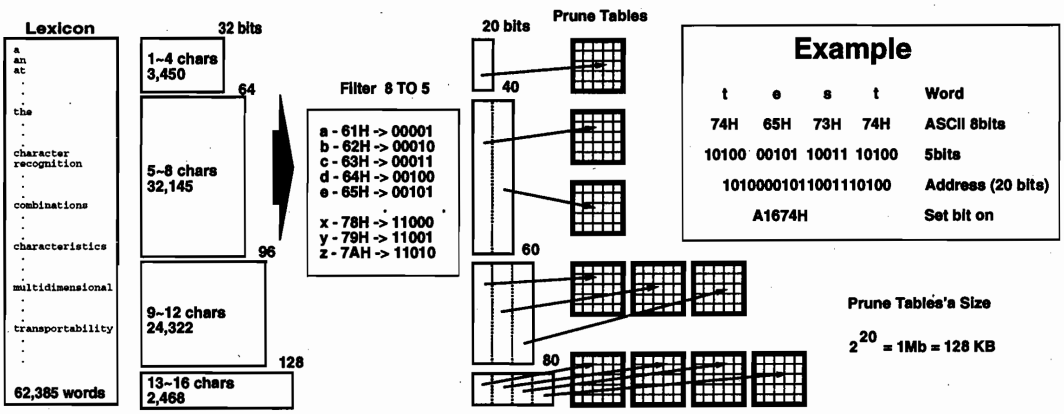


Figure 3: Prune Tables

To evaluate a component, it is only needed to read all their 4 characters, represent them on 5 bits, concatenate it to obtain the address on the Prune Table, and then verify if the bit pointed is set (accepted) or not (rejected). It is simple, efficient and fast, in all the cases it takes only one memory read.

3 Tests and Results

To evaluate the performance of the proposed method we used an English Lexicon of 62,385 words, that was divided by word size as showed on table 1. Taking 3 character candidates by segment and components of 4 characters, with Prune Tables of 128KB, two experiments were conduced:

1) Rejection Rate. Using a classifier simulator [4], we produced large data sets (see table 1) to simulate the most common mistakes produced on segmentation based recognition systems. The objective was to verify how good the proposed system can reject not valid combinations of characters.

2) Real task. Using a paragraph (see table 1), which contains words of different length. Results are illustrated on table 2.

Lexicon		Data Sets			
Chars	Size	Name	Strings	Chars/String	Paragraph Test
1∼4	3,450	the	48	3,4	The goal on text recognition is to convert information represented on a spatial form into a symbolic one. A typical text recognition system receives an image of a word as an entry, segments it and, for each segmented component produces character candidates, which are then combined to form string candidates. While the list of string candidates could be very large, it is mostly composed by not valid combinations of characters, which must be pruned quickly
5∼8	32,145	alterate	62,208	8	
9∼12	24,322	window	8,820	6,7,8,9	
13∼16	2,468	animation	907,200	9,10,11,12	
		character	139,968	9,10	
		recognition	1,944,000	11, 12,13	
				75 words 40 are 4 ⇒ characters or shorter	

Table 1: Lexicon and Data sets used on the experiments

4 Discussion

Results from experiment 1, showed that the proposed system performs good on rejecting not valid components of String Candidates. From experiment 2 we confirmed that by using direct-addressing, the proposed method can achieve a very fast performance, compared with that obtained for the same task when no lexicon reduction process is used. For *Short Words* (1 to 4 characters) the proposed method can complete the Prune Process and the Search Process in ONE memory read (see fig. 4). Furthermore, on a study made by Suen [5], it was showed that most of the words found on written texts occupy two to five letters. According to that the proposed method will perform very fast on recognizing English text.

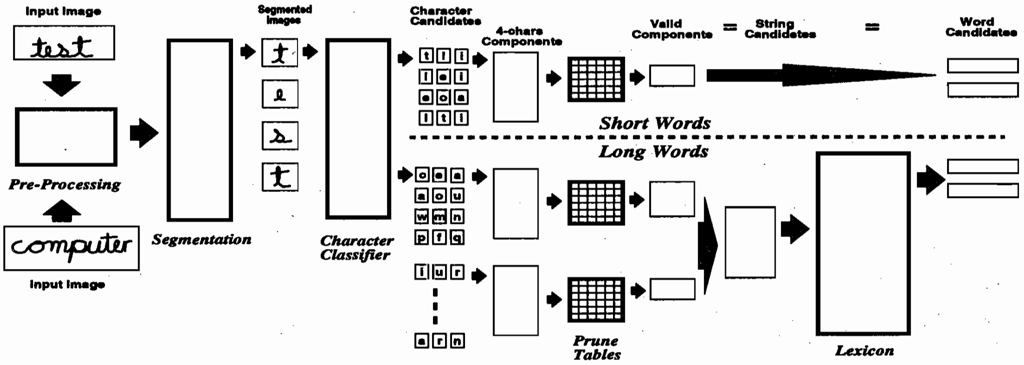


Figure 4: Text Recognition System including Prune

5 Conclusions

We presented a new method for reducing the time needed for lexical post-processing on cursive script recognition. Results obtained from our experiments showed an impressive degree of reduction without any loss on accuracy. It can be used by itself as a Search Method for words of 4 characters or less. On any case, the time needed to evaluate a 4 characters component is only one memory read. The proposed method is ideal to

Data set	Components Evaluated	Prune Time ⋆	Valid Components	String Candidates		Search Time ∗	Valid Words
				w/o Prune	w/ Prune		
Experiment 1: Degree of Rejections							
the	[48]	———	[2]	48	2	———	———
alterate	[288+181]	———	[40×19]	62,208	760	———	———
window	[153+184] †	———	[12×17] †	8,820	204 †	———	———
animation	[450+816+69]	———	[18×114×22]	907,200	45,144	———	———
character	[180+432+12]	———	[16×69×9]	139,968	9,936	———	———
recognition	[108+255+240]	———	[8×22×14]	1,944,000	2,464	———	———
Experiment 2: Real Task							
4 / 1	[3]×4	24μs	4	12			4
17 / 2	[9]×17	306μs	19	153			19
7 / 3	[27]×7	378μs	11	189			11
12 / 4	[81]×12	1,944μs	13	972			13
8 / 5	[81+3]×8	1,344μs	[9×1] [11×2] [7×2] [9×2] [13×2] [7×3] [13×1] [11×2]	1,944	145	2.9 ms	12
5 / 6	[81+9]×5	900 μs	[9×1] [13×2] [7×3] [13×4] [11×2]	3,645	130	2.6ms	7
4 / 7	[81+27]×4	864 μs	[6×1] [13×2] [7×3] [14×5]	8,748	123	2.46 ms	4
6 / 8	[81+81]×6	1,944 μs	[9×1] [13×2] [7×3][13×1] [11×2] [11×2]	39,366	113	2.26 ms	8
3 / 9	[81+81+3]×3	990 μs	[18×13×2] [9×14×3] [9×12×2]	59,049	1,062	21.2 ms	6
4 / 10	[81+81+9]×4	1,368 μs	[9×13×4] [8×6×4] [9×11×5] [16×13×4]	236,196	1,987	39.7 ms	4
4 / 11	[81+81+27]×4	1,512 μs	[16×16×4] [9×11×2] [9×12×3] [8×13×2]	708,588	1,754	35.1 ms	4
1 / 12	[81+81+81]	486 μs	[9×13×7]	531,441	819	16.4 ms	1
75 words	6,030	12.1 msⓅ			6,133	122.62 msⓈ	
Total time with Prunning Ⓟ+Ⓢ						**134.72** ms	
Total time without Prunning				1,590,303 × 20μs = **31.81** $sec.$			

⋆ The time to read a Prune Table is 2 μs. ∗ The average time to search for a string on the lexicon structure is 20μs.
† From 153 components evaluated, only 12 were valid, and from 184 evaluated, only 17 were valid, then the number of String Candidates to be evaluated was 12 × 17 = 204.

Table 2: Results

be implemented on parallel computers, or better yet in hardware, producing a very fast system. Work is in progress to incorporate this method with a fast search method to obtain a very fast lexical post-processing system to be applied on cursive script recognition.

References

[1] S. Madhvanath and S. N. Srihari. Effective reduction of large lexicons for recognition of offline cursive script. In *Proc. of the 5th Intl. Workshop on Front. in Handwriting Rec.*, pages 189–194, 1996.

[2] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* The MIT Press, 1994.

[3] C.J. Wells, L.J. Evett, P.E. Whitby, and R.J. Whitrow. Fast dictionary look-up for contextual word recognition. *Pattern Recognition*, 23(5):501–508, 1990.

[4] R. M. K. SINHA. On using syntactic constraints in text recognition. In *Proceedings of 2nd Intl. Conf. on Doc. Anal. and Rec.*, pages 858–861, 1993.

[5] Ching Y. Suen. N-gram statistics for natural language understanding and text processing. *IEEE Trans. on Patt. Anal. and Machine Intell.*, PAMI-1, No. 2:164–172, april 1979.

遞迴式類神經網路在語言模式處理上的研究

The Study Of Recurrent Neural Networks For Language Modeling

王文俊，李俊曉，劉繼謚
中華電信股份有限公司 中華電信研究所

摘要

在語音辨認及音韻分析處理，遞迴式類神經網路(recurrent neural network)已被廣泛利用並得到不錯的結果，它的特性是將前一時段的輸出層或隱藏層的輸出以延遲的方式與下一時段之輸入訊號一併輸入網路進行學習，因此可以有效利用時間軸上的資訊。本文則嘗試利用此種類神經網路架構進行語言模式的處理，實驗的語料是採用漢語平衡語料庫，而由於考慮參數量及複雜度，因此所有訓練及測試都是直接在詞類串列上進行。實驗的進行是以目前詞之詞類作為輸入，而以下一詞之詞類為目標值，對網路的各項連接加權值作最佳化的調整。對於此種方法所得結果的評估，一方面是和詞類雙連文法(part-of-speech bigram) 及詞類三連文法(part-of-speech trigram)比較，觀察其相似程度；另一方面則是觀察真實詞類串列和神經網路預測值之差異藉以判斷其能否協助決定子句段落的位置。實驗結果顯示利用類神經網路可以極類似於雙連文法及三連文法，除此以外此方法也有助於文句分析以決定子句段落位置。

一、 前言

文字處理是語音合成的重要程序，除了正確的字轉音必須在此完成以外，子句段落位置的決定也是一項重要工作。而由於大部份語音合成系統都希望能達到即時處理的要求，以及精確的文句分析絕對需要具備深厚的語言學基礎，使得這項工作並不容易完成。本篇論文則嘗試利用遞迴式類神經網路架構進行此項研究。遞迴式類神經網路已被廣泛利用在語音辨認[Hunt 1993]及音韻分析處理上，並得到不錯的結果。它的特性是將前一時段的輸出層或隱藏層的輸出以延遲的方式與下一時段之輸入訊號一併輸入網路進行學習，因此可以有效利用時間軸上的資訊。而文句中的文字串列或詞類串列也同樣具有時軸上前後的關係，這樣的關係能否用遞迴式類神經網路進行學習是本篇論文的探討重點。實驗將對漢語平衡語料庫的文句資料進行學習，根據大部份統計模式的原理，在多次的累進訓練後，高出現頻率的詞類串列必然會得到較低的誤差值。而假設網路的學習效果很好的話，則在一些子句的轉接處，神經網路的預測值和目標值的差異將會變大，而這樣的變化即可被運用來決定子句段落位置。

以中文常用字 5401 個字及常用詞 80000 個詞進行分析，將會有訓練語料不足、參數量太大等問題，因此在此將只探討詞類間的關係。也就是利用已作好詞類標示的漢語平衡語料庫，將每一個詞的詞類依序作為輸入，而以下一個詞的詞類為目標值，對網路的各項連接加權值作最佳化的調整。評估實驗結果時，除了與實際值作比較外，訓練結果和詞類雙連文法(part-of-speech bigram) 及詞類三連文法(part-of-speech trigram)的相似程度也是比較的重點；另一方面則也將觀察真實詞類串列和神經網路預測值之差異藉以判斷其能否協助決定子句段落的位置。

二、 遞迴式類神經網路

遞迴式類神經網路在很多方面都和傳統的前進式神經網路(feed-forward neural network)類似，最大的不同點是在它的架構變化如圖一所示，將隱藏層或輸出層經過延遲後和下一時段之輸入特徵一併再作輸入，其餘小變化的作法也包括將延遲的數目增加以及在不同層加入不同的輸入等等。遞迴式類神經網路對時軸序列訊號的分類有非常不錯的效果，它可以學習到訊號間的複雜關係，因此對大多數語音辨認處理而言，遞迴式類神經網路是一種非常好的架構。

至於利用遞迴式類神經網路在語言模式應用的例子[Elman 1990]則有屬於小範圍的文字預測，這類實驗是利用約 30 至 40 個詞構成一些短句子，再將這些句子輸入神經網路進行訓練，訓練過程是以下一個詞為目標值，而隱藏層的輸出則可以被用來作類似詞類的分類，因為隱藏層的輸出呈現出有限狀態機器(finite state machine)的分佈，而這些有限狀態是受到前後詞的影響。至於本文的運用則是以經過詞類標示之語料進行訓練，輸入目前詞的詞類而將目標值定為下一個詞之詞類。

如上所述，遞迴式類神經網路的優點是對訊號具有分類的能力，不過卻需要耗費相當大的計算資源，雖然有很多方法被提出來提高訓練速度及分類的準確性，如加權值調整的方法，輸出轉換函數的選擇以及輸出延遲等，但本文將不在這些地方進行討論，而將探討此種網路架構是否有助於解決本文所提出的問題以及增加延遲的數目如圖二所示能否使模式更精準，另外為避免訓練耗時因此訓練語料並未使用整個漢語平衡語料庫，而有關訓練語料庫大小對實驗結果的影響也將在此作比較。

三、實驗描述

本文之實驗設計除想比較神經網路與雙連文法或三連文法之差異，另外也想探討資料量對文法模式訓練之影響。因此希望能比較出語料庫大小的影響，目前是用漢語平衡語料庫之前五十個檔案共約 140 萬詞構成一個大語料庫，而小語料庫的設計則為考慮文句涵蓋範圍而將漢語平衡語料庫之各個檔案隨機選取數句構成一個約含三萬詞的測試語料庫。

實驗的進行是利用上述之大小語料庫分別去統計出詞類雙連文法及詞類三連文法，另外以小語料庫進行遞迴式神經網路之訓練，訓練時是輸入目前詞之詞類，而以下一詞之詞類作為理想值，網路之架構為 55 個輸入節點(46 類 POS 及 9 類標點符號[中研院 1995])， 60 個隱藏層節點， 55 個輸出層節點。為加快訓練速度及收斂速度，我們同時採用了累進訓練(incremental training)的方法，也就是說在頭幾次的訓練僅使用到少量短句的語料，待接近收斂後再增加語料。如此的作法即可使訓練速度加快，訓練之結果隨著次數增加其誤差值呈現遞減的變化。有關最佳一至五的結果含有理想值之包含率如表一所示，訓練結果的好壞若直接與理想值作平均誤差之估算其結果實不令人滿意，因此我們也將評斷此結果與雙連文法及三連文法的接近程度，此結果則如表二所示，同時我們也將訓練結果和一個無機率之模式比較，也就是所有詞類均是任一詞類之可能後接詞類且其機率值均相同。而多延遲遞迴式類神經網路能否使預測模式與雙連文法、三連文法甚至於 N-gram 文法更加接近也有初步的結果將在下節介紹。

四、 實驗結果與討論

由表一可看出訓練結果的 top 1 正確率並不高，但是此模式的目標原本就不是在將預測正確率調整至百分之百，而是希望能儘可能縮小和其他統計式機率模式的差異。由表二可看出訓練結果和雙連文法非常接近，而和三連文法差異稍大；因此我們原本寄望多延遲的架構能模擬成更多詞的影響，而能降低與三連文法的差異，但由表三及表四的比較，多延遲模式所得之結果並未如預期比單一延遲之效果好，這似乎說明因為中文的特性，對詞的定義不夠明確，有些長詞都可能被斷成數個短詞，以致於2個或3個的延遲並不直接和 trigram 及 4-gram 等效。另外由表中的數據也可以看出我們提供了分別由大小語料庫統計出的雙連文法及三連文法，而發現訓練結果與得自大語料庫的統計結果較接近。

雖然利用此方法要作到自然語言了解的地步還有很大的距離，但在提供一個迅速且有效的子句段落預測模式仍有一些效果。如圖三及四所示之測試句子，在平均誤差曲線的各個谷底處都有較大可能被視為一個子句段落位置。而在審視過大部份的測試語句也發現在連接詞及副詞之前都會有明顯變化，而對於“中”、“裡”等後置詞可能是因為出現次數過低使得變化並不一致，另外在“的”之後若接名詞則會在該名詞之後產生變化，但若接具名物化之述詞時，因為在詞類選擇時將包括名物化的所有特徵都捨棄，以致於在此狀況時，大部份的變化都落在“的”之後。總括而言，對於本文所提出的模式仍有許多地方亟待改進，包括詞類分類的選擇，訓練語料的擴大以及神經網路的架構調整，而一些語法上的特殊修飾也是必須作詳細討論。

五、 參考文獻

1. Andrew Hunt, “Recurrent Neural Networks for Syllabification”, Speech Communication, vol. 13, pp. 323-332, 1993.

2. Jeffrey L. Elman, “Finding Structure in Time”, Cognitive Science, vol. 14, pp. 179-211, 1990.

3. 中央研究院，詞庫小組，“中央研究院平衡語料庫的內容與說明”， Technical Report no.95-02, 1995.

表一. 前一及前五含最佳結果之包含率

	top 1	top 5	top 10
inclusion rate	32%	63%	78%

表二. 訓練結果與得自不同語料庫之文法的平均誤差

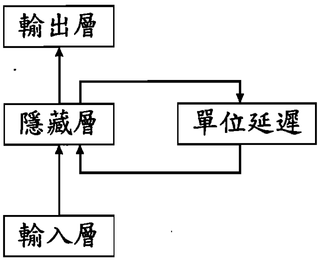
	unigram	bigram	trigram
small corpus	0.40156	0.19432	0.28523
large corpus	0.37294	0.15756	0.24137

表三. 多延遲與否的訓練結果與得自小語料庫文法的平均誤差

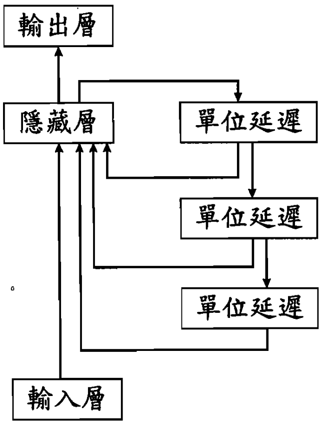
small corpus	bigram	trigram
without md	0.19432	0.28523
with md	0.19733	0.29058

表四. 多延遲與否的訓練結果與得自大語料庫文法的平均誤差

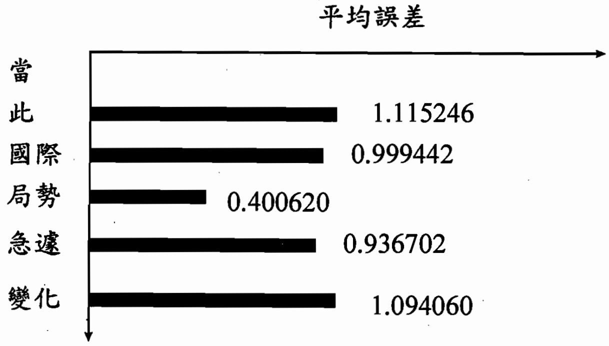
large corpus	bigram	trigram
without md	0.15756	0.24137
with md	0.16276	0.24542



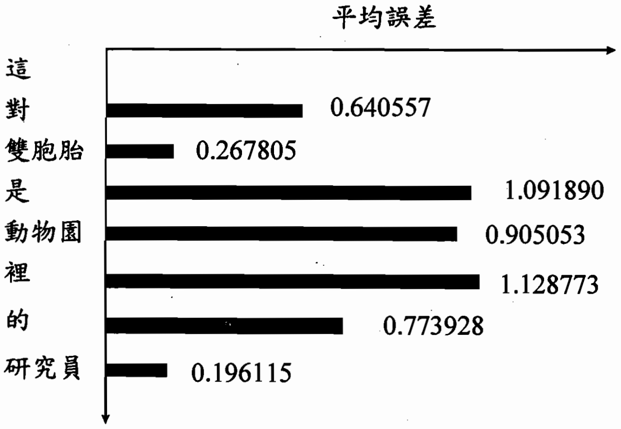
圖一. 遞迴式類神經網路之基本架構



圖二. 多延遲遞迴式類神經網路之架構



圖三. 測試句子1 之平均誤差結果



圖四. 測試句子2 之平均誤差結果

A Simple Heuristic Approach for Word Segmentation

Wing-Kwong Wong, Chenming Hsu, Jien-Iao Chen, Jien-Chi Yu
Dept. of Electronic Eng., Natl. Yunlin Inst. of Tech., Touliu, Yunlin, Taiwan.
wongwk@el.yuntech.edu.tw

Abstract

Previous approaches to Chinese word segmentation includes maximal matching heuristic, morphological rules, and POS tag statistics. This paper proposes to estimate the word occurrence probabilities with some "unlikelihood" scores based only on word lengths. Also, the problem of maximizing likelihood is shown to be equivalent to the graph problem of shortest path, whose edges stands for words with their corresponding unlikelihood scores.

Introduction

Chinese sentences have no white spaces to delimit words, unlike English. Word segmentation has been a fundamental problem in Chinese text processing since almost all applications in this area must deal with this problem. Many solutions have been published before. Almost all solutions use a word dictionary to determine whether a given character sequence is a legal word. The earliest proposal is probably the maximal matching approach, which favors long words during segmentation (e.g., [Li et al. 91], [Lochovsky & Chung 94]). Since more frequently used words are more likely to occur, word frequencies, which estimate independent occurrence probabilities, are used in ([Chang et al. 91], [Nie et al. 95]). Furthermore, independent word probabilities are not as powerful as co-occurrence statistics of grammatical categories in predicting word sequence. Thus, morphological or POS (part of speech) tagging statistics (e.g., bigrams and trigrams), are used in [Li et al. 91], [Chiang et al. 92], [Lin et al. 93], [Pan & Chang 93], [Luk 94]. Morphological rules, which use POS categories of words to predict the formation of compound words, are used in [Lin et al. 93]. When there is no unknown words in the tested sentences, all the above approaches perform quite well, with accuracy rates ranging from 95% to 99%. When many unknown words exist in the tested data, which is generally the case in real-life texts, however, performance degrades down to 60%. Some approaches to detecting unknown words are proposed in [Chiang et al. 92], [Lin et al. 93], and [Luk 94].

Model

The word segmentation problem can be stated formally in the following way:

Given a character sequence $C_1C_2C_3\ldots C_n$, determine the optimal word sequence $W_1W_2W_3\ldots W_m$, such that each W_i, where i=1,…,n, spans a subsequence of C_j's, and all such subsequences do not overlap and the entire word sequence spans the entire character sequence.

Intuitively, the optimal word sequence is one that would make the most sense out of the character sequence in the given communication context. All approaches to word segmentation attempt to give an operational, i.e., computational, approximation to this intuitive definition of optimality. For statistical approach, this is equivalent in choosing the word sequence that maximizes the conditional probability of word sequence $W_1W_2\ldots W_m$ given the character sequence $C_1C_2\ldots C_n$ (e.g., [Pan & Chang 93]):

$$\max P(\mathrm{W}_1\mathrm{W}_2\ldots\mathrm{W}_m|C_1C_2...C_n).$$

The simplest and roughest computational model is one that assumes the words are all independent of each other and independent of the given character sequence:

$$\max P(\mathrm{W}_1\mathrm{W}_2\ldots\mathrm{W}_m|C_1C_2...C_n)$$
$$\cong \max P(\mathrm{W}_1)...P(\mathrm{W}_m) \cong \max \textstyle\prod_i P(\mathrm{W}_i)$$

The maximal matching approach, which favors the longest words first met when scanning the characters in a sentence either from left to right or vice versa. Its basic intuition is that the longer a word is, the greater is its occurrence probability when its character sequence is found in a sentence. It provides an effective heuristic and can achieve a high accuracy rate. However, it is only a local heuristic and does not achieve a global optimum. Therefore, it is very efficient but can fail in cases where the first met longest word produces an incorrect segmentation. For example, 【省長上行政院】 is segmented as 【省長｜上行 ｜政｜院】, since 上行 is the longest word scanned from the left at the character 上. However, the correct segmentation is 【省長｜上｜行政院】. If more global information is used, i.e., the later but longer word 行政院 is considered, then a correct segmentation should result.

The basic flaw of maximal matching is that it is only a local heuristic. We propose to keep its basic intuition that the likelihood of a longer word is greater but to use it with global optimization, similar to the above baseline model $\max \prod_i P(\mathrm{W}_\mathrm{i})$. To achieve this goal, several mathematical transformations are needed. First, to maximize the product of probabilities is equivalent to maximize the sum of the log of these probabilities ([Chiang et al. 92]). Second, some likelihood function can be used to estimate the log of probabilities. Third, likelihood maximization is equivalent to "unlikelihood" minimization and some unlikelihood function can be selected to reflect the likelihood function:

$$\begin{aligned}
&\max \prod_i P(\mathrm{W}_\mathrm{i}) \\
\cong\ &\max \sum_i \log P(\mathrm{W}_\mathrm{i}) \\
\cong\ &\max \sum_i \mathit{likelihood}(\mathrm{W}_\mathrm{i}) \\
\cong\ &\min \sum_i \mathit{unlikelihood}(\mathrm{W}_\mathrm{i})
\end{aligned}$$

Thus, we propose to assign some unlikelihood scores to words based on their length:

In dictionary?	Word Length	Unlikelihood Score
No	Don't care	∞
No	>=5	∞
Yes	1	7
Yes	2	4
Yes	3	2
Yes	4	1

If a character sequence is not in the dictionary, then its unlikelihood score is infinity. To keep the analysis simple, word length is limited to four. This would also reduce the dictionary size and the running time for checking whether a character sequence is in the dictionary or not. Thus, there is no need to check character sequence whose length exceeds four. For words that are in the dictionary, the scores are 7, 4, 2, 1 respectively for words with lengths 1, 2, 3, and 4. This means that longer words are less likely to occur as random character sequences in natural texts and is the basic heuristic for the maximal matching method. This score assignment, however, asserts more than this basic heuristic. For example, this assignment says that a two bi-character words 【 AB | CD 】, whose unlikelihood score is 4+4=8, is more likely than a three-character word followed by a single-character word 【 ABC | D 】 or a single-character word followed by a three-character word 【 A | BCD 】, whose unlikelihood scores are 7+2=9. An example is 【台北 | 市民】 is better than 【台北市 | 民】. Similarly, 【 AB | CDE 】, with score 6, is more better than 【 A | BCDE 】 or 【 ABCD | E 】, with score 8; 【 ABC | DEF 】, with score 4, is more better than 【 AB | CDEF 】 or 【 ABCD | EF 】, with score 5. These assumptions are subject to further empirical testing.

Algorithm

Word segmentation problem is commonly portrayed as an optimization problem. [Fan & Tsai 87] uses a relaxation algorithm. [Chang et al. 91] considers segmentation as a constraint satisfaction problem and employs a dynamic programming method called arc consistency. [Nie et al. 94] presents an algorithm that seems to be a recursive version of the Viterbi algorithm (e.g., [Allen 95], [Bertsekas 87]). All these algorithms are actually variations of solutions for the shortest path problem, which is a fundamental graph problem (e.g., see [Ahuja et al. 93]). Here is how we transform word segmentation into such a graph problem:

1. Put the number 0 at the front of the sentence in question and move past the first character.
2. Put the next number between the last character and the next character.
3. Repeat Step 2 until the last character of the sentence is encountered.
4. Then put the next number following the last character.

The resulting row of symbols becomes:

$$0 \quad C_1 \quad 1 \quad C_2 \quad 2 \quad C_3 \quad \ldots \quad n\text{-}1 \quad C_n \quad n$$

where C_i is the ith character in the sentence, and n is the number of characters in the sentence. Each number x in the row of symbols is considered as a node called $node_x$ in a graph. Then any legal word $C_iC_{i+1}\ldots C_j$ found in dictionary, where i<=j, in the sentence is a directed edge from $node_i$ to $node_{j+1}$. The distance of the edge from $node_i$ to $node_{j+1}$ is the unlikelihood score of the word. Therefore the segmentation problem becomes the graph problem of finding the shortest path from $node_0$ to $node_n$---a path is a series of edges connecting the source node and the destination node and the path distance is the sum of distances of all the edges on the path.

With the proposed scheme of unlikelihood score assignment, the unlikelihood score of each edge (or character sequence) and that of each path (or segmentation) are given below (remember the node numbering scheme: 0 台 1 北 2 市 3 民 4.):

Character sequence	Edge	Unlikelihood score	Legal word?
台, 北, 市, 民	0-1, 1-2, 2-3, 3-4	7	Yes
台北	0-2	4	Yes
北市	1-3	4	Yes
市民	2-4	4	Yes
台北市	0-3	2	Yes
北市民	1-4	∞	No
台北市民	0-4	∞	No

	Segmentation	Path	Unlikelihood score
1	【台北市民】	0-4	∞
2	【台 ｜ 北市民】	0-1-4	1 + ∞ = ∞
3	【台北市 ｜ 民】	0-3-4	2 + 7 = 9
4	【台北 ｜ 市 ｜ 民】	0-2-3-4	4 + 7 + 7 = 18
5	【台 ｜ 北市 ｜ 民】	0-1-3-4	7 + 4 + 7 = 18
6	【台 ｜ 北 ｜ 市 ｜ 民】	0-1-2-3-4	7 + 7 + 7 + 7 = 28
7	【台北 ｜ 市民】	0-2-4	4 + 4 = 8
8	【台 ｜ 北 ｜ 市民】	0-1-2-4	7 + 7 + 4 = 18

According to the above data, the shortest path is 0-2-4, which corresponds to Segmentation 7 【台北 ｜ 市民】, since edge 0-2 stands for the word 台北 and edge 2-4 stands for the word 市民. Comparing automatic segmentation results to human segmented texts indicates an accuracy rate of 98.5%.

Acknowledgment

The authors thank Dr. Wen-Lian Hsu for discussing ideas on Chinese text processing and for providing a dictionary. This research is supported by the NSC with grant NSC86-2511-S-224-001-ICL.

References

Ahuja, R. K., T. L. Magnanti & J. B. Orlin, "Network Flows," Prentice Hall, New Jersey, 1993.

Allen, J. "Natural Language Understanding," Benjamin-Cummins, Redwood City, CA, 1995, p.202.

Bertsekas, D. P., "Dynamic Programming," Prentice-Hall, N.J., 1987, p.30.

Chang et al. 張俊盛、陳志達、陳舜德，限制式滿足及機率最佳化的中文斷詞方法， ROCLING IV, 1991, pp. 147-165.

Chiang, T. H., J. S. Chang, M. Y. Lin and K. Y. Su. "Statistical models for word segmentation and unknown word resolution." ROCLING V, 1992, pp. 123-146.

Fan, C. K. & W. H. Tsai, "Automatic word identification in Chinese sentences by the relaxation technique," Proc. of National Computer Symposium, 1987, pp. 423-431.

Li et al. 黎邦洋、藺蓀、孫朝奮、孫茂松，一種主要使用語料庫標記進行歧義校正的、最大匹配漢語自動分詞算法設計， ROCLING IV, 1991, pp. 147-165.

Lin, M. Y., T. H. Chiang and K. Y. Su, "A preliminary study on unknown word problem in Chinese word segmentation," ROCLING VI, 1993, pp. 119-141.

Lochovsky, A. F. & K. H. Chung, "Word segmentation for Chinese phonetic symbols," 1994 International Computer Symposium, Vol. 2, 1994, pp. 911-916.

Luk, W. P. R., "Chinese-word segmentation based on maximal-matching and bigram techniques," ROCLING VII, 1994, pp. 273-282.

Nie, J. Y., X. Ren and M. Brisebois, "A unifying approach to segmentation of Chinese and its application to text retrieval," ROCLING VIII, 1995, pp. 175-190.

Pan & Chang 彭載衍、張俊盛，中文辭彙歧義之研究——斷詞與詞性標示， ROCLING VI, 1993, pp. 173-193.

Prosody Generation in a Chinese TTS System Based on a Hierarchical Word Prosody Template Tree

Chung-Hsien Wu and Jau-Hung Chen

Institute of Information Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
E-mail: {chwu, chenjh}@server2.iie.ncku.edu.tw

Abstract

In this paper, a prosody generation method based on a hierarchical word prosody template tree for the generation of prosodic information in a sentence is proposed. The hierarchical word prosody template tree is established from a large continuous speech database according to the linguistic features: tone combination, word length, part of speech (POS) of the word, and word position in a sentence. This template tree stores the prosodic features including pitch contour, energy contour, and syllable duration of a word for possible combinations of linguistic features. In the inside test, the prosody of the synthesized speech resembles that of the original speech for a typical sentence.

1. Introduction

In recent years, text-to-speech systems have been able to generate highly intelligible synthesized speech. However, further improvement of synthesized speech is expected with respect to the prosodic information. Two general approaches have been proposed for generation of the prosodic information: the rule-based approach (D. H. Klatt 1987, L. S. Lee 1989) and the data-driven approach (C. A. Moor 1995, H. S. Hwang 1995). However, the rule-based and data-driven approaches use small number of prosodic patterns to represent the diverse prosody. Consequently, the prosody generated by rules or neural networks is an average version of all the original prosody in the same linguistic conditions. These approaches cannot give the best result even though the sentence (or the linguistic features) are identical to a sentence (or the linguistic features) in the training database. Therefore, a more detailed and accurate description between prosodic and linguistic features of a sentence is desired to achieve a better synthesized result.

In this paper, a word prosody template tree recording all the relationship between the linguistic features and the word prosodic templates in the speech database is established. Each word prosodic template contains the syllable duration, energy contour and pitch contour of the word. For each word in a sentence/phrase, the word position is first determined and used to traverse the template tree. Word length is then used to find the tone combination subtree. Finally, tone combination for the word is used to retrieve the word prosody template.

2. Construction of Word Prosody Template Tree

In the Chinese TTS system, some linguistic features are relevant to word prosodic information. They are tone combination, word length, POS of the word, and word position in a sentence. These features are discussed in more detail in the following.

(1) Tone combination: A word with length n consists of n syllable(s) in which each syllable has a tone. However, the neural tone generally appears at the end of a word. As a result, there are $4^{n-1} \cdot 5$ tone combinations for an n-syllable word.

(2) Word length: The intonational or prosodic relationship between syllables within a word is more obvious than that between two words. Therefore, word length of an n-syllable word is used to choose its corresponding word prosodic patterns with word length n.

(3) POS of the word: POS is also an important linguistic feature to determine the word prosody. In this paper, POS is divided into 21 categories. The distance between two categories is defined by the distance of their corresponding average prosodic patterns in the training database, i.e., word pitch contour, word energy contour, and syllable duration in the word. This distance is then normalized to lie between 0 and 1. Therefore, a POS distance table was established.

(4) Word position in a sentence: A word position ratio is defined as the order of the word position in the sentence divided by the number of words in the sentence.

Using the above linguistic features, a word prosody template tree is constructed. The linguistic features of a word, i.e., tone combination, word length, POS of the word and word position in a sentence, are associated with a set of prosodic patterns, i.e., syllable duration, energy contour, and pitch contour. To establish the word prosody template tree, a continuous speech database established by the Telecommunication Laboratories, Chunghwa Telecom Co., Taiwan, containing 655 reading utterances was used. The speech signals were digitized by a 16-bit A/D converter at a 20-kHz sampling rate. The syllable segmentation and phonetic labels were manually done. A total number of 38907 syllables and their phonetic labels were obtained. Using the text analysis, 9698 reference words (including 2-, 3-, and 4-syllable words) and their corresponding word prosodic patterns were obtained.

The structure of the word prosody template tree is shown in Fig. 1. There are three levels: word position level, word length level and tone combination level. In the word position level, an input word pattern is classified into one of the following three categories: the beginning part of a sentence (BOS), the middle part of a sentence (MOS), and the end part of a sentence (EOS). Furthermore, each category has four branches according to word length. As illustrated in the word length level, they are monosyllable words, 2-syllable words, 3-syllable words, and 4-syllable words. Finally, the tone information of a word is described in the tone combination level. This level contains five groups according to their ending tones. The reason for this grouping configuration is to link the word prosody correlation between adjacent words. For each tone combination, a word prosody template is established to store the prosodic features: pitch contour, energy contour, and syllable duration. Besides, the POS and the word position ratio of the word are also stored.

3. Generation of Word Prosody Templates

The generation of the word prosody templates is shown in Fig. 2. An input sentence/phrase is first decomposed into a sequence of words by a word segmentation parser. Each word contains linguistic features including tones, word length, POS, and word position in a sentence. In our system, a word with length more than four is further divided into combinations of 1- to 4-syllable words. The deep first search algorithm (G. Chartrand 1993) is

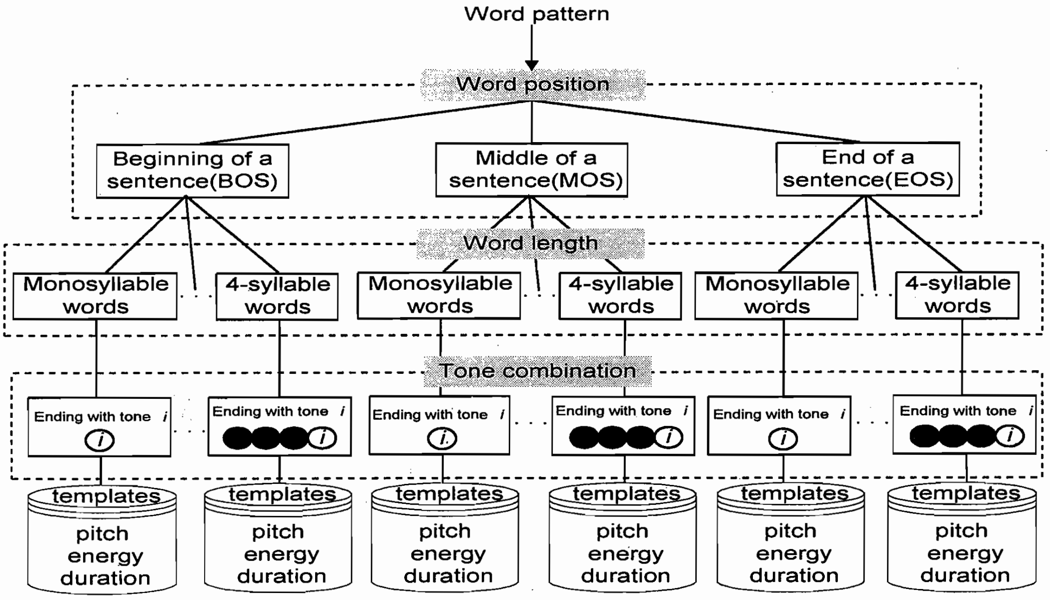
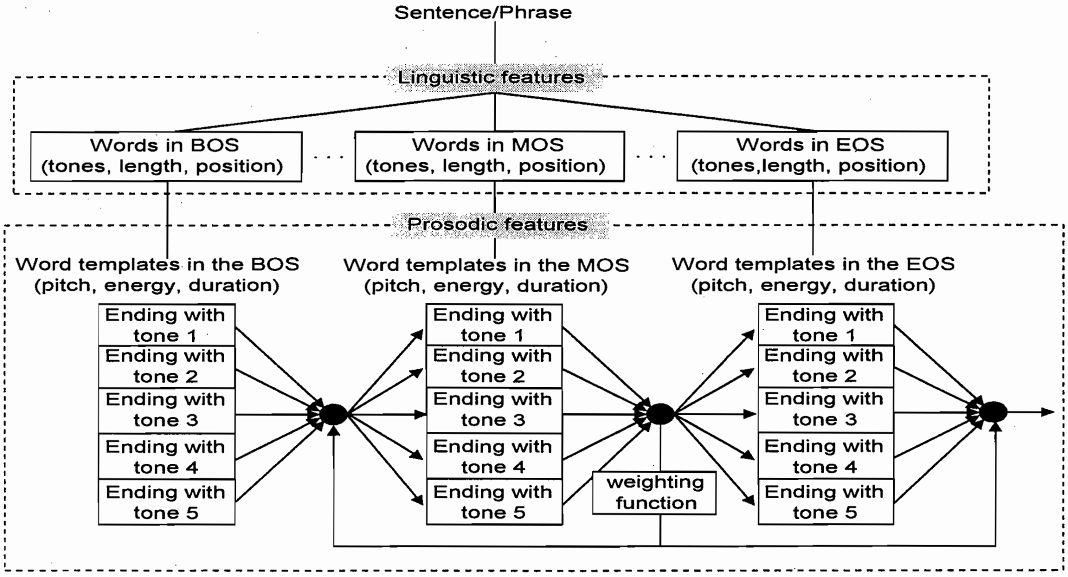


Fig. 1. Structure of the hierarchical word prosody template tree for word templates ending with tone *i*.



Sentence/Phrase
Linguistic features
Words in BOS (tones, length, position)
Words in MOS (tones, length, position)
Words in EOS (tones,length, position)
Prosodic features
Word templates in the BOS (pitch, energy, duration)
Word templates in the MOS (pitch, energy, duration)
Word templates in the EOS (pitch, energy, duration)
Ending with tone 1
Ending with tone 2
Ending with tone 3
Ending with tone 4
Ending with tone 5
weighting function

Fig. 2. Generation of the word prosody templates for a sentence/phrase.

employed to traverse the word prosody template tree. For each input word in a sentence, the word position and word length were first used to reach the word length level. Second, the tone combination is used to locate the corresponding word prosody template(s). There are two conditions of the target template in the tree: reachable and unreachable. They are discussed as follows.

(1) Reachable: If a target template can be reached and there is only one template found then output it. Otherwise, one of the template candidates is selected by calculating the linguistic distance between the input word and the reference words in the tree. The prosodic template corresponding to the reference word with the minimum linguistic distance is chosen as the output using the following distance estimation method.

$$j^* = \min_j \{ d_T(T_I, T_j) + d_C(C_I, C_j) + d_P(P_I, P_j) \}, \quad j=1,\cdots,J \qquad (1)$$

where J is the total number of words in a sentence corresponding to a given word position, word length, and tone combination. $d_T(T_I, T_j)$ represents the pitch distance between the average pitch of the input word T_I and that of the reference word T_j in the tree. $d_C(C_I, C_j)$ represents the linguistic distance between the POS of the input word C_I and that of the reference word C_j in the tree. $d_P(P_I, P_j)$ represents the absolute distance between word position ratio of the input word P_I and that of the reference word P_j.

(2) Unreachable: When a target template is unreachable, that means the linguistic feature of the input word does not appear in the speech database. There is no corresponding prosodic pattern in the template tree. However, we should find a suitable prosodic pattern to correspond to the input linguistic feature. In this case, the other two subtrees in the word length level are traversed in the following order according to the current word position: (A) If the current word position is BOS then traverse MOS subtree followed by EOS subtree. (B) If the current word position is MOS then traverse BOS subtree followed by EOS subtree. (C) If the current word position is EOS then traverse MOS subtree followed by BOS subtree.

4. Results

Fig. 3 illustrates an example of pitch contours of the original speech and synthesized speech. The first two panels display the waveform and the pitch contour of the original speech selected from the speech database. The last panel shows the corresponding pitch contours of the synthesized speech. By examination of the pitch contours, the synthesized pitch contours generated from the hierarchical word prosody template tree resemble their original counterparts. The results of listening test also confirm the good performance of this scheme.

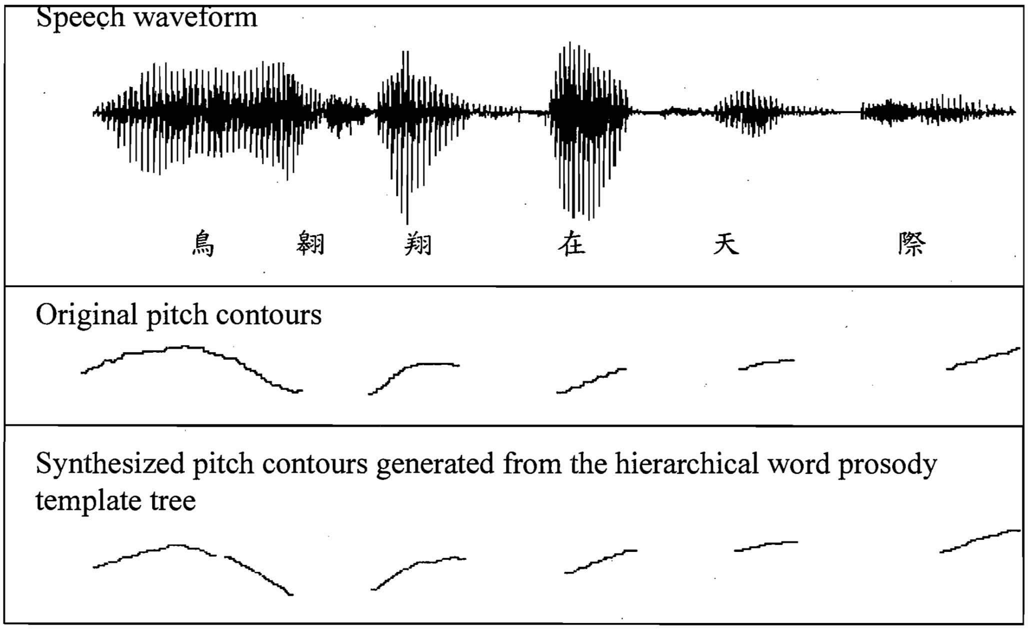


Fig. 3. An example of pitch contours of the original speech and synthesized speech.

5. Conclusions

In this paper, the construction and generation of prosodic information for Chinese text-to-speech conversion has been proposed to enhance the conventional rule-based approach. The prosodic information including pitch contour, energy contour, and syllable duration, was stored in a hierarchical word prosody template tree generated from a large speech database. Appropriate word prosodic templates in a sentence are selected from the tree according to the linguistic features. Evaluation by means of listening tests has confirmed the good performance of this scheme.

References

Klatt, D. H., "Review of text-to-speech conversion for English," J. Acoust. Soc. Amer., vol.82, No.3, 1987, pp.737-793.

Lee L. S., C. Y. Tseng, and M. Ouh-Young, "The synthesis rules in a Chinese text-to-speech system," IEEE Trans. ASSP, Vol.37, 1989, pp. 1309-1320

Moor C. A., J. F. Cohn and G. S. Katz, "Quantitative description and differentiation of fundamental frequency contours," Computer Speech and Language, Vol.8, 1994, pp.385-404.

Hwang S. H. and S. H. Chen, "A prosodic model of Mandarin speech and its application to pitch level generation for text-to-speech," in Proc. ICASSP, 1995, pp. 616-619.

Chartrand G. and O. R. Oellermann, *Applied and algorithmic graph theory*, McGraw-Hill, Inc., 1993, p.74.

AMBIGUITY RESOLUTION USING LEXICAL ASSOCIATION

Juntae Yoon and **Seonho Kim** and **Mansuk Song**
{queen, pobi, mssong}@december.yonsei.ac.kr
Department of Computer Science
Yonsei University, Seoul, Korea

Abstract

Lexical information has been shown to be crucial for decisions on ambiguities. Many statistical parsers is based on probabilities of this dependencies. Our system tried to conjoin lexical information to the best first parsing method and to show that every nodes can be determined using GAT(global association table) ,which is a new data structure to manage the lexical associations. In Korean, the structual ambiguity and the grammatical case ambiguity influence the accuracy of the parser. Lexical information between pairs of words is computed by co-occurrence data extracted from the corpus and to be extended to the conceptual association with thesaurus ,which it attempts to reduce parameter space.

1. Introduction

In Korean, to parse a sentence is to analyze the dependency relation among eojeols. Lexical association between eojeols can be applied in analyzing the dependency realtions in the agglutinative language such as Korean. Therefore, it is necessary to measure the lexical association to choose the correct parse tree. Besides, the grammatical cases of noun phrases are unknown in Korean when the NP has the auxiliary postposition, the postpostion of the NP is omitted, or the NP is moved by the relativization. To identify the unknown grammatical case, the lexical association between the verb and the noun phrase with a postposition is required because the grammatical case is determined by the postposition in Korean.

In this paper, we suggest the global association table(GAT) where lexical associations are globally controlled. The GAT provides the parser with the useful information required for parsing such as the lexical association. The lexical association between the predicate and the NP, is estimated by the cooccurrence relations extracted from corpus. On the basis of the associations presented by the GAT, the actions of the parser are directed and the unknown grammatical cases are identified. We extracted verb and noun co-occurrence data by the partial parser from 30 million eojeol corpus. To reduce parameter space the thesaurus was used on the assumption that the words in the same group behave similarly. Thus the associations of the predicate and the noun were estimated by the co-occurrence of verbs and noun classes. The system was shown to be efficient and precise by experiments.

2. Two kinds of ambiguities

Two types of ambiguities can be appeared in noun phrases and verbs in Korean sentences. The first is the structural ambiguities that are common in most languages. As the head follows its complement in Korean, the ambiguities are inevitable. In (Table 1), the nominal eojeol , '컴퓨터를(computer)', has the possibility to be dependent on '이용해서(using)' and '찾는다(seek)' in the parsing process. Second, the role of the noun phrase is decided by the postpositions for the most part but undecided sometimes. Several types of postpositions serves noun phrases as case markers. For instance, the postposition '을/를' makes nouns *objects*. However, The grammatical case cannot be turned out until parsing process under certain circumstances. For example, auxiliary postpositions add some meaning to NPs instead of marking grammatical cases. The postpositions of NPs can be sometimes omitted. The grammatical cases are veiled in these NPs, so they are uncovered in parsing process. The movement by relativization is also another

Structural ambiguity
많은(many) 사람들이(people) 컴퓨터를(computer) 이용해서(using) 자료를(data) 찾는다(seek). → Many people seek data using the computer.
Ambiguities of grammatical cases caused by the auxiliary postpostion
The first noun phrase is object and the second is subject though their postpositions are the same. . . . 책(book)-도 좋아했다(liked) . . . 나(I)-도 책을(book) 좋아했다(liked)
Ambiguities of grammatical cases caused by the relativization
. . . 메리가(Mary) 만난(meet) 친구(friend) . . . → . . . the friend whom Mary met . . . . . . 학교에(school) 간(go) 친구(friend) . . . → . . . the friend who went to school . . .

Table. 1: The examples of the ambiguities

example. The NP of the clause is moved out of relative clauses. In the relativization, two ambiguities should be resolved. The parser detects the moved noun and then catches what its grammatical case is. In (Table 1), '도' is the auxiliary postposition. The grammatical cases of the nominal eojeols, '책-도' and '나-도', can be identified in the parsing process dynamically. In the third example of the table, '친구' was moved from the clause. It is the object in the former sentence and the subject in the latter one.

3. Defining Global Association Table

We define the global association table as the data structure to record the association between eojeols. In order that the parser obtains information for disambiguations, it looks up the GAT in parsing. Our parser should resolve two ambiguities - structure and grammatical cases. Therefore, the GAT provides two kinds of information. One is for the comparison of associations and the other is for the identification of the grammatical cases.

The row and the column of the table represent eojeols occurring to the left-hand side and to the right-hand side in the parsing process, respectively. The left-hand side eojeol is the complement, and the right-hand side, the candidate for its head. That is, the $GAT(i,j)$ describes the degree of association in case the ith eojeol has a dependency relation to the jth eojeol. Because the head follows its complement in Korean, and the table is a triangular matrix.

To evaluate the association, we extracted co-occurrence data between predicates and nouns by the partial parser. The number of the pairs is 2,000,000 , but the number of the pairs whose the frequency is more than 2, is only 450,000. Considering the number of words in the word dictionary, we can't get enough co-occurrence data for analysis from the corpus. We use the thesaurus (Lim, 1992) to compute the association between groups of words. The parameter space for verb-noun co-occurrences can be reduced in to the co-occurrences of verbs and noun classes. This follows the assumption that words within a group behave similarly. In addition, the requirement ratio for the postposition of the verb is defined. That is, the parameter space was built in terms of the groups of nouns and the grammatical case that the verb demands.

3.1. Lexical Association

We use lexical associations for disambiguation. The lexical association of a nominal eojeol and a predicative eojeol is based on the frequency of co-occurrence. The association of modifier-head relations such as an adverb and a verb, or an adnominal and a noun, is estimated by distance.

First, the co-occurrence data of verbs and nouns were collected. The co-occurrence pairs of nominal eojeols and predicative eojeols were extracted by the partial parser from a corpus of 30 million eojeols. This approach explored by (Hindle, 1993) was shown to be effective for disambiguation aof the preposition attachment.

Second, the selectional restrictions were extracted from the co-occurrence data. Since about 15 percent of the words in our thesaurus have more than two categories, there are few words to have multiple categories. We assigned the thesaurus classes of the words which has a single category.

Third, we builded up the co-occurrence data of verbs and functional words, which was made with the data described above.

We use the data to define the association between the verb and the noun phrase as follows. Let

$$V = \{v_1, \ldots, v_l\},\ N = \{n_1, \ldots, n_m\},$$
$$C = \{c_1, \ldots, c_n\},\ S = \{\phi, \text{가}\ , \text{를}, \text{에}, \ldots\}$$

V, N, C, S be the sets of predicates, nouns, noun classes, and syntactic relations respectively. The postposition is given ϕ, in case the grammatical case is unknown. Given $v \in V, s \in S, c \in C, n \in N$, association score, *Assoc*, between v and n with syntactic relation s is defined to be

$$Assoc_{VN}(n, s, v) = \lambda_1 P(c, s|v) + \lambda_2 P(s|v) \qquad (1)$$

The conditional probability, $P(c, s|v)$ measures the strength of the statistical association between the given verb, v and the class of the noun, n with the given syntactic relation, s. That is, it favors those that have more co-occurrences of the classes of nouns and syntactic relations for verbs. As mentioned before, we use the verb-postposition collocation to back off the $P(n, s|v)$. $P(s|v)$ that means how much the verb requires the given syntactic relation. The number of the pairs is about 240,000 which reach to 12% of the number of verb-noun-postposition triples. The λ_1 and λ_2 are set up by experiments.

3.2. Making GAT

The association value of two eojeols is recorded in the GAT only when the eojeols have a dependency relation. The association is represented by a pair, ⟨*association-value, syntactic-relation*⟩. The association value is calculated by the formula (2) and (3) described in the previous section. The parser uses the value to resolve the structural ambiguities of verbs and nouns. If the unknown syntactic case occurred in noun and verb relations, the candidates are recorded in the GAT with the possible syntactic relation. It is easy to find out the syntactic relation from the formula. Several candidates for the grammatical case is written in the GAT[i,j] when the unknown grammatical case occurred. Three candidates are good for Korean because the maximum three complements can be subcategorized by the head in general cases. The GAT is sorted by the association to look up the most probable phrase in the parsing process. Thus, the global association table is implemented by the global association list.

The following example is represented by (Table 2),

ex 1) (0) 공룡에(dinosaur) (1) 대한(for) (2) 자료를(data) (3) 가진(to have) (4) 화일을(file) (5) 지금(now)
(6) 찾아라(find)
→ Find the files that have the data for dinosaur now

In (Table 2), the cells which are marked with '-' mean that two eojeols don't have any dependency relation. It is most likely that the first eojeol has the possibility to have the dependency relation to the second eojeol. The GAT gives the association to the parser while parsing. The unknown case occurred in the relative clause caused by the fourth eojeol. The omitted postposition is presumed by the GAT. The table indicates that it is most probable that '화일(file)' was moved out of the object of the former clause. That is, the eojeol, '화일(file)' can be the object of the eojeol, '가지다(have)'. In addition, it may be moved out of the subject of the relative clause by the GAT. Therefore, the parser checks both possibilities. If it's all

	0	1	2	3	4	5	6
0	-	(0.08,에)	-	(0.01,에)	-	-	(0.01,에)
1	-	-	(0.08,ϕ)	-	-	-	-
2	-	-	-	(0.07,를)	-	-	(0.04,를)
3	-	-	-	-	(0.06,를) (0.02,가)	-	- -
4	-	-	-	-	-	-	(0.06,를)
5	-	-	-	-	-	-	(1, ϕ)
6	-	-	-	-	-	-	-

Table. 2: The global association table(GAT) for the example sentence, ex 1

right that the nominal eojeol is the object of the predicative eojeol, two eojoels are merged in the objective relation. However, the alternative(subject movement) is checked if it makes an erroneous result.

4. Parsing Algorithm

4.1. Parsing Algorithm

The parsing is directed by the following three operation in the stack and input buffer. Basic operations are CREATE, ATTACH, and DROP. However, its operation is conditioned not by rule matching but by the lexical association of the GAT as shown in the following description.

CREATE If the most probable candidate for the head of the eojeol, e_i, is e_j, that is, $j = index(max(G(i)))$, then merge e_i or the phrase including e_i with e_j or the phrase including e_j, and generate a new phrase.

ATTACH If the e_j is not the most probable candidate for the head of the eojeol e_i, that is, $j \neq index(max(G(i)))$ then wait until e_i meets the most probable candidate indicated by the GAT.

DROP DROP operation is accompanied with CREATE operation in our system because binary grammar is constructed for Korean and thus the binary relation between words is considered.

The GAT gives the parser the prediction of the best candidate, here expressed by the function, $index(max(G(i)))$, which returns the eojeol index of the most probable candidate for the head of the ith eojeol, e_i. When the new node is generated, the unknown grammatical case is recovered, if any. In case that a nominal eojeol has the unknown case caused by the auxiliary postposition or the omission of the postposition, the parser tries to identify the grammatical case. When the noun phrase is moved out of the relative clause, both the moved noun phrase and its grammatical case have to be identified from the predicative eojeol of the clause. The parser turns out the postpostition of the moved NP with the item given by the GAT.

4.2. Parsing

(Figure 1) represents the analysis steps of the sentence in (ex 1). In the fifth row of the figure, the ATTACH operation is executed by the GAT in (Table 2), because the lookahead is not the candidate for the head of the complement on the stack top. Thus the eojeol, '자료를', has to wait until it meets its best candidate. The eojeol, '찾아라' is the best candidate for the eojeol, '자료를', which was estimated by the GAT.

The CREATE operation executed because it is most probable that the eojeol, '가진' is dependent on the next eojeol, '자료를'. The unknown grammatical case is identified in the fourth row because the predicative eojeol is relativized by virtue of the adnominal ending. First, the moved constituent is assumed as the object of the clause by virtue of (Table 2). However, the parser recognizes that the object has been already governed by the predicatve eojeol. Thus it tries for the alternative, that is, the second item of GAT(3,4). The grammatical case is subjective by the given association.

	OP	Stack Top		First Lookahead	
		Constituents	Head	Constituents	Head
1	A		공룡에(dinosaur)		대 한(for)
2	A	공룡에 대한	대 한(for)		자료를(data)
3	A	공룡에 대한 자료를	자료를(data)		가진(to have)
4	A,C	공룡에 대한 자료를 가진	가진(to have)		화일을(file)
5	B	공룡에 대한 자료를 가진 화일을	화일을(file)		지 금(now)
6	A	지금	지 금(now)		찾아라(find)
7	A	공룡에 대한 자료를 가진 화일을	화일을(file)	지금 찾아라	찾아라(find)

Fig. 1: an example of analyzing the sentence in (ex 1). OPs are A: Create & Drop operation, B: Attach operation, C: Identification of the unknown case

	Brackets of noun and verb	Correct Brackets	%
result	1727	1595	92.4

Table. 3: experimental results for structural ambiguity resolution

5. Experiment Results

We report the result of analyzing 408 sentences, which were separated from the training corpus. Two kinds of tests have been executed to estimate the resolution of ambiguities. First, a complement has several candidates for its head. To ensure that our method is effiective, the experiment was conducted for the case that the complement is the nominal eojeol and the candidate for the head, the predicative eojeol. (Table 3) shows the accuracy of the structural ambiguity resolution.

Second, the identification of the unknown cases was checked. The results are represented in (Table 4). To improve the accuracy of the system, the parser has to consider linguistic knowledge. The movement of the NP in the relativization is the linguistic phenomenon and all NP cannot be moved.

References

Allen, J. 1995. *Natural Language Understanding.* Benjamin Cummings.

Collins, M. J. 1996. *A New Statistical Parser Based on Bigram Lexical Dependencies* In *Proceedings of 34th Annual Meeting of Association for Computational Linguistics.*

Hindle, D. and Rooth, M. 1993. *Structural Ambiguity and Lexical Relations* Computational Linguistics

Kobayasi Y., Tokunaga T., and Tanaka H. 1994. *Analysis of Japanese Compound Nouns using Collocational Information* In *Proceedings of COLING-94.*

Lim, H. 1992. The Research for Classification of the Korean, (*in Korean*). National Language Research Institute., 1992

Marcus, M. 1980. *A Theory of Syntactic Recognition for Natural Language.* Cambridge, MA: MIT Press.

Total number of unknown cases	Success		Failure	
378	302	80%	76	20%

Table. 4: The results of the identification of unknown cases

The Description of the Intra-State Feature Space in Speech Recognition

Fang Zheng, Mingxing Xu, Wenhu Wu
Speech Lab, Dept. of Comp. Sci. & Tech., Tsinghua Univ., Beijing 100084, China
fzheng@cenpok.net, [fzheng, xumx]@sp.cs.tsinghua.edu.cn

Abstract

In speech recognition, the description of the intra-state feature space is an important issue in systems based on HMM-derived acoustic models. The existing techniques include the famous methods based on VQ technique and mixture Gaussian densities. In this paper, a method based on sub-space division is proposed. Experiments are done to find how many densities should be used to better describe the intra-state feature space, and the experimental results show that the number of densities should depend on the particular distribution of that space and can be judged by a kind of criterion.

1. Introduction

In speech recognition, how to describe or represent the intra-state feature space for a HMM-based system is an important problem.

In general, there is an assumptions for traditional HMM, i.e., the current observation depends only on the current system state, which indicates that the observation output of the intra-state is independent and identical distributed (***i.i.d.***). Though it is simple, the results are not bad, which introduces discrete HMM (DHMM), continuous density HMM (CDHMM), semi-continuous HMM (SCHMM) [Huang 1989] and some other similar models.

According to the above assumption, the intra-state feature space can be described according to the Theory of Probability. The common used approaches include Mixture Gaussian Densities (**MGD**) [Wilpon 1989, Huang 1989] and tied Mixture Gaussian Densities (**TMGD**) [Bellegarda 1990]. Zheng *et al* [Zheng 1996, 1997] describes the feature space by sub-space division (**SSD**) method.

No matter what kind of method is adopted, there is an important problem to solve, that is, how many probability density functions (**PDFs**) should be used to better describe the space when the maximum number of densities is limited ? According to the Theory of

Probability, the more Gaussian densities, the better the space is approached. Due to the limitation of computer processing, a suitable number of densities must be determined.

In that case, should the same density number be chosen for every speech recognition unit (**SRU**) ? The answer is no. In this paper, the experimental results will be given based on the CDCPM [Zheng 1996].

2. Sub-Space Division Method

The sub-space division (**SSD**) method is to divide the whole space into several independent sub-space by a certain criterion. If considering the probability distribution of a specified space, the scoring of a feature vector o_t at time *t* is given by

$$b(o_t) = \sum_{m=1}^{M} g_m f_m(o_t) \tag{1}$$

where $b(\cdot)$ is the PDF of the whole space, $f_m(\cdot)$'s are the *M* Gaussian PDFs, and g_m's are the corresponding weightings of the PDFs. If using the SSD method, the scoring equation is

$$b(o_t) = \max_{1 \le m \le M} f_m(o_t) \tag{2}$$

(Notice that $f_m(\cdot)$'s in Eq. (2) are often different from those in Eq. (1).) That is to say, the score of a feature vector is defined as the matching score with the closest sub-space. We found this kind of scoring scheme is identical to the human's cognition.

2.1 Considerations in the SSD method

In general, when dividing the space into sub-spaces in speaker-independent speech recognition tasks, the following factors will be considered: (1) gender-dependent (GD) information; (2)accent-dependent (AD) information; (3) speaker-dependent (SD) Information; (4) context-dependent (CD) information; (5) background noise (BN) information; and so on.

All these factors make the situation more complicated, it will cost much more model storage and database labeling even if only one or two of these factors are considered. It sounds not practical.

2.2 The motivation of the automatic SSD (ASSD) method

Actually, the feature space of the same SRU uttered by different speakers in different accents and different contexts has many common areas, as illustrated in Fig. 1.

In Fig. 1, symbols 1, 2, and 3 stand for three different speech sources, any two of them have a common region in the whole space. (It maybe more complicated actually.) Obviously, if we divide the feature spaces individually, then 4 densities are needed for both sources 1 and 3 while 3 densities for source 2, and totally 11 densities are needed. But actually, there are only 6 different densities. So if there are many speech sources, a great deal of redundant density storage are cost for individual space division and description.

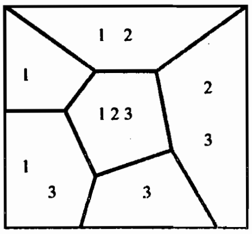


Fig. 1 The feature space of three different speech sources

The automatic SSD (ASSD) method tries to find as fewer common regions as possible for the whole feature space of several different speech sources.

2.3 Clustering: one ASSD method

As a matter of fact, many existing clustering methods can be used for sub-space division, such as LBG [Linde 1980], K-means [Furui 1989], and simulated anneal [Xu 1989] algorithms. But it is not too easy to decide how many densities should be used. A criterion will be given in this paper based on within- and between-class scatter degrees.

First of all, we will define the within-class scatter degree (**WCSD**) and between-class scatter degree (**BCSD**).

Assume there are N feature vectors totally in a space. In M'th clustering iteration, M classes (sub-spaces) are generated, and in Class m $(1 \le m \le M)$, the mean vector of the N_m vectors $\{x_m^{(i)}: 1 \le i \le N_m\}$ is μ_m. Denote the distance measure between two vectors by $y(\cdot,\cdot)$. Define average within-class scatter degree of Class m as

$$\tilde{d}_{wm} = \frac{1}{N_m}\sum_{i=1}^{N_m} y(x_m^{(i)}, \mu_m) \tag{3}$$

and the total average within-class scatter degree as

$$\tilde{d}_w = \frac{1}{N}\sum_{m=1}^{M} N_m \cdot \tilde{d}_{wm} = \frac{1}{N}\sum_{m=1}^{M}\sum_{i=1}^{N_m} y(x_m^{(i)}, \mu_m). \tag{4}$$

Define the average between-class scatter degree as

$$\tilde{d}_b = \frac{1}{M \times (M-1)}\sum_{m=1}^{M}\sum_{\substack{m1=1 \\ m1 \ne m}}^{M} y(\mu_m, \mu_{m1}) = \frac{2}{M \times (M-1)}\sum_{m=1}^{M}\sum_{m1=m+1}^{M} y(\mu_m, \mu_{m1}) \tag{5}$$

After the definitions of the within-class and between-class scatter degrees, the criterion function is define as

$$J_d(M) = \frac{\tilde{d}_w \cdot f(M)}{\tilde{d}_b} \tag{6}$$

where $f(\cdot)$ is a strictly increasing function.

In general, $\tilde{d}_w / \tilde{d}_b$ is a decreasing function of M. Consider the special case when M is equal to the number of all feature vectors, where there is only one vector in every class, so $\tilde{d}_w / \tilde{d}_b$ will decrease to 0. The increasing function $f(\cdot)$ is used as a penalty function to avoid unreasonable larger class number.

In the speech recognition system, a maximum value of class number or sub-space number is often given at the very beginning, say $M_{\max}$. So in this case,

$$M_s = \arg\min_{2 \le M \le M_{\max}} J_d(M) \tag{7}$$

is chosen as the suitable class (sub-space) number.

3. Database Description

A great deal of experiments have been done across a real-world spontaneous database. The speech data are taken from telephone network and sampled at 8KHz. The samples are 13-bit linear PCMs expanded from A-law codes. The database consists of speech data uttered by 200 people, and the amount is about 4GB. 10th order LPC-based cepstral (LPCC) analysis is performed on 32 ms speech window every 16 ms. Auto-regressive analysis is also performed on 5 adjacent frames of LPCC vectors. The LPCCs and their corresponding auto-regressive coefficients are the features used for the CDCPM [Zheng 1996, 1997] in this paper.

The SRUs are 419 Chinese syllables.

4. Experimental Results

The experimental results are given in Tab.1, where the number of states (NOS) ranges from 3 to 6. In the number of densities (NOD) column, “Fix” stands for using the maximum number of densities, i.e., 16, while “Var” stands for using different number of densities for different syllables, but the maximum density number is 16.

From Tab. 1, we can draw the conclusion for any number of states for the CDCPM: choosing different density number for different syllables is better than using fixed density number, the former scheme can improve the system by 1.6% when NOS is 6.

Tab.1 Experiment on choosing the intra-state number of densities

Top *n* candidates NOS	NOD	1	2	3	4	5	6	7	8	9	10
3	Fix	69.44	77.94	82.05	84.70	86.51	87.95	89.14	90.04	90.72	91.39
	Var	69.95	78.47	82.59	85.29	87.11	88.54	89.75	90.64	91.34	92.02
4	Fix	74.03	81.42	85.03	87.34	88.95	90.02	90.96	91.79	92.49	92.99
	Var	74.90	82.32	85.96	88.24	89.87	90.94	91.84	92.68	93.41	93.92
5	Fix	77.41	83.79	86.68	88.44	89.75	90.85	91.56	92.20	92.79	93.39
	Var	78.66	85.07	87.95	89.73	91.04	92.13	92.85	93.48	94.08	94.67
6	Fix	78.86	85.12	87.84	89.40	90.71	91.41	92.12	92.75	93.27	93.67
	Var	80.46	86.71	89.45	91.03	92.32	93.03	93.77	94.40	94.91	95.31

5. Conclusion

In this paper, we study the description methods for intra-state feature space based on HMM-derived acoustic models. The experiments are done for choosing the suitable number of sub-spaces. The experimental results show that using the same number of densities to describe every state of every SRU performs worse, different state of different syllable should have different density number according to a reasonable criterion. Studies and experiments are focused on the SSD scheme, we think the conclusion is also right for the MGD method.

6. References

[1] **Bellegarda, J.R., Nahamoo, D.**, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. on ASSP*, vol.ASSP-38, No.12, pp.2033-2045, Nov. 1990

[2] **Linde, Y., Buzo, A., Gray, R.M.**, "An algorithm for vector quantization," *IEEE Trans. On COM*, 28(1), Jan. 1980

[3] **Furui, S.**, Digital Speech Processing, Synthesis and Recognition, *Marcel Dekker, Inc.*, 1989

[4] **Huang, X.-D., Jack, M. A.**, "Semi-continuous hidden Markov models for speech signals," *Computer Speech and Language*, 3:239-251, 1989.

[5] **Wilpon, J.G., Lee, C.-H., Rabiner, L.R.**, "Application of hidden Markov models for recognition of a limited set of words in unconstrained speech," *ICASSP-89*, 3: 254-257

[6] **Xu, L.**, "A kind of new clustering method: Simulated Anneal," *Pattern Recognition and Artificial Intelligence*, 2 (1), March 1989 (in Chinese)

[7] **Zheng, F., Wu, W.-H., Fang, D.-T.**, "CDCPM with its applications to speech recognition," *Chinese J. of Software*, 7: 69-75, Oct. 1996 (in Chinese)

[8] **Zheng, F., Chai, H.-X., Shi, Z.-J., Wu, W.-H., Fang, D.-T.**, "A real-world speech recognition system based on CDCPMs," *Int'l Conf. on Computer Processing of Oriental Languages (ICCPOL'97)*, Apr. 2-4, 1997, Hong Kong

Similarity Comparison between Chinese Sentences

Lina Zhou[12] James Liu[2]

[1]Institute of Computational Linguistics, Peking University
Beijing, China , 100871
cslzhou@comp.polyu.edu.hk

[2]Department of Computing, Hong Kong Polytechnic University
Kowloon, Hong Kong
csnkliu@comp.polyu.edu.hk

Abstract

Identifying and extracting similar sentences from the example base is an essential procedure in machine-aided human translation (MAHT) and example-based machine translation (EBMT) system. A method for measuring the similarity between a pair of Chinese sentences has been proposed in this paper. Obviating from the common thesaurus-based strategy, a new principle based on word grammatical features is presented thereafter. Moreover, a dynamic mechanism is built into the method to increase the robustness and flexibility of the matching algorithm. From observations on the initial results, we' ve found that the expected most similar sentence in the example base for an input is listed among the first four candidate sentences in most cases, which is very helpful for both MAHT and EBMT.

1. Introduction

It is regarded as an essential process to measure the similarity between an input sentence and the stored examples or translation candidates in machine-aided human translation system and example-based machine translation system.

There has been no accurate definition for similarity comparison available in the field of machine translation, though, it is clear that similarity comparison is a cloning process, which measures the matching scores between two objects in terms of certain similarity metric. As far as sentence pairs are concerned, the purpose of similarity comparison is to identify and extract sentences from the stored base, which are similar to the input sentence. The criteria for comparison can be based on attributes of the word, phrase structure or sentence. The most popular strategy is based on the thesaurus relation. (Furuse, 1992; Nirenburg, 1993; Sato, 1992; Cranias et. al., 1994; Maruyama, 1992; Zhang et. al. 1995). Either the performance of such method is not satisfactory or it relies much on pre-processing efforts to obtain acceptable results. As a result of trade-off among the factors influencing the accuracy and efficiency of the matching algorithm, a word grammatical attribute oriented approach is proposed for comparing Chinese sentences, which takes the following significance:

- It conforms with the most frequently used syntax-based translation techniques. The basis upon which similarity metric is built can be directly applied in the transfer stage.
- It explores the prospect of word feature oriented approach and the possibility of improving comparison results by elaborating the grammatical features of words.

In this paper, the knowledge base for the similarity metric will be presented in the next section. The new metric for measuring similarity will be described in Section 3 and followed with an algorithm in Section 4. Finally, some experimental samples are given in Section 5.

2. Grammatical Knowledge-base

The knowledge base was originated from the *Electronic Dictionary of Grammatical Information for Contemporary Chinese* (Yu et. al., 1996). Since the dictionary was designed for general-purpose applications, elaborately defined features have to be filtered or selected to facilitate the identification of the right translation candidates when applied to sentence comparison. Following the above principles, eight sub-dictionaries were employed, i.e. noun, verb, adjective, adverb, pronoun, classifier, preposition and time dictionaries, etc. The specific features helpful for sentence comparison were selected in every dictionary.

3. Similarity Metric

Based on the features defined for each category, the similarity metric between a pair of Chinese sentences (A,B) was defined as:

Assume that A = $a_1a_2\ldots\ldots a_n$, B = $b_1b_2\ldots\ldots b_m$, $a_i(b_j), 0<i<n+1, (0<j<m+1)$ is the *ith* (*jth*) word in sentence A (B). F is the whole feature set of a certain word category, E a subset of F, and |E| stands for the number of features in E. *$fea_k(a)$, sub_pos(a) and pos(a)* represent the *kth* feature, sub-category and part-of-speech of word *a* respectively. *Ss(A,B)* represents the similarity metric between A and B, while *$S_w(a_i,b_j)$* the similarity score between a_i and b_j. $a_1^i(b_1^j)$ represents the string from a_1 (b_1) to $a_i(b_j)$. L(A,B) is the normalizer for the sum of the similarity score.

$$Ss(A,B) = \frac{Ss(a_1^n, b_1^m)}{L(A,B)} \tag{1}$$

$$Ss(a_1^i, b_1^j) = \begin{cases} 0, \text{ } if \text{ } i<1 \cup j<1 \\ Ss(a_1^{i-1}, b_1^{j-1}) + Sw(a_i, b_j), \text{ } if \text{ } i>1 \cap j>1 \cap Sw(a_i,b_j)>0 \\ Ss(a_1^i, b_1^{j-k}), \text{ } else \text{ } if \text{ } j>k>1 \cap Sw(a_i, b_{j-k})>0 \\ Ss(a_1^{i-1}, b_1^j), \text{ } otherwise \end{cases} \tag{2}$$

$$Sw(a_i,b_j) = \begin{cases} 0 & if \text{ } pos(a_i) \neq pos(b_j) \\ 0.25 & else \text{ } if \text{ } sub\_pos(a_i) \neq sub\_pos(b_j) \\ 0.5 & else \text{ } if \bigcup\limits_{\substack{k \in E \\ E \subset F \\ |E| \leq 0.5*|F|}} fea_k(a_i) = fea_k(b_j) \\ 0.8 & else \text{ } if \bigcup\limits_{\substack{k \in E \\ E \subset F \\ 0.5*|F| < |E| < |F|}} fea_k(a_i) = fea_k(b_j) \\ 1 & else \end{cases} \tag{3}$$

In contrast with the common static definition for *L(A,B)*, a new and dynamic formula is given thereafter:

$$L(A,B) = N(n,m) + F(n,m)/3 \tag{4}$$

where *N(n,m)* is the number of comparison times; while *F(n,m)* is the number of words failed to get matched. It could be in some cases that the algorithm doesn' t provide the optimum matching for an input sentence. However, the adoption of a dynamic mechanism does ensure better efficiency for matching, and *F(n,m)* is introduced as a penalty factor to improve the results.

4 Algorithm

For an input sentence A= $a_1a_2\ldots\ldots a_{n+1}$ [1]and a stored sentence B=$b_1b_2\ldots\ldots b_{m+1}$,

[1] Here an extra word is appended to indicate the end of the sentence.

```
1.  Initialize $i = 1, j = 1; t = 0, f = 0;$
2.  While $a_i \neq a_{n+1}$
        if  $b_j \neq b_{m+1}$
            if  $S_w(a_i, b_j) < 0.25$
                $j_0 = j$
            else
                $i = i + 1$
                $Sum = Sum + S_w(a_i, b_j)$
            endif
            $j = j + 1$
            $t = t + 1$
        else
            $j = j_0$
            $f = f + 1$
            $i = i + 1$
        endif
4.  $S_s(A,B) = \frac{Sum}{t + f/3}$
```

5. Experimental Samples

A parallel bilingual corpus with about 3,000 Chinese and English sentence pairs has been utilized and pre-processed (Zhou and Liu, 1997). Both sides have been annotated with part-of-speech.

The testing results are classified into five categories: complete match, word replacement, word insertion and deletion, phrase replacement and modification, and composition, etc. Several samples are provided for explanation:

(1) Word Replacement

In: 東京是世界上 人口最多的城市[2]。

Re: 上海/n 是/v 世界/n 上/f 最/d 大/a 的/u 城市/n 之一/m 。/w (Shanghai is among the largest cities in the world.) <0.85>

[2] The focus part in each category is underlined. The similarity score of the result (Re) for the input (In) is put in the brackets.

(2) Phrase Replacement and Modification

In: 我有足夠的錢買書・

Re: 我/r 有/v 足夠/v 的/u 錢/n 買/v 這/r 兩/m 本/q 書/n ・/w (I have enough money to buy these two books.) <0.89>

5. Conclusion and Future Directions

A new algorithm for measuring the similarity between a pair of Chinese sentences has been proposed in this paper. It emphasizes the grammatical features of Chinese words supported by the comprehensive electronic dictionaries. In addition, a dynamic mechanism and penalty score are built in to increase the robustness and flexibility of the algorithm. From observation on the matching results, we feel that most of the selected sentences are much related with the input on the syntactic and even semantic level. The expected most similar sentence from the example base is listed among the first four candidate sentence s in most cases, which is considered to be very informative and helpful for both MAHT and EBMT. In view of the basic ideas introduced, the approach is easy to be tested on other languages.

References

Furuse, O. and H. Iida, "An Example-based Method for Transfer-driven MT", in TMI' 92, , 1992, pp. 139-148.

Nirenburg, S. "Two Approaches to Matching in EBMT", in TMI'93 , 1993, pp. 47-57.

Sato, S. "CTM: An Example-based Translation Aid System", in COLING'92 , 1992, pp. 1259-1263.

Cranias, L. et. al., "A Matching Technique in Example-based Machine Translation", in COLING' 94, 1994, pp.100-104.

Maruyama, H. "Tree Cover Search Algorithm for EB Translation", in TMI'92. 1992, pp. 173-184.

YU, S.W., Y.F. Zhu, H. Wang and Y.Y. Zhang. "Specification for Grammatical Information Dictionary of Contemporary Chinese", in Journal of Chinese Information Processing, Vol. 10,No.2, 1996, pp.1-15.

Zhou, L.N. and J. Liu, "Extracting More Word Translation Pairs from Small-sized Bilingual Parallel Corpus: integrating rule and statistics-based method", in ICCPOL'97. 1997, pp.250-255.

Zhang, M., S. Li, T.J. Zhao and M. Zhou. "An Algorithm for Calculating the Similarity between Chinese Sentences and its Application", in the Third National Joint Conference on Computational Linguistics, Tsinghua University Press, 1995, pp. 152-158.

Attributive Clauses in Chinese: Theory and Implementation

Xiaokang Zhou, University of Melbourne, Australia

Francis Y. Lin, Oxford University, UK

Abstract

This paper concerns attributive clauses in Mandarin Chinese - ie. clauses which assign properties to some entity. It identifies the semantic and grammatical range of these clause types, providing a network representation of their paradigmatic relatedness. This network provides the semantic basis for the generation of feature clusters which - via realisation rules and potential structures - output the target clauses.

1. Introduction

In this paper we present a lexicogrammar of attributive clauses in Chinese. The theory adopted is Systemic Functional Grammar (SFG), according to which grammar is a set of choices in meaning, which can be arranged in the form of system networks, mediating the various functions that language serves (Halliday 1985, Fawcett 1987).

2. Types of Participant Roles in Attributive Clauses

Attributive clauses represent relationships involving an entity and some other entity which might be a thing or quality. They realise processes which assign an attribute, one of the participant roles associated with the process, to another participant role, 'Carrier', where 'process' refers to actions, events, states and relations (Halliday 1970:146). Processes are defined in terms of Participant Roles(PRs) and their configurations. Four PRs are relevant to this part of the grammar: Carrier (thing that displays a property), Attribute (property displayed), Agent (doer, actor) and Affected (patient, goal). Tests have been developed for these roles, but it is beyond scope of this work to discuss them (see Zhou in preparation).

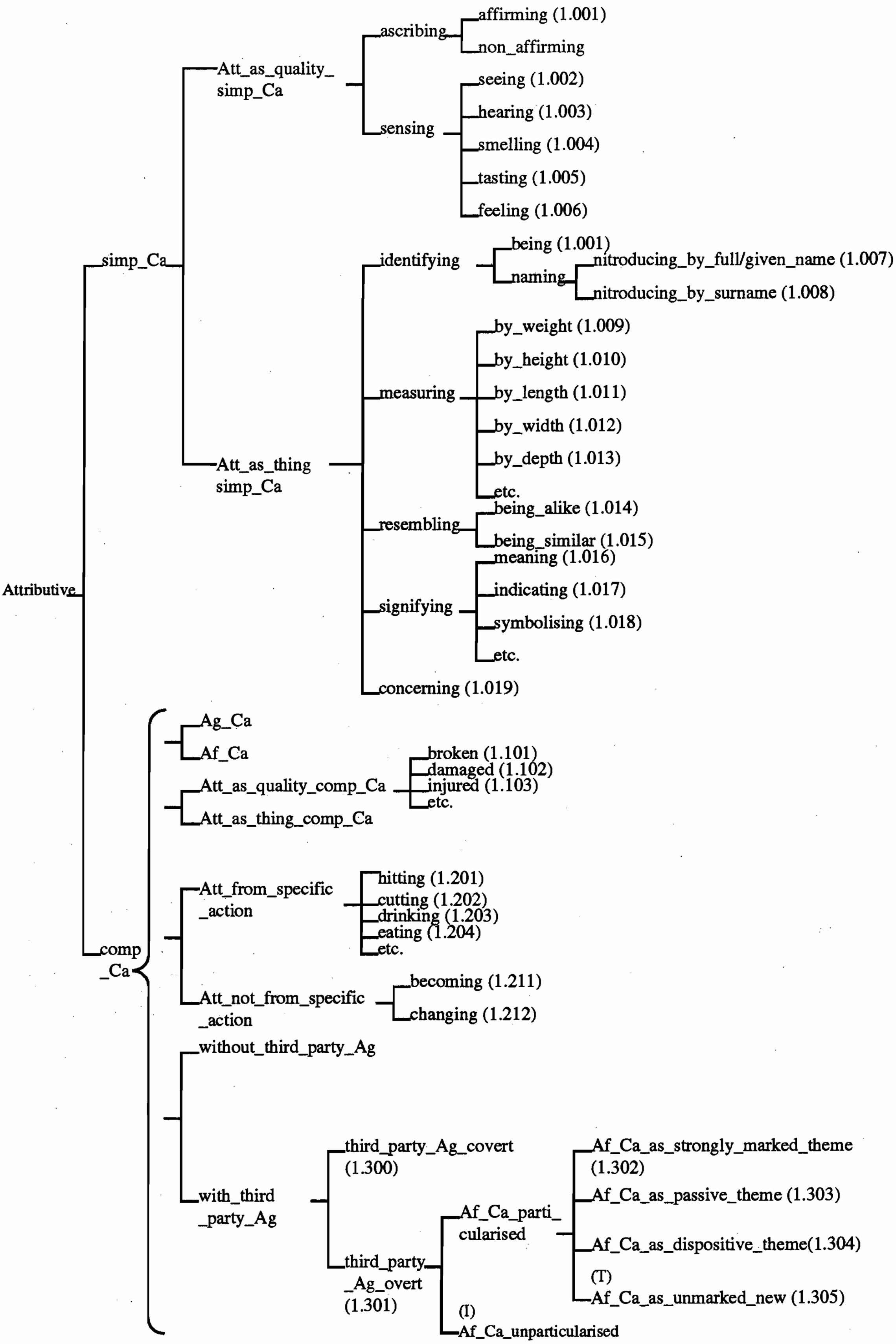


Fig. 1 A System Network for Attributive Clauses in Chinese

3. Network Representation of Attributive Clauses in Chinese

Figure 1 is a network representation of attributive clauses in Chinese. Observe that a primary distinction is made between 'simple Carrier attributive' and 'compound Carrier attributive' process types, according to whether the role of Carrier is conflated with the Agent or Affected. For detailed discussion of the various oppositions in the network (see Zhou in preparation).

4. Implementation

The implementation is based on the sentence generator developed in the COMMUNAL Project at the University of Wales, Cardiff, UK (see Fawcett, Tucker and Lin (1993) and Lin, Fawcett and Davies (1993)).

The implementation requires two things in addition to the network: potential structures and realisation rules. Potential structures characterise grammatical units in terms of the range of possible items that constitute them, and their order; they provide order-class characterisations. A simplified potential structure of a Chinese clause relevant to attributive processes is (elements which can occur in more than one places, such Complement, are not specfied in the potential structure - they are specified by the relevant realisation rules):

unit ('Cl') : S @ 10, Atp @ 11, Xcon @ 19, M @ 23, Cr @ 26, Xperf @ 28[1]

There are five major types of realisation rule in the grammar:

1. **Unit insertion rules**, which add a syntactic unit to the existing sentence structure. For example, if one selects the feature [situation] in the network, the unit 'Clause' will be generated. The realisation rule which does this is:

1 : situation : 'Cl'.

2. **Componence rules**, which add elements into the potential structure, and put them in their correct places. For example, the features [Af_Ca_unmarked_new] and [Af_Ca_dispositive_Theme] are associated with realisation rules as follows:

[1] Key to symbols: Z = sentence, Cl = Clause, S = Subject, Atp = Time-position Adjunct, Xcon = Aspect:Continuous, M = Main Verb, Cr = Resultative Complement, Xperf = Aspect:perfective, C = Complement, Ag = Agent, Af = Affected, Ca = Carrier, ngp = nominal group,mrgp = minimal relationship with thing group (preppsitional group), qlgp = quality group, cv = completive.

```
1.305 : Af_Ca_unmarked_new :
        'C2' @ 32,
        'Af_Ca' by 'C2',
        for 'Af_Ca' prefer [thing],
        for 'Af_Ca' re_enter_at entity.

1.304 : Af_Ca_dispositive_Theme :
        'C2' @ 22,
        if attribute_as_thing then,
        for 'C2' prefer [thing, minimal_relationship_with_thing, dispositive_marker],
        for 'C2' re_enter_at entity.
```

Rule 1.305 puts 'C2' (Second complement = direct object) at place 32, which is after the Main Verb, whereas rule 1.304 puts it at place 22, ie. before the Main Verb.

3. **Conflation rules** place PRs (such as a Carrier) by elements, e.g. 'Ca' by 'S', or 'Af_Ca' by 'C2' (see rule 1.305 above). Thus participant roles are present in the structural output from the generator.

4. **Exponence rules** state that an element is expounded by a lexical item. For example, M < "*da*'" says that the main verb is expounded by the word "*da*" 'hit'. Intonation can also be generated by exponence rules.

5. **Re-entry rules** make it possible for the generator to traverse a network a second time. For example, rule 1.305 has a re-entry rule as a part, which says that to generate the content of 'C2' we must re-enter the network (and make more choices).

5. An Example: Generating a Compound-Carrier Attributive Clause

The target clause is *Wo ba beizi da po le* 'I broke the mug'. To generate this clause, the following choices are made: [comp_Carrier_attributive], [Affected_Carrier], [Attribe_as_quality_comp_Carrier], [Attribte_from_specific_action], [hitting], [broken], [with_third_party_Agent], [third_party_Agent_overt], [Affected_Carrier_particularised], [Affected_Carrier_as_dispositive_theme]. The realisation rules attached to the relevant features are next applied, building the following structure after the first pass:

```
Z|Cl ->       S/Ag
     ->       C
     ->       M < “da”
     ->       Cr
     ->       Xperf < “le”
```

After the second pass, the following structure is generated, giving the target clause as output: *wo ba beizi da po le* ‘I broke the cup’ (Due to space limitation, details have been glossed over):

```
Z|Cl ->       S/Ag|ngp ->   h < “wo”
     ->       C|mrgp   ->   mr1 < “ba”
     ->       cv|ngp   ->   h < “beizi”
     ->       M < “da”
     ->       Cr|qlgp  ->   a < “po”
     ->       Xperf < “le”
```

6. Conclusion

This paper has provided an SFG description of attributive clauses in Mandarin Chinese. SFG offers a rich framework for describing and analysing various clause types and provides an understanding of their richness as well as a means of generating clauses from a system network. We have been able to identify a range of clause types which otherwise remain obscure. This investigation has shown that SFG offers a useful and insightful set of strategies for computational linguistics.

References

Fawcett, R.P. 1987. ‘The Semantics of Clause and Verb for Relational Processes in English’. In M.A.K. Halliday and R.P. Fawcett (eds.), *New Developments in Systemic Linguistics*. London: Frances Pinter.130-183.

Fawcett, R. P., Tucker, G. H. and Lin, F. Y. 1993. ‘The role of realisation in realisation: how a systemic functional grammar works’. In Horacek, H. and Zock, M. (eds) From Planning to Realisation in Natural Language Generation, pp. 114-86. Pinter, London.

Halliday, M.A.K. 1985. *An Introduction to Functional Grammar.* London: Edward Arnold.

Lin, F. Y., Fawcett, R. P. and Davies, B. L. 1993. ‘GENEDIS: the discourse generator in COMMUNAL’. In A. Sloman et al (eds.) Prospects for Artificial Intelligence, pp. 148-57. IOS Press, Amsterdam.

Zhou, X.K. in preparation. Material and Relational Transitivity in Mandarin Chinese, Ph.D dissertation.