A Statistically Emergent Approach for Language Processing: Application to Modeling Context Effects in Ambiguous Chinese Word Boundary Perception

Kok-Wee Gan*
Hong Kong University of Science and Technology

Martha Palmer†
University of Pennsylvania

Kim-Teng Lua‡
National University of Singapore

This paper proposes that the process of language understanding can be modeled as a collective phenomenon that emerges from a myriad of microscopic and diverse activities. The process is analogous to the crystallization process in chemistry. The essential features of this model are: asynchronous parallelism; temperature-controlled randomness; and statistically emergent active symbols. A computer program that tests this model on the task of capturing the effect of context on the perception of ambiguous word boundaries in Chinese sentences is presented. The program adopts a holistic approach in which word identification forms an integral component of sentence analysis. Various types of knowledge, from statistics to linguistics, are seamlessly integrated for the tasks of word boundary disambiguation as well as sentential analysis. Our experimental results showed that the model is able to address the word boundary ambiguity problems effectively.

1. Introduction

This paper suggests that the language understanding process can be effectively modeled as the statistical outcome of a large number of independent activities occurring in parallel. There is no global controller deciding which processes to run next. All processing is done locally by many simple, independent agents that make their decisions stochastically. The system is self-organizing, with coherent behavior being a statistically emergent property of the system as a whole. The model, in a nutshell, simulates language understanding as a crystallization process. This process consists of a series of hierarchical, structure-building activities in which high-level linguistic structures are formed from their constituents and get properly hooked up to each other as the process converges.

The essential features of the model are:

- The process of sentence analysis is a series of computational activities that determine how various constituents in a sentence can be meaningfully related.

* Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
† Department of Computer Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389
‡ Department of Information Systems & Computer Science, National University of Singapore, Lower Kent Ridge Road, Singapore 119260, Republic of Singapore

- All computational activities are carried out by a large number of procedures known as **codelets**.
- A linguistic structure is not built by a single codelet. Rather, it is constructed by a sequence of codelets. The execution of this sequence of codelets is interleaved with other codelets that are responsible for building other structures.
- The order by which structures are built is not explicitly programmed, but is an emergent outcome of chains of codelets working in an asynchronous parallel mode.
- Computational activities are a combination of top-down and bottom-up activities.
- Computational activities are indirectly guided by a semantic network of linguistic concepts, which ensures that these activities do not operate independently of the system's representation of the context of a sentence.
- Decision making is stochastic, with the amount of randomness being controlled by a parameter known as the **computational temperature**.

We have applied our model to the task of capturing the effect of context on the perception of ambiguous word boundaries in Chinese sentences (Gan 1993). Our approach differs from existing work on Chinese word segmentation (Liang 1983; Wang, Wang, and Bai 1991; Fan and Tsai 1988; Chang, Chen, and Chen 1991; Chiang et al. 1992; Sproat and Shih 1990; Wu and Su 1993; Lua and Gan 1994; Lai et al. 1992; Sproat et al. 1994; Sproat et al. 1996) primarily in that our system performs sentence interpretation, in addition to word boundary identification. Our system figures out where the word boundaries of a sentence are by determining how various constituents in a sentence can be meaningfully related. The relations the system builds represent its interpretation of the sentence. In the initial stage of a run, the system constructs relations between characters of a sentence. Through a spreading activation mechanism, the system gradually shifts to the construction of words and of relations between words. Later, the system progresses to identifying and constructing chunks (in other words, phrases), and to establishing connections between chunks. Note that there is no top-level executive that decides the order of these activities. At any given time, the system stochastically selects one action to execute. Therefore, efforts toward building different structures are interleaved, sometimes cooperating and sometimes competing. The system's high-level behavior, therefore, arises from its low-level stochastic actions.

We will give a detailed description of this application in this paper. In Section 2, we introduce the problem of ambiguous Chinese word boundary perception, and follow, in Section 3, with a summary of the current practices in Chinese word identification. We describe our model in Section 4, showing a sample run of our program in Section 5 to illustrate the behavior of the model. Finally, some discussions of the model are covered in Section 6. In Section 7, we compare our model with others, and explore areas for future research in Section 8.

2. Ambiguous Chinese Word Boundary Perception

A written Chinese sentence consists of a series of evenly spaced Chinese characters. Each character corresponds to one syllable. A word in Chinese can be made up of a single character, such as 飯 *fàn* 'rice', or it can be a combination of two or more characters, such as 水果 *shuǐguǒ* 'fruit'. It is possible that the component characters of a word are free[1], such as 水 *shuǐ* and 果 *guǒ* of the word 水果 *shuǐguǒ* 'fruit', which mean 'water' and 'fruit' respectively. For any two Chinese characters in a sentence, denoted as *x* and *y*, if *xy* cannot be combined together to function as a word, a single word boundary exists between these two characters. If *x* and *y* can be constituents of the same word, yet at the same time may also be free, then word boundary ambiguity exists in these two characters. If there is a unique word boundary before *x* and after *y*, we refer to the ambiguity existing in *xy* as a **combination ambiguity**. On the other hand, if there is a word boundary ambiguity between the characters *xy* and the character that precedes or follows them, say *z*, and these three characters can be grouped into either *xy z* or *x yz*, then we say that an **overlap ambiguity** exists. A sentence that allows an ambiguous fragment to have multiple word boundaries will end up with more than one interpretation. This type of ambiguity is called **global ambiguity** with respect to the sentence. On the other hand, if only one way of segmenting the word boundary of an ambiguous fragment is allowed in a sentence, we call this **local ambiguity** with respect to the sentence. Global ambiguity can only be resolved with discourse knowledge. An example for each category is shown in (1) to (4).[2] Throughout this paper, we follow the guidelines on Chinese word segmentation adopted in China.[3]

Overlap, Local Ambiguity

(1)	這	位	職員	工作	的	壓力	很	大
	zhè	*wèi*	*zhíyuán*	gōngzuò	de	yālì	hěn	dà
	this	CL[4]	worker	work	STRUC[5]	pressure	very	great

'This worker faces great pressure in his work.'

The underlined fragment 員工作 *yuán gōngzuò* in (1) has overlap, local ambiguity. The middle character 工 *gōng* can combine with the previous character 員 *yuán* to form the word 員工 *yuángōng* 'worker', leaving the third character functioning as a monosyllabic word 作 *zuò* 'do'. The middle character can also combine with the next character to form the word 工作 *gōngzuò* 'work', leaving the first character alone. The sentence containing this fragment allows only one way of segmenting the word boundary, which is shown in (1). The character 員 *yuán* combines with the character preceding it, 職 *zhí*, to form the bisyllabic word 職員 *zhíyuán* 'worker', and the two characters 工 *gōng* and 作 *zuò* form a word.

Overlap, Global Ambiguity

(2)a.	我們	要	學生	活	得	有	意義
	wǒmen	*yào*	*xueshēng*	*huó*	dé	yǒu	yìyì
	we	want	student	live	CSC[6]	have	meaning

'We want our students to have a meaningful life.'

[1] A free character is one which can occur independently as a word (Li and Thompson 1981).

[2] The characters underlined in sentences (1) to (4) are the locations of word boundary ambiguities we would like to focus on. This convention will be used throughout in this paper.

[3] See *Contemporary Chinese Language Words Segmentation Standard Used for Information Processing*, fifth edition, 1988, published in China.

[4] CL stands for a CLassifier.

[5] STRUC stands for the STRUCture word 的 *de*.

[6] CSC stands for the Complex Stative Construction word 得 *de*.

b. 我們 要 學 生活 得 有 意義
wŏmen yào xué shēnghuó dé yŏu yìyì
we want learn life CSC have meaning
'We want to learn how to lead a meaningful life.'

The fragment 學生活 *xué shéng huó* also has overlap ambiguity, where the middle character can either combine with the first character to form a word, or combine with the last character to form a word. The sentence containing this fragment has two plausible interpretations as shown in (2a) and (2b). Both alternations: 學生 活 *xuéshéng huó* 'student live' (2a) and 學 生活 *xué shènghuó* 'learn life' are acceptable.

Combination, Local Ambiguity

(3) 你 的 表情 十分 滑稽
nĭ de biăoqíng shífēn huájī
you STRUC look very funny
'You look very funny.'

In (3), the two characters in the fragment 十分 *shífēn* can either function as two autonomous words 十 *shí* 'ten' and 分 *fēn* 'mark', or they can combine together to function as a bisyllabic word 十分 *shífēn* 'very'. Given the sentential context of (3), however, only the second alternation is correct.

Combination, Global Ambiguity

(4)a. 我們 都 很 難 過
wŏmen dōu hĕn nán guò
we all very hard live
'We all have a hard life.'

b. 我們 都 很 難過
wŏmen dōu hĕn nánguò
we all very sad
'We all feel very sad.'

The fragment 難過 *nánguò* also has combination ambiguity. It differs from (3) in that the sentence in which it appears has two plausible interpretations. Hence, this fragment can either be segmented as 難 *nán* 'hard' and 過 *guò* 'live' in (4a), or as 難過 *nánguò* 'sad' in (4b).

Word boundary ambiguity is a very common phenomenon in written Chinese, due to the fact that a large number of words in modern Chinese are formed from free characters (Chao 1957). The problem also exists in continuous speech recognition research, where correct interpretation of word boundaries in an utterance requires linguistic and nonlinguistic information. However, people have a fascinating ability to fluidly perceive groups of characters as words in one context but break these groups apart in a different context. This human capability highlights the fact that there is a continual interaction between word identification and sentence interpretation. We are therefore motivated to study how our statistically emergent model can be used to simulate the interactions between word identification and sentence analysis. In particular, we want to study how the model (i) handles fragments with local ambiguities, such as those in sentences (1) and (3), when they appear in different sentential contexts and (ii) handles fragments with global ambiguities, such as those in sentences (2) and (4), when there is no discourse information.

3. Existing Approaches

Traditionally, word identification has been treated as a preprocessing issue, distinct from sentence analysis. We will therefore only discuss current practices in word identification, leaving sentence analysis aside. Several techniques have been used in word identification, ranging from simple pattern matching, to statistical approaches, to rule-based methods. The most popular pattern-matching method is based on the Maximum Matching heuristics, commonly known as the MM method (Liang 1983; Wang, Wang, and Bai 1991). This method scans a sentence from left to right. In each step, the longest matched substring is selected as a word by dictionary look-up. For example, in sentence (5),

(5) 計算機 的 發明 意義 重大
jìsuànjī *de* *fāmíng* *yìyì* zhòngdà
computer STRUC invention implication profound
'The invention of the computer has profound implications.'

the first three characters are identified as the word 計算機 *jìsuànjī* 'computer' because it is the longest matched substring found in a word dictionary. With the same reasoning, the words 的 de 'STRUC', 發明 *fāmíng* 'invention', 意義 *yìyì* 'implication', and 重大 *zhòngdà* 'profound' are identified.

Statistical techniques include the relaxation approach (Fan and Tsai 1988; Chang, Chen, and Chen 1991; Chiang et al. 1992), the mutual information approach (Sproat and Shih 1990; Wu and Su 1993; Lua and Gan 1994), and the Markov model (Lai et al. 1992). These approaches make use of co-occurrence frequencies of characters in a large corpus of written texts to achieve word segmentation without getting into deep syntactic and semantic analysis. For example, the relaxation approach uses the usage frequencies of words and the adjacency constraints among words to iteratively derive the most plausible assignment of characters into word classes. First, all possible words in a sentence are identified and assigned initial probabilities based on their usage frequency. These probabilities are updated iteratively by employing the consistency constraints among neighboring words. Impossible combinations are gradually filtered out, leading to the identification of the most likely combination. The mutual information approach is similar to the relaxation approach in principle. Here, mutual information is used to measure how strongly two characters are associated. The mutual information score is derived from the ratio of the co-occurrence frequency of two characters to the frequency of each character. In a sentence, the mutual information score for each pair of adjacent characters is determined. The pair having the highest score is grouped together. The sentence is split into two parts by the two characters just grouped. The same procedure is applied to each part recursively. Eventually, all word boundaries will be identified.

Both the pattern-matching and the statistical approaches are simple and easy to implement. It is well known, however, that they perform poorly when presented with ambiguous fragments that have alternate word boundaries in different sentential contexts. For instance, the fragment 十分 *shífēn*, which is a bisyllabic word in sentence (3a), functions as two separate words in sentence (6).

(6) 他 只 考 到 十 分
tā *zhǐ* *kao* *dào* *shí* fēn
he only score ASP ten mark
'He scores only ten marks.'

The MM method will regard this fragment as a bisyllabic word 十分 *shífēn* 'very' regardless of the sentential context in (3a) and (6), since this word is longer than the lengths of the two monosyllabic words 十 *shí* 'ten' and 分 *fēn* 'mark'. As a result, this method fails to correctly identify the word boundaries in sentence (6). Within statistical approaches, considering, for example, the mutual information method (Lua and Gan 1994), the same fragment is identified as a bisyllabic word in both sentences (3a) and (6)[7].

By checking the structural relationships among words in a sentence, rule-based approaches aim to overcome limitations faced by pattern-matching and statistical approaches. However, many of the rules in existing rule-based systems (Huang 1989; Yao, Zheng, and Wu 1990; Yeh and Lee 1991; He, Xu, and Sun 1991; Chen and Liu 1992) are either arbitrary and word-specific, or overly general. For example,

Rule
Given an ambiguous fragment *xyz* where *x*, *z*, *xy*, and *yz* are all possible words, if *x* can be analyzed as a so-called direction word, segment the fragment as *x yz*, else segment it as *xy z* (Liang 1990).

This syntactic rule works in sentence (7).

(7) 他 俯 下 身子
tā fǔ xià shēnzi
he bend down body
'He bends down his body.'

The fragment 下身子 *xià shēn zi* in sentence (7) is ambiguous. As 下 *xià* 'down' is a direction word, the fragment is segmented as 下 身子 *xià shēnzi* 'down body', which is as desired.

Similarly, this rule will segment the fragment 外國人 *wài guó rén* as 外 國人 *wài guórén* 'out citizen', since 外 *wài* 'out' is also a direction word. Therefore, when this fragment appears in sentence (8a),

(8)a. 他 是 外國人
tā shì wàiguórén
he COPULA foreigner
'He is a foreigner.'

the word boundaries identified will be:

b. 他 是 外 國人
tā shì wài guórén
he COPULA out citizen

which is incorrect.

Examples (7) and (8) illustrate that although syntactic information has been incorporated in word segmentation, there are still errors. In contrast, people are extremely flexible in their perception of word boundaries of ambiguous fragments appearing in different sentential contexts. We believe that the separation of word identification from the task of analysis accounts for the difference in performance. This has motivated us to study how word identification and sentence analysis can be integrated.

[7] This result is reported in Gan (1994).

4. The Statistically Emergent Model

This model is inspired by the work done in the Fluid Analogies Research Group (Hofstadter 1983; Meredith 1986; Mitchell 1990; French 1992). There are four main components in this model. Namely, (i) the **conceptual network**, which is a network of nodes and links representing some permanent linguistic concepts; (ii) the **workspace**, which is the working area in which high-level linguistic structures representing the system's current understanding of a sentence are built and modified; (iii) the **coderack**, which is a pool of structure-building agents (codelets) waiting to run; and (iv) the **computational temperature**, which is an approximate measure of the amount of disorganization in the system's understanding of a sentence.

4.1 The Conceptual Network

This is a network of nodes and links representing some permanent linguistic concepts (Figure 1).

In the network, a node represents a concept. For example, the node labeled *character* represents the concept of character; the node *word* represents the concept of word; the node *chunk* represents the concept of chunk; the nodes *character-1, character-2,* up to *character-n* represent the actual characters in a sentence; the *affix* and *affinity* nodes represent the concepts of relations between characters; the nodes *classifier, reflexive adjective, structure,* etc., represent the concepts of relations between words; the nodes *agent, patient, theme,* etc., represent the concepts of relations between chunks.

A link represents an association between two nodes. There are four types of links: (i) **category-of links**, or *is-a* links, which connect instances to types, for example, the connections from *character-1, character-2,* up to *character-n* to the *character* node; (ii) **has-instance links**, the converse of category-of links; (iii) **has-relation links**, which associate a node with the relations it contributes, for example, the connection from the *character* node to the *affix* node represents that the *character* node contributes to the character-based relation named as *affix*; (iv) **part-of links**, which represent *part-of* relations between two nodes. The direction of a *part-of* link, for instance, the link from the *character* node to the *word* node, is interpreted as 'the *character* is part of the *word*'.

During a run of the program, nodes become activated when perceived to be relevant, and decay when no longer perceived to be relevant. Nodes also spread activation to their neighbors, and thus concepts closely associated with relevant concepts also become relevant. The activation levels of nodes can be affected by processes that take place in the workspace. Several nodes in the network (e.g., *agent, patient, word, chunk,* etc.), when activated, are able to exert top-down influences on the types of activities that may occur in the workspace in subsequent processing. The context-dependent activation of nodes enables the system to dynamically decide what is relevant at a given point in time, and influences what types of actions the system engages in.

4.2 The Workspace

The workspace is meant to be the region where the system does the parsing and construction required to understand a sentence. This area can be thought of as corresponding to the locus of the creation and modification of mental representations that occurs in the mind as one tries to form a coherent understanding of a sentence. The construction process is done by a large number of processing agents.

Figure 2 shows an example of a possible state of the workspace when the system is processing sentence (9).

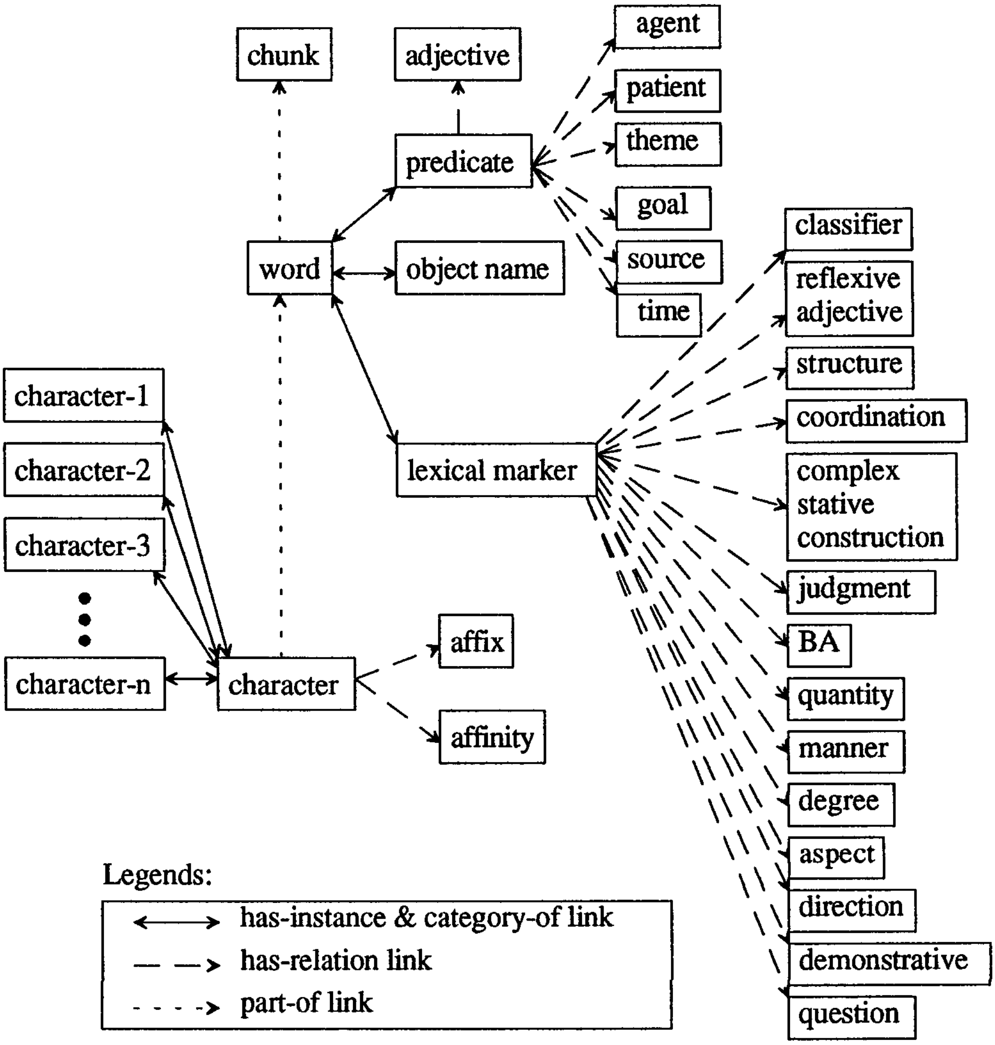

Figure 1
The conceptual network.

(9)	她	本人	生	了	三	個	孩子
	tā	*běnrén*	*shēng*	le	sān	gè	háizi
	she	self	give birth	ASP	three	CL	child

'She herself has given birth to three children.'

There are three types of objects that may exist in the workspace: **character objects**, **word objects**, and **chunk objects**. The Chinese characters in Figure 2 not enclosed by rectangles, namely, the characters 三 *sān* and 個 *gè*, are character objects. When a few Chinese characters are enclosed by a rectangle, for example 本人 *běnrén*, it indicates that these characters make up a word object. The constituent characters of the word still exist in the workspace but they become less explicit in the figure. If a group of characters is enclosed by two rectangles, for example, the character 生 *shēng*, it indicates that a chunk object exists, made up of word objects. In short, the immediate constituents of a word object are character objects, and those of a chunk object are

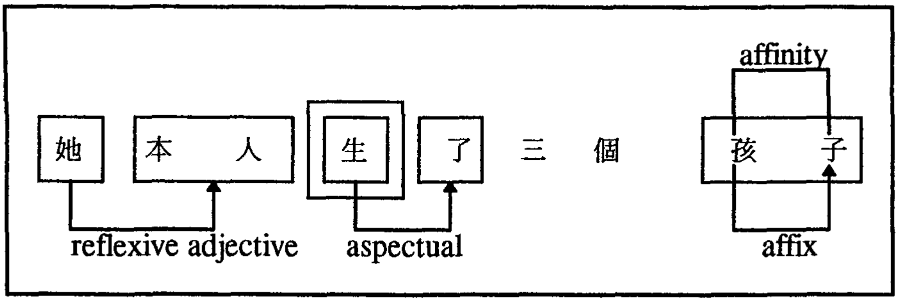

Figure 2
A possible state of the workspace.

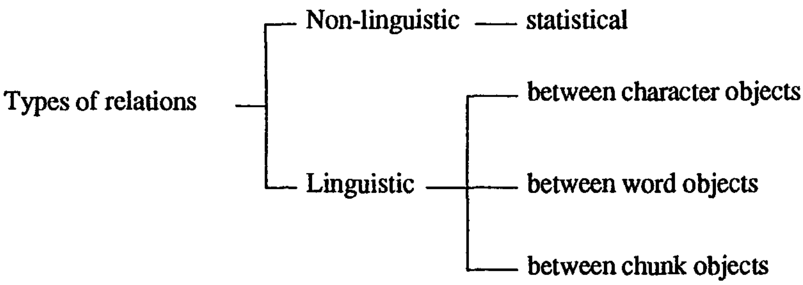

Figure 3
An overview of the types of relations.

word objects. It is possible to have unitary constituency whereby one object is the only part of another object. The chunk object 生 *shēng* 'give birth' is an example.

Each object in the workspace has a list of descriptions not shown in Figure 2. For example, descriptions of character objects include their morphological category (stem/affix) and whether they are bound or unbound.[8] Descriptions of word objects include their categorial information and sense. Descriptions of chunk objects may also include these two descriptions, except that here, these two descriptions are derived from the category and the sense of the word that is the governor.

The directed arc connecting two objects in Figure 2 denotes a linguistic relation between the objects connected. We adopt the dependency grammar notation (Tesnière 1959; Mel'čuk 1988) in which the object pointed to by an arrow is the dependent while the object where the arrow originates is the governor. The undirected arc connecting the characters 孩 *hái* and 子 *zi* in Figure 2 represents a statistical relation, and statistical relations are undirected in our representation.

An overview of our classification of relations is shown in Figure 3.

A list of all types of relations is summarized in Table 1; a detailed exposition can be found in Gan (1994).

In Figure 2, the connection between the word objects 她 *tā* 'she' and 本人 *běnrén* 'self' is a reflexive adjective relation, the connection between the word objects 生 *shēng* 'give birth' and 了 *le* 'ASP' is an aspectual relation, and the two arcs connecting the character objects 孩 *hái* and 子 *zi* are affix and affinity relations.

[8] A bound character cannot occur independently as a word.

Table 1
A list of all types of relations.

Object Type	Relation Type	Example	
		Object 1	Object 2
character	affinity relation	學	生
character	affix relation	第	一
word	classifier relation	條 'CL'	蛇 'snake'
word	reflexive adjective relation	他們 'they'	本身 'self'
word	structure relation	的 'STRUC'	父親 'father'
word	coordination relation	和 'and'	李四 'Lisi'
word	adjective relation	藍 'blue'	天 'sky'
word	complex stative relation	得 'STRUC'	好 'good'
word	attitude relation	的確 'really'	去 'go'
word	disposal relation	把 'BA'	門 'door'
word	quantity relation	我們 'we'	都 'all'
word	manner relation	會 'able'	唱 'sing'
word	degree relation	很 'very'	緊張 'nervous'
word	aspectual relation	睡 'sleep'	了 'ASP'
word	direction relation	桌子 'table'	上 'on'
word	demonstrative relation	這 'this'	魚 'fish'
word	interrogative relation	什麼 'what'	時候 'time'
chunk	agent relation	他 'he'	打破了 'broke'
chunk	patient relation	門 'door'	壞了 'broke'
chunk	theme relation	念 'chant'	經 'scripture'
chunk	source relation	從中國 'from China'	回來 'return'
chunk	goal relation	到房里 'to room'	拿 'get'
chunk	time relation	今天 'today'	不舒服 'not well'

4.3 The Coderack

The building of linguistic structures (e.g., word and chunk objects, descriptions of objects, relations between objects) is carried out by a large number of agents known as codelets. These codelets reside in a data structure called the coderack. A codelet is a piece of code that carries out some small, local task that is part of the process of building a linguistic structure. For example, one codelet may check for the possibility of building an aspectual relation between the words 生 *shēng* 'give birth' and 了 *le* 'ASP' of sentence (9). There are several codelet types. Each type is responsible for building one of the relations shown in Table 1. In addition, there are **word** and **chunk** codelet types, which are responsible for the construction of words and chunks. Two special codelet types, namely, **breaker** and **answer**, will be explained in Section 5. Here, we make a distinction between codelets and codelet type. The latter is a prewritten piece of code while the former are instances of the latter.

In the initial stage when the program is presented with a sentence, the default codelets initialized in the coderack are **affix** and **affinity** codelets. They will construct relations between character objects. Some default bottom-up word codelets are also posted to determine whether monosyllabic words could be constructed from character objects. When the *word* node in the conceptual network becomes activated by activation spreading from the *character* node, more top-down word codelets will be posted. When word objects are constructed, nodes denoting relevant relations between words will be activated. These nodes in turn cause the posting of codelets that will build relations between word objects. Again, by activation spreading to the *chunk* node, codelets building chunk objects will be posted, which will further lead to the posting of codelets that determine how chunk objects can be related.

Note that there is no top-level executive deciding the order in which codelets are executed. At any given time, one of the existing codelets is selected to execute. The selection is a stochastic one, and it is a function of the relative urgencies of all existing codelets. The **urgency** of a codelet is a number assigned at the time of its creation to represent the importance of the task that it is supposed to carry out (this is an integer between 1 to 7, with 1 as the least urgent and 7 as the most urgent). Many codelets are independent and they run in parallel. Therefore, efforts towards building different structures are interleaved, sometimes co-operating and sometimes competing. The rate at which a structure is built is a function of the urgencies of its dedicated codelets. More promising structures are explored at high speeds and others at lower speeds. Almost all codelets make one or more stochastic decisions, and the high-level behavior of the program arises from the combination of thousands of these very small choices. In other words, the system's high-level behavior arises from its low-level stochastic substrate. To summarize, the macroscopic behavior of the system is not preprogrammed; the details of how it emerges from the low-level stochastic architecture of the system are given in Sections 5.2 and 5.3.

4.4 The Computational Temperature

The computational temperature is an approximate measure of the amount of coherency in the system's interpretation of a sentence: the value at a given time is a function of the amount and quality of linguistic structures that have been built in the workspace. The computational temperature is in turn used to control the amount of randomness in the local action of codelets. If many good linguistic structures have been built, the temperature will be low, and the system will make decisions less randomly. When few good linguistic structures have been found, the temperature will be high, leading to many more random decisions and hence to more diverse paths being explored by codelets.[9]

The notion of temperature used here is similar to that in simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983). Both start with a high temperature, allowing all sorts of random steps to be taken, and slowly cool the system down by lowering the temperature. However, the decrease in temperature in our system is not necessarily monotonic. It varies according to the amount of coherency in the system's interpretation of a sentence. Thus, our system has an extra degree of flexibility, which allows uphill steps in temperature; in effect, this means that the system is annealing at the metalevel as well.

5. An Example

We will use a sample run of the program on sentence (9) to illustrate many central features of the model, including the selection of a codelet; the selection of competing alternatives; the interaction between the workspace and the conceptual network; etc. Note that this section would be overwhelmed with details if a step-by-step explanation were given. A detailed trace of the system's execution on this sentence can be found in Gan (1994), and a short description of the program's behavior can be found in Gan (1993). Here, only selected snapshots are highlighted.

Sentence (9) is an example with local, overlap, and combination ambiguities in the

[9] "Diverse paths" refers to different ways of analyzing the structure of a sentence.

Table 2
Initial state of the coderack.

Codelet Type	Urgency (U)	Temperature-regulated Urgency		Quantity
		$U_t = 100$	$U_t = 0$	
word	2	2	16	14
affinity	3	2	81	20
affix	3	2	81	8

fragment 本人生 *běn rén shēng*. Without considering the sentential context, these three characters have three possible word boundaries: 本 人 生 *běn rén shēng* 'CL human give birth', 本人 生 *běnrén shēng* 'self give birth' or 本 人生 *běn rénshēng* 'CL life'. Given the sentential context of (9), however, only the second alternative is correct.

5.1 Initial Setup

When the parsing process starts, the program is presented with the sentence. The temperature is clamped at 100 for the first 80 cycles to ensure that diverse paths are explored initially (the range of the temperature varies between 0 and 100). A **cycle** is the execution of one codelet. The number 80 is decided based on intuition and trial-and-error; it is not necessarily optimal. The workspace is initialized with nine character objects, each corresponding to a character of the sentence. Since the workspace contains only character objects, the only relevant concepts are: character, affinity, affix, and each character of the sentence. The corresponding nodes in the conceptual network, namely: *character*, *affinity*, *affix*, 她 *tā*, 本 *běn*, up to 子 *zi*, are set to full activation. Fourteen instances of word codelet are posted to the coderack. They are responsible for identifying and constructing monosyllabic words. Twenty instances of affinity codelet are also posted to identify and construct affinity relations between characters. Eight instances of affix codelet are posted to identify and construct affix relations between characters. In general, the number of codelets posted is a function of the length of a sentence.

5.2 Selection of a Codelet

Among all codelet instances that exist in the coderack, only one of them is stochastically selected to execute each time. The choice of which codelet instance to execute depends on three factors: (i) its urgency, (ii) the number of codelet instances in the coderack that are of the same type as the individual instance, and (iii) the current temperature. At cycle 0, the coderack contains the statistics as shown in Table 2.

The temperature-regulated urgency (U_t) is derived in the following way:

$$U_t = U^{(120-t)/30} \tag{1}$$

where t denotes the temperature, which ranges between $[0, 100]$. This equation is used to magnify differences in urgency values when the temperature is low. Conversely, at high temperatures, it will minimize differences in urgency values. The idea is to let the system explore diverse paths when the temperature is high, while always stick to one search path when the temperature is low.

At cycle 0 where the temperature is 100, the temperature-regulated urgencies of the three codelet types are the same. The probability of selecting an instance of a word codelet, an affinity codelet, and an affix codelet is 33.3%, 47.6%, and 19.1% respectively.

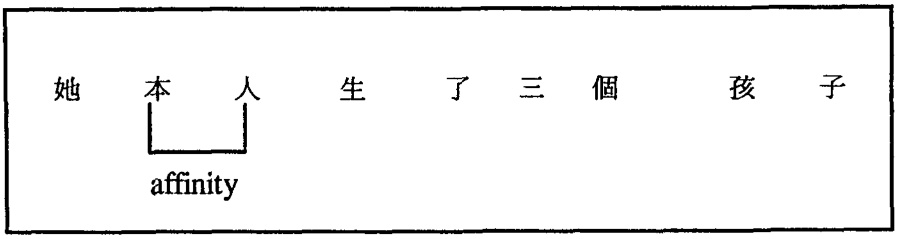

Figure 4
State of the workspace at cycle 17.

These probabilities are derived as follows:

$$P_t(C_j) = \frac{U_{j,t} \times Q_j}{\sum_{i=1}^{n}(U_{i,t} \times Q_i)} \tag{2}$$

where Q_i and Q_j are the quantities of codelet types C_i and C_j respectively, $U_{i,t}$ and $U_{j,t}$ are the urgencies of codelet types C_i and C_j at temperature t respectively, and n is the total number of codelet types.

Supposing that the coderack contains the same types of codelets with the same quantities, but the temperature is 0, the probability of selecting an instance of a word codelet, an affinity codelet, and an affix codelet becomes 8.99%, 65.01%, and 26.00% respectively. Therefore, at low temperatures, codelets with high urgency are preferred.

5.3 Construction of Linguistic Structures

Linguistic structures include high-level objects (words and chunks) and relations between two objects (see Table 1). In this run, for example, an affinity relation between the character objects 本 *běn* and 人 *rén* is constructed by an instance of an affinity codelet at cycle 17 (Figure 4).

An affinity codelet works on any two adjacent character objects to evaluate whether an affinity relation should be built between these two characters. The affinity relation is a quantitative measure that reflects how strongly two characters co-occur statistically. It is derived from mutual information (Fano 1961), which is the probability that two characters occur together versus the probability that they are independent. Mathematically, it is:

$$A(a,b) = \log_2 \frac{P(a,b)}{P(a)P(b)} \tag{3}$$

where $A(a,b)$ is the affinity relation between the character objects a and b, $P(a,b)$ is the probability that the two character objects co-occur consecutively, $P(a)$ and $P(b)$ are the probabilities that a and b occur independently. To derive affinity relations between characters, we have the usage frequencies of 6,768 Chinese characters specified in the GB2312-80 standard, and the usage frequencies of 46,520 words derived from a corpus. The total usage frequency of these words is 13,019,814. (The data was obtained from Liang Nanyuan, Beijing University of Aeronautics and Astronautics.)

Note that efforts towards building different structures are interleaved, as many codelets are independent and they run in parallel. Apart from the initial set of codelets present at the onset of processing, new codelets are sometimes created by old codelets to continue working on a task in progress, and these codelets may in turn create other codelets, and so on. The cycle in which a structure is built is not preprogrammed. Rather, it emerges from the statistics of the interaction of all codelets in the coderack.

5.4 Selection of Competing Structures

It may happen that a structure being constructed is in conflict with an existing structure. In this run, for example, an affinity relation between the characters 人 *rén* and 生 *shēng* is being considered at cycle 79. This structure is in conflict with the previously constructed affinity relation between the characters 本 *běn* and 人 *rén*. The decision about which competing structure should win is decided stochastically as a function of two factors: (i) the **strengths** of the competing structures, and (ii) the temperature. The strength of a structure is an approximate measure of how promising the structure is. It is an integer ranging between 0 and 100, inclusive. The strengths of different structures are derived according to either linguistic knowledge encoded in the lexicon or certain statistical measures. Equation (3) is a key factor in deriving the strength of an affinity relation. In this run, the strength of the proposed affinity relation between the characters 人 *rén* and 生 *shēng* is 55, while that of the existing affinity relation between the characters 本 *běn* and 人 *rén* is 56. These two values are adjusted by the temperature according to equation (4).

$$S_t = S^{(120-t)/40} \qquad (4)$$

where S_t is the temperature-regulated strength, S is the original strength, and t is the temperature. The effect of equation (4) is similar to equation (1): to maximize differences in strength values at low temperatures, and to minimize differences at high temperatures. At cycle 79, the temperature is still clamped at 100, and hence the temperature-regulated strengths of these two competing structures are both 7 (rounded up to the nearest integer). The decision about which structure should win is therefore a random one, as both have an equal probability of success. According to equation (4), at low temperatures, it is increasingly difficult for a new structure of lesser strength to win in competition against existing structures of greater strength. Since the system's behavior is more random at high temperatures, it is able to explore diverse paths in the initial stage when little structure has been built. When a large number of structures deemed to be good have been found, which entails a low temperature, the system will proceed in a more deterministic fashion, always preferring good paths to bad ones. Indeed, in this case, the new affinity relation between the characters 人 *rén* and 生 *shēng* has won. Instead of destroying the affinity relation between the characters 本 *běn* and 人 *rén*, this structure is retained, but it becomes dormant in the workspace.

5.5 The interaction between the Workspace and the Conceptual Network

Activated nodes in the conceptual network spread activation to their neighbors, and thus concepts closely related to relevant concepts also become relevant. In this run, for example, the nodes *word* and *chunk* become activated at cycle 80 due to activation spreading from the *character* node. Activated nodes influence what tasks the system will focus on subsequently through the posting of top-down codelets. For example, at cycle 80, the activated *word* node causes the proportion of *word* codelets to increase to 93%. This is an important feature of the system: the context-dependent activation of nodes, which enables the system to dynamically decide what is relevant at a given point in time, and influences what actions to take through the posting of top-down codelets.

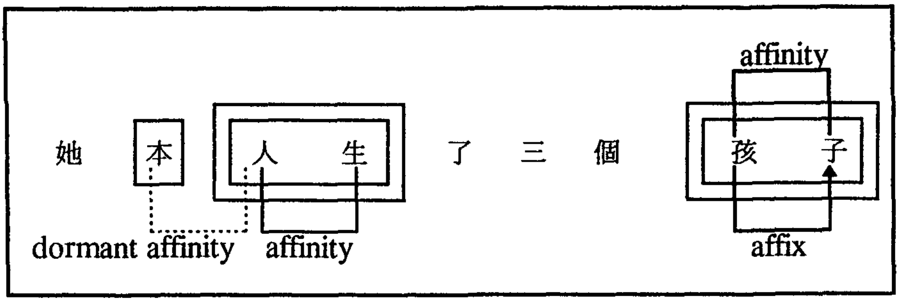

Figure 5
State of the workspace at cycle 180.

5.6 Detection and Resolution of Erroneous Structures

By the end of cycle 180, the following structures have been built (Figure 5):

- active relations: an affinity relation between the characters 人 *rén* and 生 *shēng*, 孩 *hái* and 子 *zi*, an affix relation between the characters 孩 *hái* and 子 *zi*;
- active word objects: 孩子 *háizi* 'child', 人生 *rénshēng* 'life', and 本 *běn* 'CL';
- active chunk objects: 人生 *rénshēng* 'life', and 孩子 *háizi* 'child';
- dormant relations: an affinity relation between the characters 本 *běn* and 人 *rén*.

Among them, the word 本 *běn* 'CL' is a classifier. This word has activated the *classifier* node in the conceptual network, which in turn causes the posting of *classifier* codelets to the coderack. The responsibility of this type of codelet is to explore the possibility of establishing a classifier relation between a classifier and an **object name**.[10] The use of a classifier is in general idiosyncratic. This type of idiosyncrasy is encoded in the lexicon. Since 本 *běn* cannot be the classifier of the object name 人生 *rénshēng* 'life', a special type of codelet known as a breaker codelet is posted to the coderack. The role of a breaker is to identify erroneous linguistic structures, and set them to dormant, restoring any dormant competing structure when necessary.

At cycle 187, a breaker codelet is executed that examines structures that are "in-trouble", namely, the words 本 *běn* and 人生 *rénshēng* 'life'. Since the component characters of the second word can be free, the breaker codelet concludes that this is an erroneous grouping. The word 人生 *rénshēng* 'life' is made dormant. The other structures that support the word 人生 *rénshēng* 'life', namely the affinity relation between the characters 人 *rén* and 生 *shēng* and the chunk 人生 *rénshēng* 'life', are also made dormant. The competing alternative, the affinity relation between the characters 本 *běn* and 人 *rén*, is reactivated. This snapshot also illustrates an important feature of the system: syntactic analysis can be performed without waiting for the system to complete the task of word identification.

[10] The term object name is borrowed from Meaning-Text linguistics (Mel'čuk 1988). It refers to words that cannot have a semantic dependent. A more formal attempt to define this term can be found in Polguère (to appear).

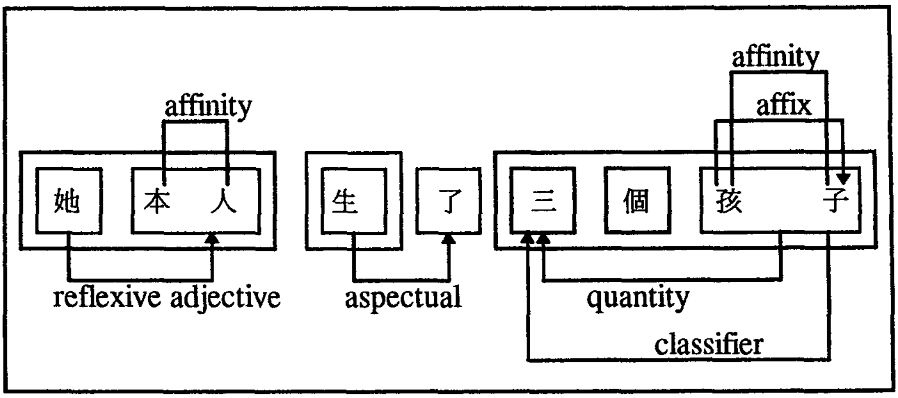

Figure 6
State of the workspace at cycle 373.

5.7 The Final State

Figure 6 shows the state of the workspace at the end of cycle 373.

For easy reference, sentence (9) is repeated here:

(9) 她 本人 生 了 三 個 孩子
tā běnrén shēng le sān gè háizi
she self give birth ASP three CL child
'She herself has given birth to three children.'

The list of structures built are:

- active relations: an affinity relation between the characters 本 *běn* and 人 *rén*, 孩 *hái* and 子 *zi*, an affix relation between the characters 孩 *hái* and 子 *zi*, a reflexive adjective relation between the words 她 *tā* 'she' and 本人 *běnrén* 'self', a classifier relation between the words 個 *gè* 'CL' and 孩子 *háizi* 'child', a quantity relation between the words 三 *sān* 'three' and 孩子 *háizi* 'child', an aspectual relation between the words 生 *shēng* 'give birth' and 了 *le* 'ASP';
- active words: 她 *tā* 'she', 本人 *běnrén* 'self', 生 *shēng* 'give birth', 了 *le* 'ASP', 三 *sān* 'three', 個 *gè* 'CL', and 孩子 *háizi* 'child';
- active chunks: 她本人 *tā běnrén* 'she herself', 生 *shēng* 'give birth', and 三個孩子 *sān gè háizi* 'three CL children';
- dormant relations: an affinity relation between the characters 人 *rén* and 生 *shēng*;
- dormant words: 人生 *rénshēng* 'life';
- dormant chunks: 人生 *rénshēng* 'life'.

Comparing the above structures with the complete analysis of the sentence in Figure 7 (for simplicity, we have omitted relations between characters in Figure 7), it is observed that the system has not yet constructed the agent and theme relations. They were not identified because the system has come to a stop at cycle 381, after an instance of **answer** codelet was executed. This type of codelet reports on the word

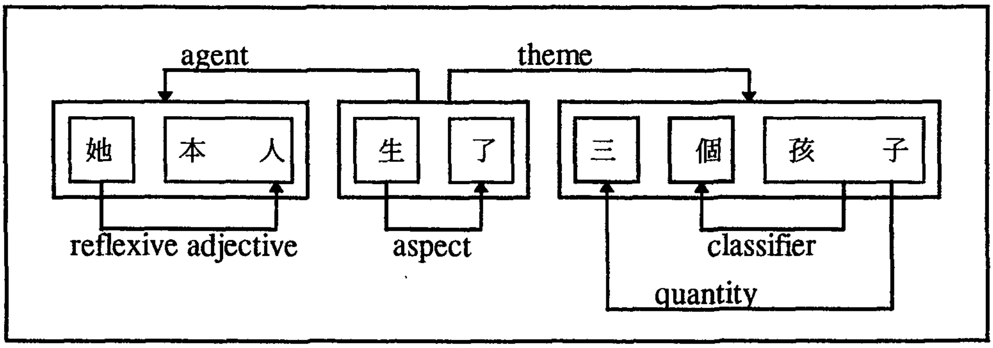

Figure 7
A complete analysis of sentence (9).

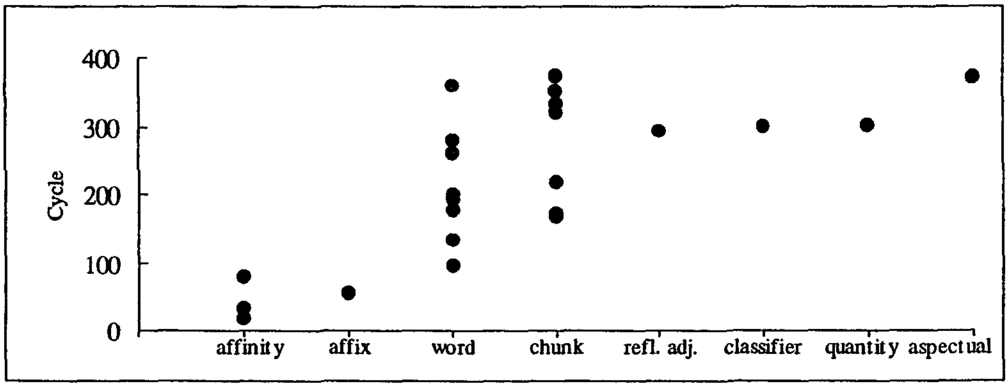

Figure 8
A graph of structures constructed against cycle number.

boundaries of a sentence. The system currently adopts a greedy approach and starts posting large numbers of this type of codelet as soon as it has identified a plausible interpretation of the word boundaries of a sentence. Hence, although instances of agent and theme codelets were present in the coderack, they were being overwhelmed by the ubiquitous answer codelets.

Figure 8 summarizes the cycle number in which various types of structures were constructed during this run. In this figure we see that affinity relations are built earlier than words, reflecting the system's preference for words of greater lengths. The system makes use of statistical information (the mutual information scores) to make quick and reliable guesses of the locations of these words. It can also be observed that overall, there is a gradual shift in the types of operations executed, from being character-centered initially, to word-centered, and then to chunk-centered. From time to time, however, the construction of different types of structures is interleaved.

6. System Performance and Discussions

Thirty ambiguous fragments that have alternating word boundaries in different sentential contexts were presented to the system and the system was able to resolve all the ambiguities. The test set covers the four types of word boundary ambiguities described in Section 2. When the sentential contexts of locally ambiguous fragments (both the overlap and combination type) were varied, our system was able to identify the correct word boundaries. When the system was presented with sentences with global ambiguities, it produced all the plausible alternative word boundaries. However, at any run of such a sentence, only one alternative is generated. The system's behavior is similar to human performance in the goblet/faces recognition problem in perception (Hoffman and Richards 1984). We cannot see both the goblet and the faces at the same time, but we are able to switch back and forth between these two interpretations. The frequencies of generating all the alternatives vary from one sentence to another. It is important to note that such frequencies are not meant to indicate some kind of "goodness" measure of alternative word boundary interpretations. Neither are they meant to reflect the preferences of a human. They are merely a reflection of the usage frequencies of Chinese characters and words in our dictionary.

The system's ability to generate different word boundaries for a globally ambiguous sentence arises from its stochastic search mechanism, which does not rule out a priori certain possibilities. This feature enables the system to occasionally discover less-obvious interpretations of word boundaries. For example, in addition to the two apparent ways of aligning the fragment 已經過 *yǐ jīng guò* as either 已經 過 *yǐjīng guò* 'already over' or 已 經過 *yǐ jīngguò* 'already go through' in sentences (10a) and (10b), a less-obvious possibility that the system has identified is: 已 經 過 *yǐ jīng guò* 'already experience over', where 過 *guò* 'over' is the complement of 經 *jīng* 'experience'.

(10)a.	我	已經	過	了	學生	時代
	wǒ	*yǐjīng*	*guò*	*le*	*xuéshēng*	*shídài*
	I	already	over	ASP	student	period

'My student days are over.'

b.	我	已	經過	了	學生	時代
	wǒ	*yǐ*	*jīngguò*	*le*	*xuéshēng*	*shídài*
	I	already	go through	ASP	student	period

'I have already gone through the period as a student.'

c.	我	已	經	過	了	學生	時代
	wǒ	*yǐ*	*jīng*	*guò*	*le*	*xuéshēng*	*shídài*
	I	already	experience	over	ASP	student	period

'I have already experienced student life.'

The system rarely produces the less-obvious interpretations. This demonstrates that its mechanisms are able to strike an effective balance between random search and deterministic search, imbuing it with both flexibility and robustness.

An issue that arises from the nondeterministic feature of the system is: will the word boundaries of a locally ambiguous sentence vary at different runs? To address this, we ran the program with each sentence 20 times. We found that for sentences covered by our current set of linguistic descriptions, the system arrived at the same word boundaries despite different paths being taken at each run. For linguistic phenomena not yet covered, suboptimal solutions may sometimes be generated. For example, when the program worked on sentence (10), it produced sentence (11) once as the answer.

(11)	中國	已	開發	和	尚	未
	zhōngguó	*yı*	*kāifā*	*hé*	*shàng*	*wèi*
	China	already	exploit	and	yet	not
	開發	的	資源	都	很	多
	kāifā	de	zīyuán	dōu	hěn	duō
	exploit	STRUC	resource	all	very	many

'China has many resources which have either been exploited or not yet been exploited.'

(12)*	中國	已	開發	和	尚	未	
	zhōngguó	*yǐ*	*kāifā*	*hé*	*shàng*	*wèi*	
	China	already	exploit	and	yet	not	
	開	發	的	資源	都	很	多
	kāi	fā	de	zīyuán	dōu	hěn3	duō
	open	distribute	STRUC	resource	all	very	many

In this run, the bisyllabic word 開發 *kāifā* 'develop' has been wrongly identified as two monosyllabic words 開 *kāi* 'open' and 發 *fā* 'distribute'. To determine the proper use of two juxtaposed predicates, such as 開 *kāi* 'open' and 發 *fā* 'distribute' in this case, requires a careful study of serial verb constructions. It is inevitable that the system would make such a mistake as our linguistic descriptions have not yet covered this phenomenon.

In comparison, consider the performance of a strictly statistical approach based on mutual information (Lua and Gan 1994): the latter wrongly identified the word boundaries in 11 out of the 30 ambiguous fragments. For the 6 fragments that appear in globally ambiguous sentences, the mutual information approach gave only one interpretation of the word boundaries. In terms of processing speed, the mutual information approach took an average of 110.4 ms to process one character; our approach took 1.7 s.[11] The extra time in our approach is spent in parsing sentences.

7. Conclusion

In this paper, we reported on a stochastically emergent model for language processing and described its application to the modeling of context effects in ambiguous Chinese word boundary interpretation. The model simulates language processing as a collective phenomenon that emerges from a myriad of microscopic and diverse activities. The proposed mechanism, whereby word objects and chunk objects are formed by the hooking up of character objects as the latter are gradually cooled down, is analagous to the crystallization process in chemistry.

Our application is distinct from existing work in two main respects:

- Word identification: We show that the full power of natural language processing can be brought to bear on the issue of word identification effectively and seamlessly. The model is able to resolve ambiguities appearing in different sentential contexts. This is an improvement over statistical approaches such as the relaxation method (Fan and Tsai 1988), which generates all possible ways of grouping the characters of a sentence into words, and then uses some scoring function to select the

[11] The mutual information approach was written in Borland C, version 2.0 while the new approach was written in Borland C++, version 3.0. Both ran on a 33 MHz, 386 machine.

best combination. At the same time, this model eliminates the use of ad hoc rules, as syntactic and semantic analysis are interleaved with word identification. This application is diametrically opposed to the reductionist approach of separating word segmentation and sentence analysis into two distinct stages. We have argued that our approach can avoid the computational problem of combinatorial explosion as the architecture has appropriate mechanisms to regulate run-time resources dynamically.

- Sentence analysis: We show that a sentence can be analyzed without assuming a presegmented input. The main feature is that there is no fixed, predetermined order of morphological, syntactic, and semantic analysis, since the control mechanism is a nondeterministic one. Essentially, the order in which these analyses are carried out is dependent on what has been discovered so far by the system, and the system's perception of what is relevant to the task it is currently investigating.

The essential idea of the proposed model is that of stochastically guided convergence to what is called a globally optimum state. This model shares some features with APRIL (Annealing Parser for Realistic Input Language) (Sampson, Haigh, and Atwell 1989). APRIL uses simulated annealing to determine the most plausible parse tree of a sentence. It begins with an arbitrary tree. Many local modifications are generated randomly. They are either adopted or rejected according to their effect on a plausibility measure. Modifications that improve the plausibility measure are always accepted; while unfavorable modifications are rejected only if the loss of merit exceeds a certain threshold. The threshold value is generated randomly but its mean value decreases according to some predefined schedule. This differs from the behavior of the computational temperature in our system, which does not have a monotonically decreasing property. Our system further differs from APRIL in the following aspects: (i) APRIL begins with an arbitrary parse tree whereas our system begins with no parse structure; (ii) APRIL's plausibility measure is defined using statistics collected from a treebank of manually parsed English text while ours is derived from mutual information statistics and linguistic constraints; (iii) parse trees in APRIL are immediate-constituency type while ours are dependency-based. That is, nodes in our system are either characters, words, or chunks. There are no nonterminal nodes defined with grammatical categories.

Our model also shares some features with connectionist models, such as fine-grained parallelism, local actions, competition, spreading activation, and statistically emergent effects from a large number of small, subcognitive events. On the other hand, the representation of concepts is quite different: they are encoded as atomic, symbolic primitives instead of distributed as weighted connections between nodes in a network, which is common in connectionist systems. Therefore, in terms of the degree to which concepts are distributed, our representation has a strong symbolic flavor; in terms of the extent to which high-level behavior emerges from lower-level processes, ours has a strong subsymbolic orientation. By providing an account of the language understanding process at such an intermediate level of description, it is hoped that our results will provide a guide to connectionists studying how such intermediate-level structures can emerge from neurons or cell-assemblies in the brain.

8. Future Work

Our application, which handles only thirty sentences at present, has enabled us to focus on the mechanisms that underlie the process of sentence comprehension, and their interactions. With the progress made in this study, which would not have been possible if we had plunged straight into large-scale unrestricted texts, our next concern would be to address the issue of scalability. There are two aspects to this issue.

- The effect of various parameter values chosen for the formulae shown in Section 5 on the operation of the program: These values are set by trial-and-error. They are not specifically tailored to our test set. To finesse these parameters in order to completely weed out unpromising search paths is impossible, since decision making in the system is stochastic. We therefore do not anticipate that the setting of the various parameter values is an issue during scaling up. The values of the parameters may affect the rate of convergence, but they will not affect the accuracy of the system in terms of the analysis results.
- The possibility of generating thousands of codelets as a result of using a large lexicon: We do not expect such a scenario to occur. Instead, having a large lexicon means that the system is able to handle more sentences. The number of codelets spawned to process a sentence is determined by the number of characters and words in the sentence, and the types of words and chunks in the sentence, not by the size of the lexicon. In addition, there are built-in mechanisms to manage the growth of codelets. We have demonstrated in Section 5 how we have made use of statistics (the maximum matching heuristics and mutual information) to avoid generating all possible word boundary combinations. The sample run in Section 5 has also demonstrated that the program need not finish executing all codelets in the coderack before it is allowed to stop, and that simpler and more clear-cut decisions tend to be made before the more subtle ones. Furthermore, certain features of the system, namely, the stochastic selection of a codelet by relative urgencies, the use of the conceptual network as a top-down controller, the interactions between the conceptual network and the workspace, enable the system to dynamically decide on the number and the types of codelets to be generated.

The real bottleneck when scaling-up is the acquisition of linguistic descriptions, as our current work has limited breadth and depth of coverage. Therefore, the current system has less practical value to people working on the word segmentation problem, where the main concern is to develop algorithms that work for large-scale text. However, the proposed model provides a useful architecture for us to study the root of what people do when they encounter unknown words in text. This issue of unknown-word resolution has been the single major problem in the segmentation of unrestricted text. Understanding how higher-level knowledge is brought to bear on this issue is essential to the design of an effective solution. Hence, our next goal is to apply the model to handle the unknown-word problem, including treatments of unknown compounds such as personal names, previously unseen place names, foreign names in transliteration, and company names.

Acknowledgments
Throughout the course of this work, we have benefited from discussions with Alain Polguère, Melanie Mitchell, Robert French, Ngai-lai Cheng, Chew Lim Tan, Loke Soo Hsu, Gee Kim Yeo, Guojin, Zhibiao Wu, and Paul Wu. We would like to express our thanks to them. We are also grateful to the reviewers for their insightful comments and suggestions.

References

Chang, Jyun-Sheng, C. D. Chen; and S. D. Chen. 1991. Chinese word segmentation through constraint satisfaction and statistical optimization (in Chinese). In *Proceedings of ROCLING-IV*, R.O.C. Computational Linguistics Conference, pages 147–165.

Chao, Yuen-Ren. 1957. Formal and semantic discrepancies between different levels of Chinese structure. *Bulletin of The Institute of History and Philosophy*, XXVIII: 1–16.

Chen, Keh-Jiann and Shing-Huan Liu. 1992. Word identification for Mandarin Chinese sentences. In *Proceedings of COLING-92*, pages 101–107.

Chiang, Tung-Hui, Jing-Shin Chang, Ming-Yu Lin, and Keh-Yih Su. 1992. Statistical models for word segmentation and unknown resolution. In *Proceedings of ROCLING V*, R.O.C. Computational Linguistics Conference, pages 121–146.

Fan, Charng-Kang and Wen-Hsiang Tsai. 1988. Automatic word identification in Chinese sentences by the relaxation technique. *Computer Processing of Chinese and Oriental Languages*, 4(1): 33–56.

Fano, Robert M. 1961. *Transmission of Information.* MIT Press, Cambridge, MA.

French, Robert M. 1992. *Tabletop: An Emergent, Stochastic Computer Model of Analogy-Making.* Ph.D. thesis, University of Michigan.

Gan, Kok-Wee. 1993. Integrating word boundary identification with sentence understanding. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 301–303. Ohio State University, June.

Gan, Kok-Wee. 1994. *Integrating Word Boundary Disambiguation with Sentence Understanding.* Ph.D. thesis, Department of Information Systems & Computer Science, National University of Singapore.

He, Ke-Kang, Hui Xu, and Bo Sun. 1991. Design principle of expert system for automatic word segmentation in written Chinese (in Chinese). *Journal of Chinese Information Processing*, 5(2): 1–14.

Hoffman, Donald D. and Whitman A. Richards. 1984. Parts of recognition. *Cognition*, 18: 65–96.

Hofstadter, Douglas R. 1983. The architecture of JUMBO. In *Proceedings of the International Machine Learning Workshop*, edited by Ryszard Michalski.

Huang, Xiang-Xi. 1989. A produce-test approach to automatic segmentation of written Chinese (in Chinese). *Journal of Chinese Information Processing*, 3(4): 42–48.

Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science*, 220: 671–680.

Lai, T. B. Y., S. C. Lun, C. F. Sun, and M. S. Sun. 1992. A tagging-based first-order Markov model approach to automatic word identification for Chinese sentences. In *Proceedings of the 1992 International Conference on Computer Processing of Chinese & Oriental Languages*, pages 17–23.

Li, Charles N. and Sandra A. Thompson. 1981. *Mandarin Chinese: A Functional Reference Grammar.* University of California Press.

Liang, Nan-Yuan. 1983. Automatic word segmentation in written Chinese and an automatic word segmentation system—CDWS (in Chinese). In *Proceedings of the National Chinese Language Processing System.*

Liang, Nan-Yuan. 1990. The knowledge of Chinese words segmentation (in Chinese). *Journal of Chinese Information Processing*, 4(2): 29–33.

Lua, Kim-Teng and Kok-Wee Gan. 1994. An application of information theory in Chinese word segmentation. *Computer Processing of Chinese & Oriental Languages*, 8(1): 115–123.

Mel'čuk, Igor A. 1988. *Dependency Syntax: Theory And Practice.* State University Press of New York.

Meredith, Marsha J. 1986. Seek-Whence: A model of pattern perception. Technical Report 214, Computer Science Department, Indiana University, Bloomington, IN.

Mitchell, Melanie. 1990. *Copycat: A Computer Model of High-Level Perception and Conceptual Slippage in Analogy-Making.* Ph.D. thesis, University of Michigan.

Polguère, Alain. To appear. Meaning-text semantic networks as a formal language. In *Current Issues In Meaning-Text Linguistics*, edited by Leo Wanner.

Sampson, Geoffrey, Robin Haigh, and Eric Atwell. 1989. Natural language analysis by stochastic optimization: A progress report on project APRIL. *Journal of Experimental and Theoretical Artificial Intelligence*, 1(4): 271–287.

Sproat, Richard and Chilin Shih. 1990. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese & Oriental Languages*, 4(4): 336–351.

Sproat, Richard, Chilin Shih, William Gale, and Nancy Chang. 1994. A stochastic finite-state word-segmentation algorithm for Chinese. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 66–73.

Sproat, Richard, Chilin Shih, William Gale, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3).

Tesnière, Lucien. 1959. *Eléments de la syntaxe structurale.* Klincksieck, Paris.

Wang, Xiaog-Long, Kai-Zhu Wang, and Xiao-Hua Bai. 1991. Separating syllables and characters into words in natural language understanding (in Chinese). *Journal of Chinese Information Processing*, 5(3): 48–58.

Wu, Ming-Wen and Keh-Yih Su. 1993. Corpus-based automatic compound extraction with mutual information and relative frequency count. In *Proceedings of R.O.C. Computational Linguistics Conference VI*, pages 207–216.

Yao, Tian-Shun Gui-Ping Zhang, and Ying-Ming Wu. 1990. A rule-based Chinese automatic segmenting system (in Chinese). *Journal of Chinese Information Processing*, 4(1): 37–43.

Yeh, Ching-Long and Hsi-Jian Lee. 1991. Rule-based word identification for Mandarin Chinese sentences—A unification approach. *Computer Processing of Chinese & Oriental Languages*, 5(2): 97–118.