LFG for Chinese: Issues of Representation and Computation

Sun Maosong
The State Key Lab. of Intelligent Technology & Systems, Tsinghua University
Beijing 100084, China
<sms@s1000e.cs.tsinghua.edu.cn>

Proceedings of the LFG01 Conference

University of Hong Kong, Hong Kong

Mirriam Butt and Tracy Holloway King (Editors)

2001

CSLI Publications

ABSTRACT

LFG has been widely used to analyze English language as well as other languages from linguistic point of view [Joan Bresnan 2001; Louisa Sadler 1996], including Chinese [Lian-Cheng Chief 1996; One-Soon Her. 1997]. A new direction in LFG research field is applying it to language computation, ranging from parsing to machine translation [Louisa Sadler, Josef van Genabith, and Andy Way 2000; Mark Johnson 2000; Miriam Butt, Stefanie Dipper, Anette Frank, and Tracy Holloway King 1999]. However, the LFG-based work in Chinese computing is rather rare [Lian-Cheng Chief, Chu-Ren Huang, Keh-Jiann Chen *et al* 1998].

The current framework of LFG shows two folds when being employed in Chinese computing tasks: it is quite powerful for linguistic representation, but seems not to be strong enough for Chinese computation – there exists some room for improving the formalism of LFG. This paper will focus on these two issues, suggesting some possible augmentations on LFG paradigm, though the idea is still preliminary. The author believes linguistic resources, such as annotated corpora, mainly semantics-oriented, are also required to make manipulations on the augmented paradigm possible. The total solution is based on not only academic research but also engineering realization – it will not work without either.

1. Introduction[*]

LFG has been widely used to analyze English as well as other languages from a linguistic point of view [Joan Bresnan 2001; Louisa Sadler 1996], including Chinese [Lian-Cheng Chief 1996; One-Soon Her. 1997]. A new direction in LFG research field is applying it to language computation, ranging from parsing to machine translation [Louisa Sadler, Josef van Genabith, and Andy Way 2000; Mark Johnson 2000; Miriam Butt, Stefanie Dipper, Anette Frank, and Tracy Holloway King 1999]. However, the LFG-based work in Chinese computing is rather rare [Lian-Cheng Chief, Chu-Ren Huang, Keh-Jiann Chen *et al* 1998].

The current framework of LFG shows two folds when being employed in Chinese computing tasks: it is quite powerful for linguistic representation, but seems not to be strong enough for Chinese computation – there exists some room for improving the formalism of LFG. This paper will center on these two issues.

2. LFG for Representing Chinese Linguistic Phenomena

LFG is powerful in describing linguistic phenomena of Chinese. Even a very sophisticated sentential construction could be successfully explained by LFG. Take sentence 1 as an example:

(1) Zhang-san fang4 gou3 yao3 si3 le Li-si.
person1 send dog bite die AUX person2
N1 V1 N2 V2 V3 AUX N3
Zhang-san sent the dog to bite Li-si, and Li-si died.

The following observations hold for this quite complex sentence: (i) N1 and N2 are SUBJECT and OBJECT of V1 respectively; (ii) V2 and V3 form a verbal phrase VP in c-structure (V3 serves as the complement of V2 syntactically); (iii) from f-structure point of view, N2 and N3 should logically be SUBJECT and OBJECT of V2, meanwhile N3 logically be SUBJECT of V3; and (iv) “yao3 si3 le Li-si” is XCOMP of V1 (see Fig 1).

[*] This research is supported by 973 Project of China under grant no G1998030507 and Foundation for the University Key Teacher by the Ministry of Education of China.

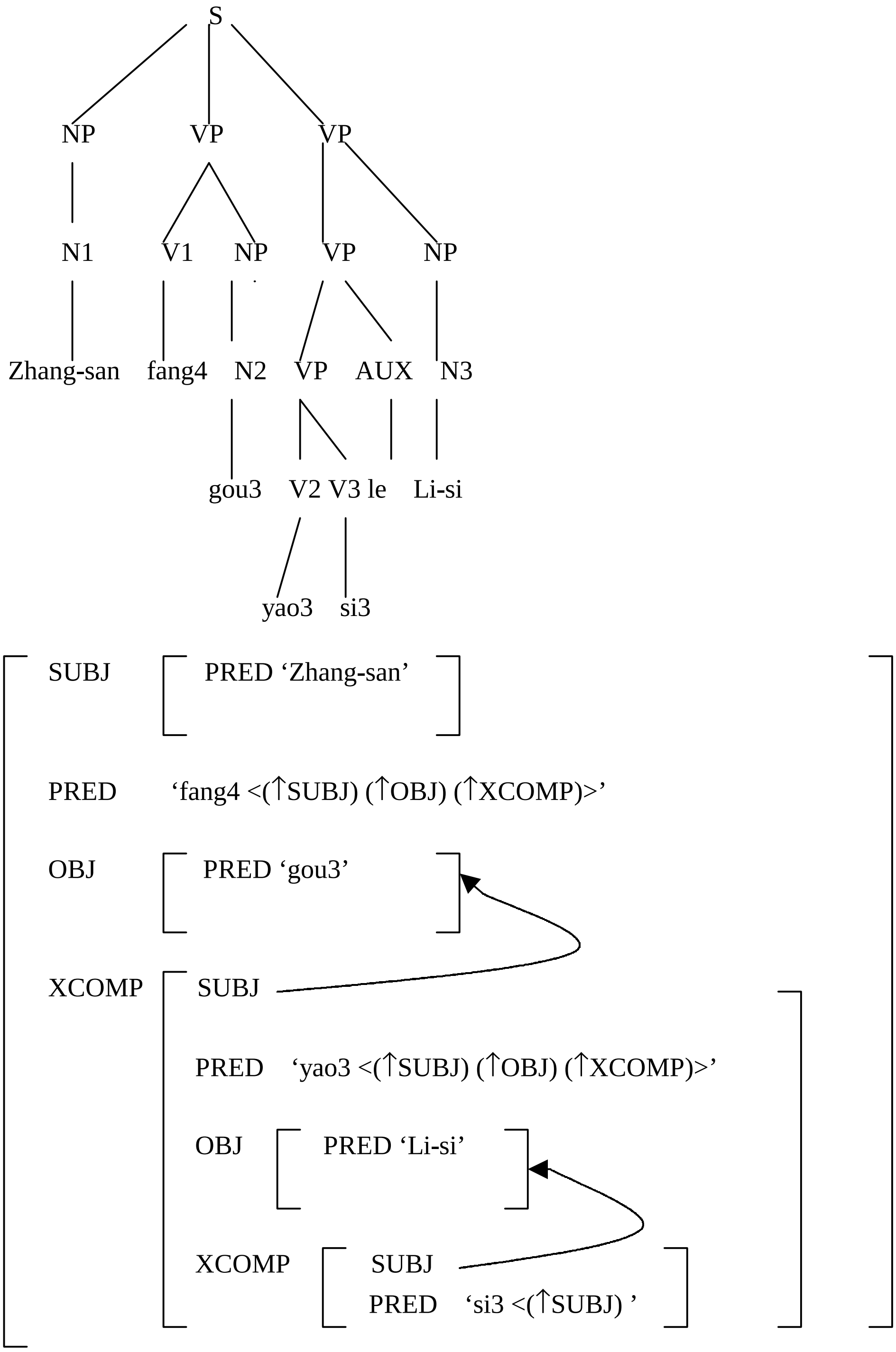


Fig.1. The c-structure and f-structure of sentence (1)

Though the mapping from c-structure to f-structure for (1) is not straightforward (note that V2 and V3 should be combined syntactically while split out semantically), it can still be built up quite easily supposing that one has already comprehended the sentence in advance. We would get similar conclusions when dealing with other types of typical linguistic constructions in Chinese.

3. LFG for Chinese Computing

3.1. The Role of Semantics in Computation

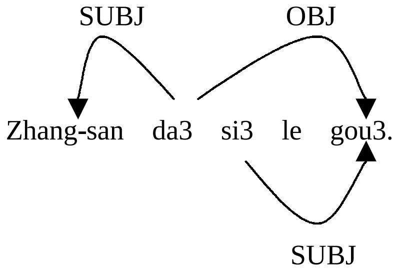
Here, I would like to show that LFG, as a computational formalism, still has some limitations for Chinese computing. The fact that Chinese is an inflection-free language (for instance, neither change in form nor explicit marker is used when a verb functions as main verb, clause, infinitive, modifier of nouns, or head of noun phrases) may result in large number of ambiguities at every linguistic level for machines. The mapping between c-structure and f-structure, as well as the mapping between f-structure and a-structure are extremely difficult to figure out, if semantic information is not provided sufficiently. Consider a group of sentences:

(2a) Zhang-san da3 si3 le gou3.
person hit die AUX dog
N1 V1 V2 AUX N2
Zhang-San hit the dog, and the dog died.

(2b) Zhang-san he1 zui4 le jiu3
person drink drunk AUX wine
N1 V1 V2 AUX N2
Zhang-san drank (the wine), and (Zhang-san) got drunk.

(2c) Zhang-san ku1 zhong3 le yan3jing1.
person cry ‘get turgid’ AUX eye
N1 V1 V2 AUX N2
Zhang-San cried, and (his) eyes got turgid.

The c-structures of these three sentences are patterned in the same way, but their f-structures are quite different, as illustrated in Fig.2: in (2a), N2 is both OBJECT of V1 and SUBJECT of XCOMP; in (2b), N2 is still OBJECT of V1, but SUBJECT of XCOMP becomes N1; in (2c), N2 serves only as SUBJECT of XCOMP, no longer OBJECT of V1.



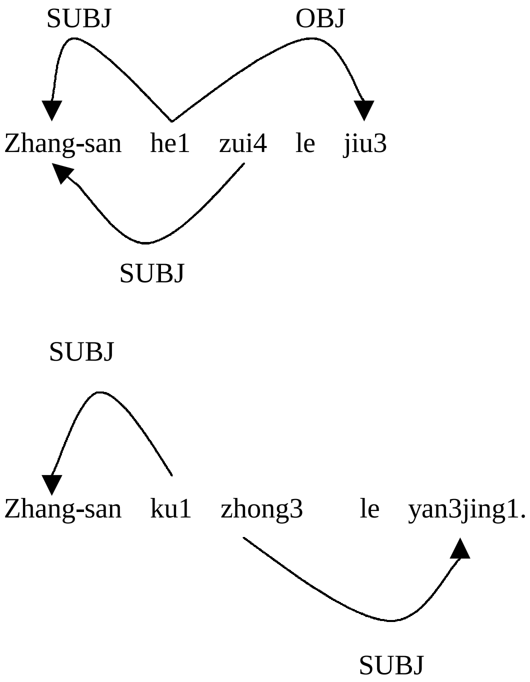

SUBJ
OBJ
Zhang-san he1 zui4 le jiu3
SUBJ
SUBJ
Zhang-san ku1 zhong3 le yan3jing1.
SUBJ

Fig.2. The f-structures (sketch) of sentences (2a) (2b) and (2c)

What makes the distinction here? The answer is obvious: it is nothing but semantic constraints among V1, V2, N1 and N2 control the one-to-many mapping processes from c-structure to f-structures.

Similar cases can be frequently encountered in Chinese. Consider another group of sentences which concerns "V1+N1+de+N2", a popular syntactically ambiguous construction in Chinese:

(3a) Yao3 lie4ren2 de gou3
bite hunter AUX dog
V1 N1 AUX N2
The dog that bites the hunter (NP)
To bite the hunter's dog (VP)

(3b) Yao3 lie4ren2 de ji1
bite hunter AUX chicken
V1 N1 AUX N2
To bite the hunter's chicken (VP)

(3c) Yao3 tu4zi3 de gou3
bite rabbit AUX dog
V1 N1 AUX N2
The dog that bites the rabbit (NP)

Both the c-structures and f-structures of these sentences differ this time. Again, semantic constraints among V1, N1 and N2 play critical role in the relevant analyses, determining which sentence out of (3a) (3b) and (3c) is realized as 'true' syntactic ambiguity and which one does not (Fig. 3):

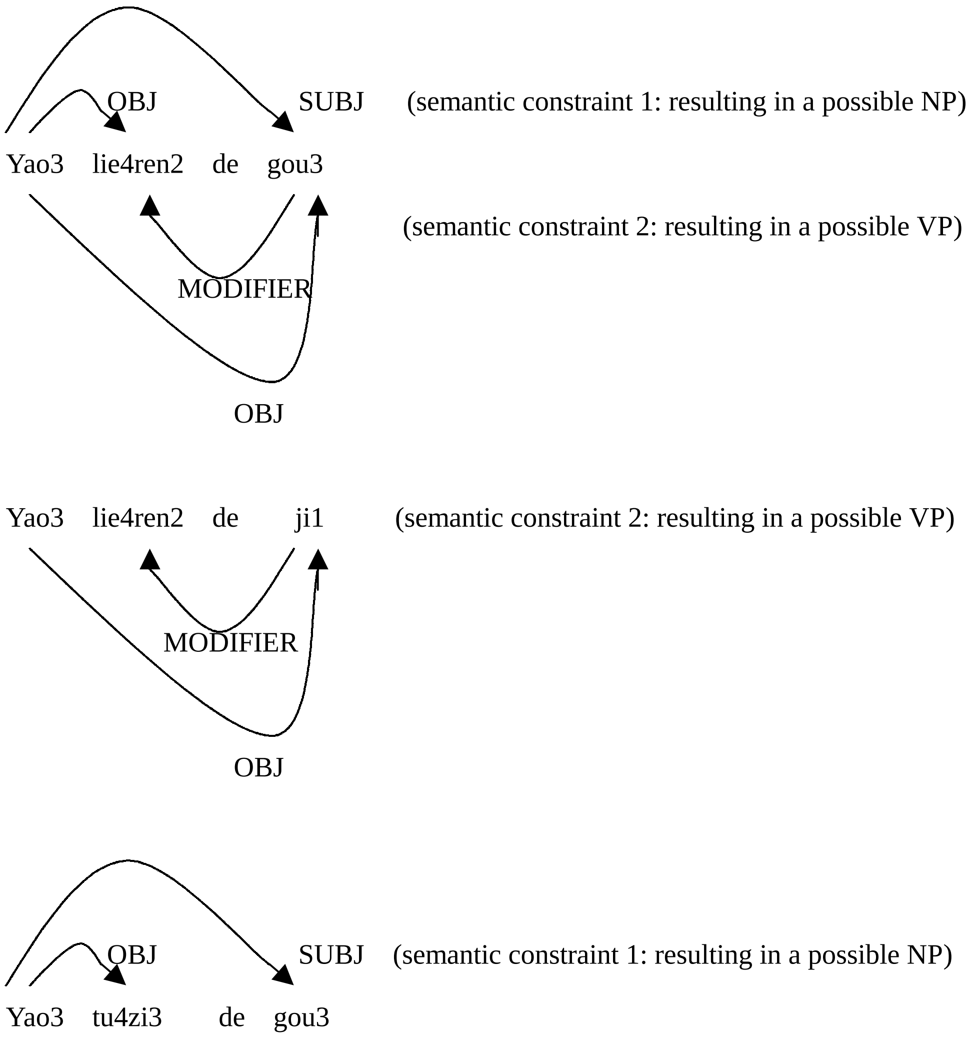

OBJ
SUBJ
(semantic constraint 1: resulting in a possible NP)
Yao3 lie4ren2 de gou3
(semantic constraint 2: resulting in a possible VP)
MODIFIER
OBJ
Yao3 lie4ren2 de ji1
(semantic constraint 2: resulting in a possible VP)
MODIFIER
OBJ
OBJ
SUBJ
(semantic constraint 1: resulting in a possible NP)
Yao3 tu4zi3 de gou3

Fig.3 Semantic constraints in potentially ambiguous construction "V1+N1+de+N2"

The situation can be even more complicated if word segmentation ambiguities are to be included. Incorrect segmentation may still lead to a syntactically well-formed but semantically ill-formed 'sentence'. For example, given the input sentence "Zhang san fang4 huo3 shao1 si3 le mei4 mei4" (note that each token here is a Chinese character, rather than word, and there is no spacing between either adjacent characters or adjacent words in original writings), the correct segmentation for it should be:

(4a)	Zhang-san	fang4	huo3	shao1	si3	le	mei4mei4.
	person	make	fire	burn	die	AUX	sister
	N1	V1	N2	V2	V3	AUX	N3

Zhang-San set up a fire, to burn (my) sister, and sister died.

But another possible segmentation exists, -- it is well-formed syntactically but almost ill-formed semantically (and, the sense of ‘fang4’ is changed from ‘make’ in (4a) to ‘put on’ in (4b)):

(4b)	Zhang-san	fang4	huo3shao1	si3	le	mei4mei4.
	person	put on	cake	die	AUX	sister
	N1	V1	N2	V2	AUX	N3

Zhang-San put on cake, and sister died.

In order to filter out (4b), a computational mechanism at semantic level is absolutely necessary.

3.2. Possible Augmentation on LFG Framework

The point addressed here is that semantic analysis is likely to be in a dominant position in computing Chinese sentences. Manipulations on a-structure, f-structure and c-structure should be carried out jointly and in parallel. To render LFG truly computable for Chinese, I believe that some augmentation is needed accordingly:

(i) Experience tells us it is easy for human to reveal those semantic constraints, but how about machines? Recall sentence (1). Suppose a machine is asked to derive f-structure from this input sentence. The pattern of the c-structure of the sentence tail, “gou3 yao3 si3 le Li-si”, is totally the same as that of the sentences (2a) (2b) and (2c). Which f-structure in Fig.2 should be assigned to this fragment (The correct one is (2a))? Of course, we need to feed all the relevant knowledge to the machine. To enable the machine to treat unrestricted texts, the knowledge ought to be given in detail and systematically, -- in more computational terms, it must take every combinatorial possibility of constraints among V1, V2, N1 and N2 into account. An accurate way of providing such knowledge is to take ‘word’ as basic factoring unit in lexicon, that is, attempting to enumerate every collocation-like ‘semantic’ correspondence between every possible word pair (In fact, the lexicon organized under current paradigm of LFG involves this sort of information implicitly).

da3: V, (↑PRED) = ‘ da3 < (↑SUBJ)(↑OBJ)(↑XCOMP)>’
(↑SUBJ PRED) = ‘Zhang-san’
(↑OBJ PRED) = ‘gou3’
(↑XCOMP SUBJ PRED) = ‘gou3’
(↑XCOMP PRED) = ‘si3< (↑SUBJ)>’

he1: V, (↑PRED) = ‘ he1 < (↑SUBJ)(↑OBJ)(↑XCOMP)>’
(↑SUBJ PRED) = ‘Zhang-san’
(↑OBJ PRED) = ‘jiu3’
(↑XCOMP SUBJ PRED) = ‘Zhang-san’
(↑XCOMP PRED) = ‘zui4< (↑SUBJ)>’

ku1: V, (↑PRED) = ‘ ku1 < (↑SUBJ)(↑XCOMP)>’
(↑SUBJ PRED) = ‘Zhang-san’
(↑XCOMP SUBJ PRED) = ‘yan3jing1’
(↑XCOMP PRED) = ‘zhong3< (↑SUBJ)>’

In the case, the number of combination can be potentially $|V|^2 * |N|^2$, where $|V|$and $|N|$ is number of verbs and nouns in the lexicon respectively. Imagine what a complex picture it would be! It is impossible to establish such a lexicon when facing the real world of Chinese language.

An alternative solution is to make some degree of approximations, shifting from ‘word’ to ‘semantic class of word’, and describing semantic constraints over semantic classes rather than word. The advantage of doing so is that the complexity of the task can be reduced dramatically – the number of word classes is at least one or two orders of magnitude smaller than that of words – so as to make the computation feasible:

da3: V, (↑PRED) = ‘#HIT <(↑SUBJ)(↑OBJ)(↑XCOMP)>’
(↑SUBJ PRED) = ?#HUMAN
(↑OBJ PRED) = ?#ANIMATE
(↑XCOMP SUBJ PRED)
(↑XCOMP PRED) = ‘?#DIE< (↑SUBJ)>’

he1: V, (↑PRED) = ‘ #DRINK <(↑SUBJ)(↑OBJ)(↑XCOMP)>’
(↑SUBJ PRED) = ?#HUMAN
(↑OBJ PRED) = ?#WINE
(↑XCOMP SUBJ PRED)
(↑XCOMP PRED) = ‘?#GET-DRUNK< (↑SUBJ)>’

ku1: V, (↑PRED) = ‘#CRY < (↑SUBJ)(↑XCOMP)>’
(↑SUBJ PRED) = ?#HUAMN
(↑XCOMP SUBJ PRED) = ?#EYE
(↑XCOMP PRED) = ‘?#GET-TURGID < (↑SUBJ)>’

Where the symbol ‘#’ denotes the succeeding entity is a semantic class, and ‘?’ means that the machine need to find a word with the semantic class specified by the succeeding ‘#’ in the input sentence.

(ii) In addition to the lexicon, a WordNet-like semantic system (including a conceptual hierarchy and a relation system among concepts, in particular among action concepts) is also indispensable, as required by the inference mechanism in computation.

(iii) To cope with a variety of ambiguities in conducting c-structure, f-structure as well as a-structure of any Chinese sentence effectively and efficiently, certain statistical mechanism should be incorporated into the LFG paradigm. For instance, we say that (4b) is ill-formed in meaning, -- this statement is relative: both “Zhang-san fang4 huo3shao1” and “Zhang-san si3 le mei4mei4” are well-formed semantically, so their combination “Zhang-san fang4 huo3shao1 si3 le mei4mei4” may still appear some degree of rationality. Though the sentence (4a) can be approved by ‘pure’ logical-form based calculations in terms of computational resources provided in (i) and (ii), the disapproval of the sentence (4b) will largely depend on the logical relation between two actions, #PUT-ON-ITEMS and #DIE. The decision could be made in terms of the probability of (4b):

$$\begin{aligned}
&\text{PROB(‘Zhang-san fang4 huo3shao1’, ‘Zhang-san si3 le mei4mei’)}\\
\approx\,&\text{PROB(‘fang4’(put on))*PROB(‘die’)*}\\
&\text{PROB(\#PUT-ON-ITEMS | ’fang4’)*PROB(\#DIE | ‘die’)*}\\
&\text{PROB(\#PUT-ON-ITEMS, \#DIE)}\\
\approx\,&\text{PROB(\#PUT-ON-ITEMS, \#DIE)}
\end{aligned}$$

All the above statistical parameters are to be derived from large scale annotated corpora.

(iv) In line with (iii), the operation ‘unification’ in LFG ought to be augmented to fit the statistical calculation. A new attribute ‘PROB’ should be added into both static lexical entries in lexicon and f-structures dynamically generated during parsing procedure. The value of ‘PROB’ of two unified feature sets is calculated from the value of ‘PROB’ of each, in principle.

4. Conclusion

Inspired by work on Chinese computing, this paper has suggested some augmentations on LFG paradigm, though the idea is very preliminary. Other resources, such as annotated corpora, mainly semantics-oriented, are also required to make manipulations on the augmented paradigm possible. It is obvious that the total solution is based on not only academic research but also engineering realization – it will not work without either.

References

Anette Frank. 2000. Automatic F-structure Annotation of Treebank Trees. *Proceedings of LFG’00.*

Joan Bresnan. 2001. *Lexical-functional syntax*. Oxford: Blackwell Publishers.

Lian-Cheng Chief. 1996. An LFG Account of Mandarin Reflexive Verbs. *Proceedings of LFG’96.*

Lian-Cheng Chief, Chu-Ren Huang, Keh-Jiann Chen, Mei-Chih Tsai, and Lili Chang. 1998. What Can Near Synonyms Tell Us. *Proceedings of LFG’98*.

Louisa Sadler. 1996. New Developments in LFG. In Keith Brown and Jim Miller, editors, *Concise Encyclopedia of Syntactic Theories*. Elsevier Science, Oxford.

Louisa Sadler, Josef van Genabith, and Andy Way, 2000. Automatic F-structure Annotation of Treebank Trees and CFGs Extracted from Treebanks. *Proceedings of LFG' 00.*

Mark Johnson, 2000. Stochastic Lexical-Functional Grammar. *Proceedings of LFG' 00.*

Miriam Butt, Stefanie Dipper, Anette Frank, and Tracy Holloway King. 1999. ParGram Project. *Proceedings of LFG' 99.*

One-Soon Her. 1997. The Lexical Mapping Theory and Mandarin Resultative Compounds. *Proceedings of LFG’97.*