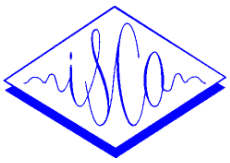
ISCA Archive
http://www.isca-speech.org/archive



6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

A TOOL TO BUILD A TREEBANK FOR CONVERSATIONAL CHINESE

Yves LEPAGE, Nicolas AUCLERC & SHIRAI Satoshi
ATR-SLT Hikaridai 2-2-2, Seika-tyō, Sōraku-gun, 619-0288 Kyōto, Japan
{lepage,auclerc,shirai}@slt.atr.co.jp

Introduction

N-grams have been extensively used with phonemes or words as basic units in speech recognition. Recently, it has been proposed to use n-grams with phrase tree structures as units to increase speech recognition quality.

In order to test this idea on Chinese, a treebank of Chinese hotel reservation conversation utterances is needed. Because no such treebank is yet available, we have to build it.

We propose to see the process of building a tree-bank as a sequence of edition and search operations:

- input or copy a new utterance (edit a text) ;
- search for similar existing utterances to get their corresponding structures and adapt them to the new utterance;
- adapt the structure (edit a tree) ;
- search for similar structures to ensure representaion and coding consistency.

This way of doing will have a benefic "snow-ball" effect: the bigger the treebank, the faster and the more consistent its extension.

1 Editing functions

Although a few visualisation/edition tools for trees exist, they are all inadequate for our purpose, either because tree edition is too cumbersome or because the layout of trees is unfamiliar to linguists. Moreover, none of them solves the problem of inputting non-latin characters.

1.1 Inputting Chinese

We faced the problem of edition of trees, and edition of texts written in a non-English language under a specialised tool, the tree editor.

The basic problem of entering and visualising non-latin character has been solved by relying on modern computer science advances in language encoding. We chose to implement our tree editor in Java, which makes the use of ISO-10646 (Unicode) transparent. With this, the problem of editing Chinese is not different from the problem of editing Arabic, Korean, etc. Entering is solved by the use of standard IME (input method editor) developed for the language in question (e.g. Wnn). Visualising is also transparent thanks to Unicode.

1.2 Editing trees

With our tool, tree edition is made as simple and direct as text edition. Interactive edition is performed directly on the canvas where the tree is drawn, without any dialogue box nor specialised menu, thanks to a rigourous parallel between node/complete subtrees on the one hand, and words/lines on the other hand.

Text	Tree
word	label of node
–	node
line	complete subtree

This parallel is valid for all functions of edition: clicking, selecting, insertion, cut, copy, paste, etc. Some equivalences are shown in Table 1.

Click	Effect	Place	
		Text	Tree
simple	position cursor in...	word	node
double	select the...	word	node
triple	select the...	line	complete subtree
Key	Effect	Place	
		Text	Tree
<space>	start a new...	word	node as right sister
<return>	start a new...	line	node as daughter
arrows	move around ...	text	tree

Table 1: Equivalences of edit functions in trees and text

Although the parallel clearly shows that a node is different from a label, people usually think that "a label is a node." To make our tool intuitive, our editor contradicts this way of thinking as little as possible.

With all this, inputting the structure

动短 (动短 (介 (从), 名 (这里), 动短 (动 (按), 名 (快门))), 副 (就), 动 (可以))

which describes the utterance

从这里按快门就可以.

is done (see Figure 2) by just typing the following sequence: 动短 <return> 动短 <return> 介 <return> 从 <↑> <space> 名 <return> 这里 <↑> <space> 动短 <return> 动 <return> 按 <↑> <space> 名 <return> 快门 <↑> <↑> <space> 副 <return> 就 <↑> <space> 动 <return> 可以

1.3 Correspondences

A special functionality of our tool is that links (correspondences in (Boitet and Zaharin 88)'s terms) between portions of the text and portions of the tree can be established. If these links are activated, selecting in the text (or the tree) simultaneously selects the corresponding part of the tree (or the text). In Figure 3, deleting the prepositional group 就可以 in the utterance, automatically deletes the corresponding part in the structure.

2 Parsing helps

The ideal way of obtaining a linguistic structure for a new utterance is to have a complete parser for the language in which the utterance is written, and to feed the utterance to this parser. Unfortunately, as parsing is still an object of research, and as precisely, we want to build a treebank to build a parser, `Boardedit` proposes parsing helps which gradually fill the gap between editing by hand and complete automatic parsing.

2.1 Matching

Retrieving similar utterances, which structures should supposedly be similar to the structure of the utterance at hand, should help the treebanker.

Exact matching allows to see whether the utterance did not simply exist already in the treebank.

Approximate matching comes in two flavours: either similar utterances which are at a certain edit distance of the utterance searched are looked for, or similar utterances which share a maximal longest subsequence with the utterance at hand are looked for. Figure 1 shows some possible settings for searching.

The first search is performed by an algorithm which has been shown to be faster than *agrep* (Wu & Manber 92). It was proposed in (Lepage 97) and became a Japanese and an American patents this year. Figure 4 gives an example of such a search. The second search is based on the algorithm de-

Figure 1: Some possible settings for the search facility.

scribed in (Hunt & Szymanski 75).

2.2 Analysis by analogy

Ideally, the treebanker would like to automatically get the structure to be built. A step in this direction is made with analysis by analogy, a technique described in (Lepage 99). New structures are built by analogy with three structures corresponding with three utterances analogical to the utterance at hand. The structure obtained in this way can then be edited by hand by the treebanker to fit the utterance at hand, using the tree editing functionalities.

Conclusion

We presented the design and our solutions for the implementation of a tool to build a treebank of utterances of conversational Chinese. It has the following features:

- On-the-spot IME for Chinese input;
- User-friendly, intuitive edition of trees;
- Search for similar utterances;
- Proposal of possible structures by analysis by analogy.

References

Christian Boitet and Zaharin Yusoff
Representation trees and string-tree correspondences
Proceedings of COLING-88, Budapest, 1988, pp 59-64.

James W. Hunt and Thomas G. Szymanski
A fast algorithm for computing longest common subsequences
Communications of the ACM, Vol. 10, No. 5, May 1977, pp. 350-353.

Yves Lepage
String Approximate Pattern-Matching
55th Meeting of the Information Processing Society of Japan, Fukuoka, August 1997, vol. 3, pp. 139-140.

Yves Lepage
Open Set Experiments with Direct Analysis by Analogy
Proceedings of NLPRS-99, Beijing, November 1999, pp 363-368.

Sun Wu & Udi Manber
Fast Text Searching Allowing Errors
Communications of the ACM, Vol. 35, No. 10, October 1992, pp. 83-91.

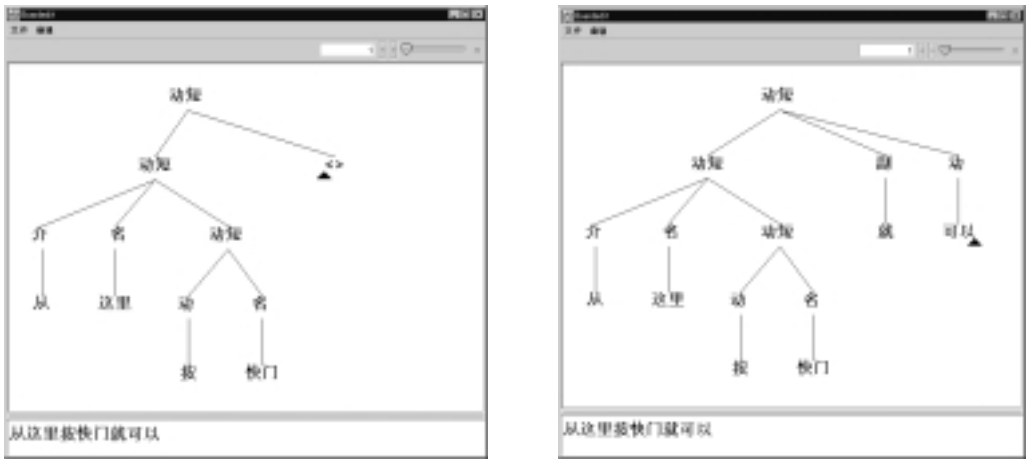


Figure 2: Completing the tree corresponding to an utterance by typing the sequence: 副 <return> 就 <↑> <space> 动 <return> 可以.

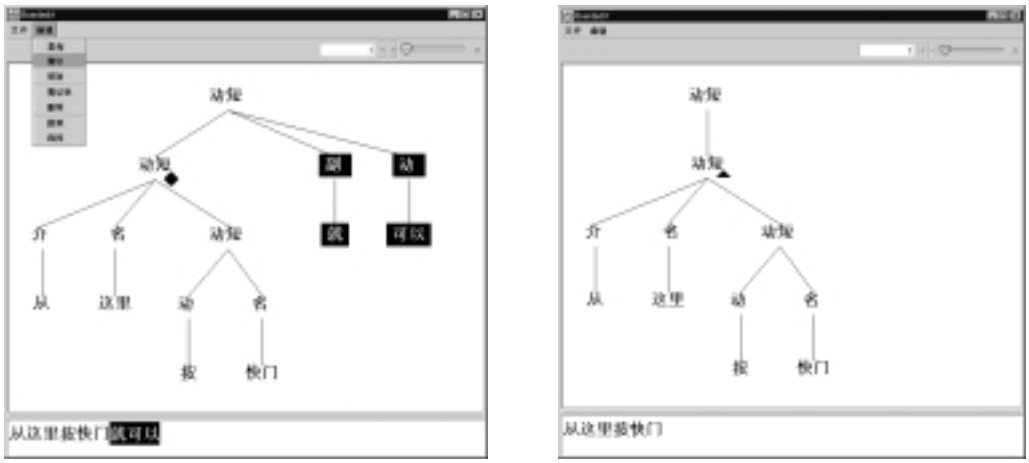


Figure 3: Simultaneous selection and deletion in the utterance and in the tree using correspondences.

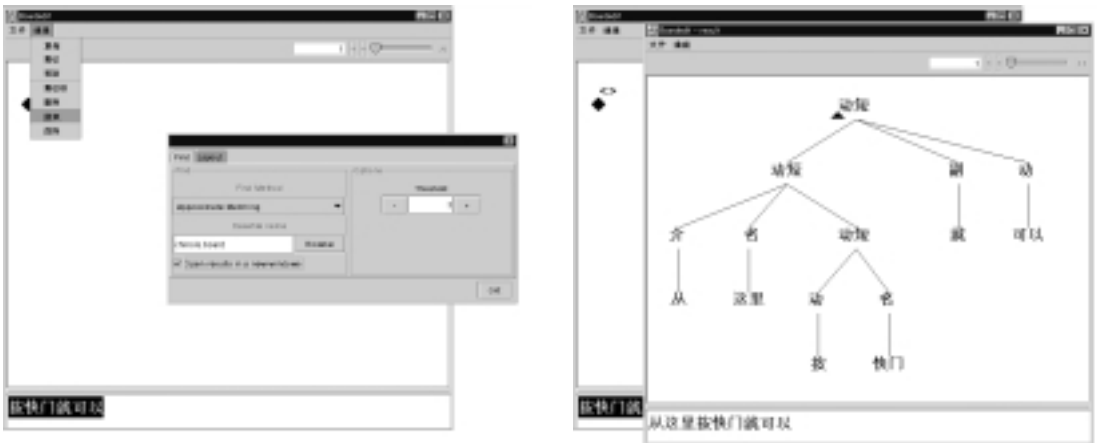


Figure 4: Search of a similar utterance with atmost 3 different characters.