1989

An investigation into the use of Prolog for Chinese-English translation

J. W. L. Millar

Follow this and additional works at: https://ro.ecu.edu.au/ecuworks

Part of the Computer Sciences Commons

This Report is posted at Research Online.
https://ro.ecu.edu.au/ecuworks/6929
Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

• Copyright owners are entitled to take legal action against persons who infringe their copyright.

• A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author’s moral rights contained in Part IX of the Copyright Act 1968 (Cth).

• Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Abstract

In order to undertake machine translation from Chinese to English, it is necessary to accomplish three tasks. Firstly, the Chinese characters need to be handled on the computer system in use - in this case an IBM PC/XT. Secondly, the grammar for the language has to be represented and here Prolog has been used with a Definite Clause Grammar. Finally, the lexicon must be stored in a manner that facilitates efficient retrieval. Arity Prolog provides a hash table that achieves this task. This report describes the current state of a project aimed at producing a Chinese-English machine translation system.
Introduction

The use of logic programming for natural language processing has been the subject of several conferences (e.g. Dahl 1988) in recent years. This application of logic programming was one of the motives behind the initial development of Prolog - the other was theorem proving (see for example Kowalski 1988, Colmerauer 1985). Currently, logic programming is the subject of considerable research in the Japanese Fifth Generation Computer System project (ICOT 1988). In recent years there has also been considerable interest in the handling of characters in character sets and scripts other than Roman characters as used by English (Becker, 1987, Archer et al. 1988). This is now also the subject of an international conference and considerable publication (Wang 1988). The time is right, therefore, for an investigation into one point at which these research activities meet, namely, the use of Prolog for Chinese to English translation.

Within the context of the Western Australian College of Advanced Education (W.A.C.A.E.), this work is useful in that it provides a framework for continuing research in a field which is of some interest to the Mt Lawley campus. This campus is the major campus for the Department of Computer Studies, which offers postgraduate level studies in artificial intelligence including natural language processing. The campus is also the home of the Department of Language Studies, which offers graduate studies in interpreting and translating and is proposing a degree programme in Chinese. It is hoped that the work described in this report will open up mutually beneficial research directions for the two departments.

Previous work in the area has been reported by Huang (1986). The grammar reported there was very simple and deliberately bidirectional. The bidirectionality arises naturally from the use of Prolog clauses that can perform an operation and its inverse, dependent upon which of the arguments are ground. The intention here is to expand the grammar past that simple level, with the inevitable outcome that the result will be unidirectional. Furthermore, whilst the target language is English, it is not intended at this stage to generate English sentences; rather the result will be a logical form of the type used by Allen (1988, p193).

Subsequent sections address the following issues. Firstly, this report reviews the manner in which the written Chinese
language is manipulated in the computer system and the implications that this has for the operation of the overall translation process. Secondly, the features of Prolog that make it suitable for this task will be considered. There are considerations deriving from both the general task of natural language processing and from the nature of the Chinese language itself. Thirdly, there is some discussion of the storage issues that arise. In particular the lexicon for a moderately complex machine translation system must necessarily be voluminous. Finally, some future directions for this work are outlined.

**Chinese Characters on a Personal Computer**

In order to build a Chinese to English translation system, it is necessary to be able to display both Chinese and English characters. There has been considerable work done both within the Peoples’ Republic of China (PRC) and Taiwan (see for example Archer 1988) on displaying and manipulating Chinese characters on a computer screen. The particular product used in this work has been CCDOS running on an IBM PC/XT compatible. This enhancement to IBM’s DOS operating system is widely available in Hong Kong.

A brief mention of some of the features of Chinese characters is appropriate at this point. Unlike the Roman alphabet, the character set used in Chinese is vast. Fortunately, common usage only requires about 5000 characters out of the 50,000 or more characters recorded. Ann (1987, p78) notes that 98.8% of the characters in common use in daily Chinese language newspapers in Hong Kong are contained in a set of 2500 characters. Based on statistics such as this, packages such as CCDOS seek to provide a set of most frequently used characters in the most efficient manner possible.

A useful characteristic of many computers is the use of the ASCII character set, which prescribes a seven bit representation of characters, within eight bit bytes. The common feature of such systems is to use the high bit to select a range of up to 128 graphics characters. CCDOS and other oriental character systems (e.g. IBM’s Double Byte Character Set - DBCS - (IBM 1986)) use a double byte to encode any particular character. CCDOS uses the high bit set to indicate that this byte and its neighbour in the data stream, also with its high bit set, should be treated as representing one character from the Chinese character set. A byte without the high bit set is treated as an ASCII character.
To input a Chinese character for use by an application program, CCDOS provides a variety of methods. One method that is commonly used is based on the phonetic pinyin Romanization of the Chinese language. Pinyin is based on the official spoken form of the language known as putonghua, which is essentially a Romanization of the Beijing accent. The CCDOS input based on this phonetic information is further complicated by the need to encode the syllable into its three (at most) components e.g. ‘zh’ as an initial sound is encoded to ‘a’. The pinyin Romanization is now commonly taught in PRC schools, however the older generations are not familiar with it. The further encoding to three input keystrokes per character complicates the input significantly for the Chinese speaker. For a computer-literate Chinese-illiterate however, this method is tolerable. An alternative input method in terms of a four digit index is even less amenable to fluency of input.

Utilizing the favoured CCDOS input method, one can generate a menu of homophones across the bottom of the screen. Selection of the preferred character then passes the appropriate double byte to the Chinese character generator within CCDOS to be displayed as a Chinese character. This pair of bytes can also be read and stored in a file by using either an appropriately modified version of software such as dBase3(TM) or Wordstar(TM), or a program written in a conventional programming language, such as Turbo Pascal(TM), which can operate in the CCDOS environment. In this fashion one can create the necessary files for use by the remainder of the suite of software.

Firstly, a dictionary of Chinese-English translations is created and stored under dBase3. This provides English translations of groups of between one and five Chinese characters and is discussed further below. The database environment provides useful sorting and searching facilities during data entry and the inevitable modifications during development. The current dictionary is the 270 entry dictionary contained within the Beijing Language’s Institute (1980) first reader on written Chinese. The initial input of new data is to a file under the control of a Turbo Pascal program running under CCDOS. Under regular DOS, this file is appended, as SDF (standard data format), onto the dictionary under development in dBase3. The field with Chinese characters displays as graphics characters under these circumstances unless a modified version of dBase3 is available. The first 20 entries in the dictionary are appended as Appendix A. Since Chinese is a tonal language, each pinyin syllable needs to be
marked to indicate which of the five tones is appropriate. The tones are generally referred to as the first to fourth tones and a neutral tone. The first to fourth tones are marked - / \ respectively after the pinyin syllable and the neutral tone is unmarked. These markings are an attempt to represent the nature of the tone e.g. the second tone is a rising tone and marked in Appendix A as / \.

Using a modified Wordstar, the input for the translation system is created as a Document file. This creation phase occurs under CCDOS; however the subsequent translation in done under Arity Prolog(TM) running under DOS. The use of the two operating systems is necessitated by the fact that some of the software used is not designed to run in the double byte version of the operating system. The rebooting between the two operating systems adds considerably to development time.

The Use of Prolog for the Translation

Once an input method has been provided to allow entry of the Chinese data to be translated, the next step is to translate it. In essence this requires the matching of input Chinese characters (double bytes) to entries in a Chinese-English lexicon. The lexicon is organised to allow retrieval by semantic category (noun etc.). The lexicon also contains extra information necessary to determine the structure and meaning of the input text. Information from the matching process can be used in association with a set of rules for the Chinese language. This set of rules, the grammar, may relate just to the syntactic or structural aspects of a sentence, or may also incorporate semantic information.

The lexicon originates from the dictionary file stored under dBase3. It is read into Arity Prolog and stored as a hash table of records. The fields in these records contain
1. the Chinese characters,
2. the pinyin,
3. the English translation and
4. a semantic category (noun etc.).

The first two bytes of field 1. above (i.e. the first Chinese character) are used as the key for the hash table. Further development may require further fields. For instance, the Chinese language has a class of words that are referred to as classifiers. Each noun has a particular classifier that is used with respect to it in a phrase such as:

\[ yi \ ge \ ren = a \ <\text{classifier}> \ person \]
This classifier is currently only checked to see that it occurs in the lexicon as a classifier. Further development of the grammar could incorporate a check that ensures the use of the correct classifier for the following noun. This would require a further field, at least for nouns, in the lexicon. Examples of classifiers can be seen in Appendix A where there are two entries of type measure with no associated English translation.

The grammar currently in use is entirely syntactic in nature. The grammar is based on the sections on Chinese grammar included in each chapter in Beijing Language Institute (1980). Currently, the grammar contains 29 rules. The marginal cost of adding further rules has been around 2 hours work per rule, including testing. Whilst the author has gained a small degree in proficiency in debugging such grammars, this has been counteracted by a rising number of interactions between the rules that need to be checked. A copy of the current grammar is appended (Appendix B). A process of validating the grammar with members of the staff of the Guangzhou Institute of Foreign Languages has been initiated. With the grammar based on the same source as the lexicon, the resultant system should be able to translate any of the text from that book. This then defines the world of test data.

One of the major reasons for using Prolog is the ease with which a grammar for natural language processing can be created. One can either write the rules for the grammar directly in Prolog or adopt a more indirect approach. One successful method has been Definite Clause Grammars (DCG), developed by Pereira and Warren (1980). The advantage of this method, exemplified below, is that the pair of list arguments needed to transmit the list being parsed into and out of each predicate are not needed. For example, in Prolog one might write:

\[
\text{sentence(In, Out) :-}
\text{noun_phrase(In, Partial), verb_phrase(Partial, Out).}
\]

The pairs of arguments are referred to as difference pairs. In contrast, the DCG formalism allows:

\[
\text{sentence --> noun_phrase, verb_phrase.}
\]

When the file containing the DCG clauses is consulted, they are automatically translated into Prolog and the arguments added. The simpler DCG clause is obscured, however, by the addition of arguments that return the logical form containing...
the result of parsing part of the sentence:

\[
\text{sentence}([\text{sentence}, N, V]) \rightarrow \\
\text{noun\_phrase}(N), \text{verb\_phrase}(V).
\]

By incorporating the returned ground terms N and V in the list structure [sentence, N, V], this partial logical form can now be unified with a single logic variable e.g.

\[
\text{parse} \rightarrow \text{sentence}(S), \text{output}(S).
\]

The difference pairs are added to the DCG formalism when the file of DCG clauses is consulted. Their function is to move data through the predicates that form the grammar. They define a section of the sentence in terms of the difference between the two lists. After a predicate has been proved, the second argument will be instantiated to the unparsed portion of the first. An example will clarify this:

\[
\begin{align*}
\text{if parsing} & \\
\text{the big dog likes me} & \\
\text{noun\_phrase should end up with arguments} & \\
\text{noun\_phrase}([\text{the, big, dog, likes, me}], [\text{likes, me}])
\end{align*}
\]

The useful nature of this process is that the parsing will identify and remove the contiguous sublist at the head of the input so that it can be dealt with. At the bottom of the parse tree, there must be primitive predicates that actually access the lexicon. These establish whether or not an entry exists in the lexicon that identifies the contiguous sublist, of one or more characters, as the part of speech currently required. This process progressively consumes the input sentence, building up a logical form to be returned through the arguments of the DCG clauses. At any point, where a goal cannot be satisfied for an expected part of speech, the Prolog backtracking will unwind the instantiations until another possible sentence form is found.

It is now appropriate to note some of the features of the written Chinese language. Firstly, a complete sentence will contain a string of Chinese characters, each of which is roughly similar to an English syllable. Each of these syllables may either have a role in its own right in the context of the remainder of the sentence, or it may be part of a multisyllabic combination of characters. Bi-syllabicism is a common feature of Chinese, where two characters form what might in English be recognised as a word (Ann, 1987, dwells considerably upon this aspect of modern Chinese). The
confusing aspect for the beginning Chinese language student is that, occasionally, three or more characters form an association in this way. Furthermore, there is no use of spaces to delimit these groupings.

It is thus necessary to provide a mechanism to access groupings of characters of arbitrary length stored in the lexicon. To do this, the lexicon and its source dictionary have been provided with a field to hold the Chinese characters with a 5 character (10 byte) field width. This is long enough to store some common expressions which might otherwise be difficult to parse. In order to store these groupings in the hash table referred to above, it is necessary to provide a key to be hashed. Since any grouping of up to five characters will always have at least one character, and since Prolog's list handling syntax provides easy access to the leading or head elements of a list, the first character (2 bytes) is used as the key to the hash table. Prolog has to therefore backtrack its way serially through all groupings in the hash table with this same lead character until a grouping that matches the leading characters on the yet to be parsed string is found. If the search fails, Prolog backtracks to a previous choice-point. This convenient and flexible manner of dealing with arbitrary length parts of speech is a desirable consequence of using Prolog.

There are some features of the Chinese language that make it easier to handle in this fashion than the Romance languages, such as English. The most notable is the lack of inflections added to words as the tense or number changes. This means that there is no need to undertake the morphological analysis necessary in those languages that make use of inflections. In part this explains the lack of normal English as the output of this system. That requires a study in its own right. There is no analogue to checking to see that number is consistently represented in inflections on widely separated parts of speech (this requires another argument in the DCG clauses for parsing English) and consequently information like number must now be inferred from contextual information in the sentence.

The questions of completeness and soundness deserve mention. The intention is to build a complete grammar for the subset of the total Chinese grammar expressed within the boundary of Beijing Languages Institute (1980). However, the grammar is unsound in that some ungrammatical constructions will be successfully parsed. In order to produce soundness there needs to be further refinement towards the leaves of the parse tree. This necessarily produces a deeper parse tree with more
backtracking and a deleterious impact on performance. This aspect of the grammar needs further consideration.

The operation of the system is demonstrated in the final appendices. Appendix C contains a sample set of Chinese sentences. The output from the program is contained in Appendix D.

**Storage Issues**

The current lexicon consists of Prolog terms that have been read directly from the dBase3 file and stored in Arity Prolog's database. The Arity Prolog implementation of Prolog gives the user several choices of storage technique. The data could be stored

a. linearly with the remainder of the program;

b. in a B-tree in a separate database world (Arity Prolog's terminology); or

c. in a hash table in the database world.

Retrieval from clauses is the slowest of all and the choice lies between the B-tree and the hash-table. Timings conducted with simple sentences show that the use of the hash table has significant benefits in terms of total time to parse, and consequently a hash table is currently being used. The processes of reconfiguring the lexicon into an alternate form and then modifying the means of accessing the lexicon are both straightforward.

The clauses that form the grammar and the hash table holding the lexicon are held in memory in 16kB pages. Currently twelve pages are required to hold the lexicon and grammar and it is possible to have these all memory resident by appropriate manipulation of the space reserved for stacks and pages. The nett result is that this package can run without paging. This obviously enhances the performance of the logic program. However, it is clear that augmentation of the grammar or the lexicon will require virtual memory management.

The greatest expansion will be to the lexicon. Currently this contains 260 characters or combinations of characters. Expanding this by a factor of 10 might provide an adequate coverage of a small domain. This will inevitably lead to considerable use of virtual memory, with most of the lexicon being held in non-resident pages. The consequent swapping will reduce performance considerably. It remains to be seen the extent to which the hash table method retains its performance advantage over B-trees in these circumstances.
Expansion of the grammar will increase the storage requirements, but not in the dramatic manner required for a sizeable lexicon. The implications are that there will be a deeper or broader tree to parse, with consequent impact on time performance. Any deeper structuring of the possible parse tree, caused by a successive refinement of semantic categories for instance, will require larger provision for stack space with an immediate impact on the number of lexicon pages that can be held resident.

**Future developments**

Developments appear to be possible and even desirable in three areas. These are

1. the means of entering Chinese characters into the computer,
2. the grammar and
3. expansion of the lexicon.

The first of these can probably be achieved by acquisition of software and hardware that specifically support the necessary functionality. It may well be that some of this work within W.A.C.A.E. can be achieved as a result of projects at undergraduate levels. Such projects could provide easier means of accessing and using bit-mapped Chinese characters, and provide the software that work with them, e.g. database software. Hardware is available (e.g. the Weiwu Hanzi card) that improves the process of handling Chinese characters, presumably by not consuming vast areas of RAM for the bit-mapped character library (in CCDOS this is around 270 kb). Software is being developed for several of the numerous methods (over 600 by one newspaper report) by which Chinese characters are being encoded. Development of a Prolog interpreter that operates in such a Chinese character oriented environment would appear to be more involved. A version of Prolog that can recognise Chinese characters is referred to in (Hsu, 1988, p18).

Improving the grammar is necessary in order to increase the range of sentence constructions that can be handled. In addition there are issues such as semantic checking and discourse analysis that need attention. One means of achieving the first of these is to move into the grammar of Book 2 of the series following Beijing Foreign Languages (1980). This corresponds with the author’s personal progress in learning Chinese, and will be done primarily as an aid to learning. However, it remains an open question as to whether this is a good direction to follow in order to build the grammar.
efficiently. Enhancements to performance available by the use of common techniques, such as the use of a chart parser (Earley 1970), are still to be employed. Semantic checking should allow resolution of questions of number and tense which can be used for English text generation. Discourse analysis on the other hand entails constructing meaning from more than one sentence in order to resolve pronominal references, anaphora and elliptical references, and then answering questions. Both of these are current research areas and could lend themselves to substantial and valuable research tasks.

Expanding the lexicon will entail both increasing the amount of data held for each entry and augmenting the total number of entries. There is, perhaps, little point in a large scale augmentation of the number of entries until the structure necessary to support the extra semantic checking of an improved grammar has been developed.

Conclusions

The use of Prolog for Chinese - English translation appears to be viable. The following features of Prolog prove to be useful:

1. Definite Clause Grammars allow convenient representation of the grammar.
2. The difference pairs, in the logic program that results from the DCG, cope with continuous streams of Chinese characters that need to be broken into groupings.
3. The particular implementation of Prolog used provides useful fast access database facilities for the lexicon.
4. In a batch mode of operation, Prolog can deal with the graphics characters that correspond to Chinese characters in another environment.

Further development of the lexicon will see performance degrade, due to the use of virtual memory and an increase of paging. Further development of the grammar will see further degradation in performance due to a greater number of rules to be matched or deeper parse trees or both.
Acknowledgements

It is necessary to acknowledge several colleagues who have kindled the author's interest in the Chinese language and the techniques necessary to deal with it on a computer. This includes colleagues Wolfgang Frick and Gek Ching Thai from W.A.C.A.E. and Yang Shi-hong, Fu Wen-yan and Ma Jian-huai whilst on exchange at W.A.C.A.E. from the Guangzhou Institute of Foreign Languages. Their patience is appreciated. Detailed reading of the draft and suggestions from Geoff Sutcliffe were also appreciated.
References


Beijing Languages Institute, (1980), *Elementary Chinese Reader Book 1*, Foreign Languages Press, Beijing, PRC.


Pereira, F.C.N. & Warren D.H.D, 1980, *Definite clause grammars for language analysis - a survey of the formalism and a comparison with augmented transition networks*, *Artificial Intelligence Report No. 89/1 - Page 13*
<table>
<thead>
<tr>
<th>Pinyin</th>
<th>English</th>
<th>Chinese Type Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>a-li</td>
<td>Ali</td>
<td>proper 阿里</td>
</tr>
<tr>
<td>an-jing</td>
<td>quiet, calm</td>
<td>adj 安静</td>
</tr>
<tr>
<td>an-na</td>
<td>Anna</td>
<td>proper 安娜</td>
</tr>
<tr>
<td>ba</td>
<td>suggestion</td>
<td>endsent 巴</td>
</tr>
<tr>
<td>ba-ba</td>
<td>8</td>
<td>number 八</td>
</tr>
<tr>
<td>ba/</td>
<td>father</td>
<td>noun 爸爸</td>
</tr>
<tr>
<td>bai</td>
<td>white</td>
<td>adj 白</td>
</tr>
<tr>
<td>hai</td>
<td>100</td>
<td>number 百</td>
</tr>
<tr>
<td>ban</td>
<td>class, squad</td>
<td>noun 班</td>
</tr>
<tr>
<td>ban/</td>
<td>half</td>
<td>number 把</td>
</tr>
<tr>
<td>bao</td>
<td>newspaper</td>
<td>noun 报</td>
</tr>
<tr>
<td>ba&quot;</td>
<td></td>
<td>measure 把</td>
</tr>
<tr>
<td>bei&quot;hai&quot;</td>
<td>Beihai</td>
<td>proper 北海</td>
</tr>
<tr>
<td>bei&quot;jing&quot;</td>
<td>Beijing</td>
<td>proper 北京</td>
</tr>
<tr>
<td>ben&quot;</td>
<td></td>
<td>measure 本</td>
</tr>
<tr>
<td>ben&quot;zi&quot;</td>
<td>note-book</td>
<td>noun 本子</td>
</tr>
<tr>
<td>bie/de</td>
<td>other</td>
<td>adj 别的</td>
</tr>
<tr>
<td>bu</td>
<td>not</td>
<td>modifier 不</td>
</tr>
<tr>
<td>cai&quot;se&quot;</td>
<td>multicolour</td>
<td>adj 彩色</td>
</tr>
<tr>
<td>cao-chang&quot;</td>
<td>sportsfield,</td>
<td>noun 操场</td>
</tr>
</tbody>
</table>

Appendix A. The first twenty entries of the Chinese-English dictionary. (Note - in the Pinyin, tones are marked as: 1= -, 2= /, 3= ~, 4= \)
parse([assertion | S]) --> sentence(S), fullstop.
parse([question, I | S]) --> sentence(S), interrogative(I), fullstop.

sentence([subject | S], [P]) --> subject(S), predicate(P).
sentence([P]) --> predicate(P).

subject(N) --> noun_phrase(N).
noun_phrase(N) --> noun_phrase1(N), noun_phrase2(N).
noun_phrase1([N]) --> name(N).
noun_phrase1([P]) --> pronoun(P).
noun_phrase1([N]) --> noun(N).
noun_phrase2([N, A]) --> attributive_clause(A), noun(N).
noun_phrase2([A, N]) --> qualifying_list(A), noun(N).

predicate([main_verb, Vt, V, [direct_object | N|M]]) --> verb_phrase([Vt, V|M]), noun_phrase(N).
predicate([main_verb, Vt, V, [no direct object] | M]) --> verb_phrase([Vt, V|M]).
predicate([is, has, N]) --> noun_phrase(N).
predicate([is, A]) --> adjective_list(A).
predicate([is, a, A, one]) --> verb(Be), (Be == be), adjective(A), de.

verb_phrase([verb, V | A]) --> adverb_list(A), verb(V).
adverb_list([]) --> [].
adverb_list([A | L]) --> adverb_phrase(A), adverb_list(L).
adverb_phrase([adverb, A]) --> adverb(A).
adverb_phrase([at_loc, P | N]) --> preposition(P), noun_phrase(N).

attributive_clause([who, V, [object | N]]) --> verb_phrase(V), noun_phrase1(N), de.

qualifying_list([A]) --> adjective_list(A).
qualifying_list([D, A]) --> demonstrative(D), measure(M), adjective_list(A).
qualifying_list([N, of | A]) --> number(N), measure(M), adjective_list(A).
qualifying_list([N, 's' | A]) --> noun_phrase1([N]), de, adjective_list(A).
qualifying_list([familiar possessive, N | A]) --> noun_phrase1([N]), adjective_list(A).

adjective_list([]) --> [].
adjective_list([A | L]) --> adjective(A), adjective_list(L).

Appendix B. The Chinese grammar.
来中国的人不说汉语。
张文在大学。
我妈妈在北京大学学习。
老师的朋友不大。
我有马的书。
她不是学生。
我没有两口爸爸。
安娜有三件毛衣。山大。
人小吗。我今年去大学。他去家好不好。我们什么时候吃饭。
这件衣服很新。
那件衣服是新的。

Appendix C.  A set of test data - Chinese Sentences.
来中国的人不说汉语。

assertion
subject person
who
verb come
object China
main_verb verb speak
direct_object chinese language
adverb not

张文在大学。

assertion
subject ZhangMen
main_verb verb to be at
direct_object university

我妈妈在北京大学学习。

assertion
subject
familiar_posessive i, I mother
main_verb verb study
no_direct_object
at_loc at, in
familiar_posessive Beijing university

Appendix D. The output based on the test data in Appendix C.
Appendix D. (Continued) The output based on the test data in Appendix C.

Research Report No. 89/1 - Page 19
assertion
subject Anna
main_verb have
direct_object
three of pullover

assertion
subject mountain
is big

question yes/no
subject person
is small

assertion
subject I, I
main_verb go
direct_object university
adverb this year

Appendix D. (Continued) The output based on the test data in Appendix C.
question good or bad?

subject he, him

main_verb verb go

direct_object family

assertion

subject we, us

main_verb verb eat

direct_object meal, cooked rice, food

adverb when

assertion

subject this [] clothes

is very new

assertion

subject that [] clothes

is a new one

Appendix D. (Continued) The output based on the test data in Appendix C.