Chapter V

Conclusion

This chapter is divided into two sections. The first section sums up the dependency grammar proposed for the analysis of Thai sentences. The second lists suggested topics for further research.

5.1 Summary of analysis

The dependency approach has been proved to be suitable for the analysis of many languages, such as English, Russian, French (Melčuk 1988), and it is claimed to be capable for analysis of any languages. In this study, it is proved to be capable for analyzing the Thai language. The dependency analysis in this study aims at the computerized parsing system. Two significant levels of dependency are recognized for Thai sentences: syntactic dependency and conceptual dependency. To obtain a representation of a syntactic dependency structure, a D-tree in this study, a system requires information on categories of each wordform in a sentence, syntactic cases which relate them and the priorities of relations to determine the construction of syntactic cases. In this study, category information is presented in terms of features: major category and minor category. There are five values of major category feature and 21 values of minor category feature. An inventory of 31 syntactic cases has been proposed for the Thai language in general, though only 28 cases are actually used for the analysis of the selected corpus. Three sets of priorities, bottom-up priority, immediacy priority and probability priority, are also proposed to determine the construction of syntactic cases in a sentence. To obtain a conceptual representation,

or a conceptual network of a sentence, an inventory of 25 conceptual cases and 23 conceptual attributes have been proposed, though only 18 cases and 11 attributes are actually used. Conceptual case constraints and case mapping are also proposed to account for the linguistic data in the corpus. There are 74 wordforms. Fifty three word-concepts, organized in a conceptual hierarchy, are postulated for these wordforms.

In addition to the above linguistic information, rules are needed for syntactic and conceptual analysis of the corpus by CUPARSE. At the syntactic level, there are three phases of rules: relative clause phase, nominalized clause phase, and main clause phase. There are altogether ten links of rules. At the conceptual level, there are three phases: subject-object resupplying phase, case assignment phase, and case selection phase. Altogether there are four links of rules.

These rules are implemented in CUPARSE together with the dictionary, each entry of which consists of a wordform, and a number of appropriate features and values. To accommodate all the 72 wordforms in the corpus, 25 features have been used to represent all the information needed for the analysis.

The actual implementation of the postulated rules and the dictionary for the analysis of the 50 sentences was successful. Though CUPARSE is capable of yielding multiple D-trees and conceptual networks for ambiguous sentences, the output obtained from the analysis of the sentence corpus consists of only a single D-tree and a conceptual network because there are no syntactic and conceptual ambiguities. The only ambiguity exists only at the lexical level.

5.2 Suggestions

5.2.1 Conceptual cases in terms of perspectives

Since the time Fillmore made acceptable the notion of case relations, there has never been an agreed set of cases and case

definitions, which seems to be both evasive and illusive. For example, "ฉัน" in sentence (2) can be assigned the case SOURCE (SOR) in one analysis or PARTNER (PARTN) in another analysis. This can also be true for "โรงเรียน" in sentence (3) and "บ้าน" in sentence (4). The case assigned in (3) can be LOCATIVE (LOC) in one analysis or TARGET (TAR) in another, and the case for (4) can be either OBJECT (OBJ) or AFFECTED (AFF).

(2) เขา ซื้อ หนังสือ จาก ฉัน
SOR, PARTN

(3) เขา ไป ที่ โรงเรียน
LOC, TAR

(4) เขา สร้าง บ้าน
OBJ, AFF

There is one interesting and significant observation in this difference among analysts. Both the SOR and PARTN cases can be equally well justified for "ฉัน" in sentence (2). The question is what causes the difference. A possible answer is the difference in perspective of the analyst. It is possible to view a situation from at least three perspectives: components of the situation, transfer of the components, and effects of the situation on the components. The difference in the assignment of case for "ฉัน" in sentence (2) can be explained in terms of these perspectives. "ฉัน" can be viewed as the source of the transfer through the sale of the book as well as the partner in this transfer. "โรงเรียน" in sentence (3) can be viewed as the target of the transfer through movement of "เขา" as well as the location at which the event occurs. In the sentence (4), "บ้าน" can be viewed as the component which comes into existence as a consequence of the situation as well as the component affected in the event.

It is evident that the problem on case analysis results from the assumption that case relation is a discrete and unique entity, which is assigned to the relationship between a noun and a predicate.

As a discrete and unique entity, a case cannot always accommodate more than one perspective. It is reasonable to envision a case which can accommodate different perspectives. If this is the assumption, case can be defined in terms of features and values. Each feature represents a different perspective. Each perspective can have different values as shown in (5).

(5) Movement : SOURCE, TARGET,...

Component : AGENT, OBJECT, PARTNER, INSTRUMENT, LOCATIVE,...

Changing : AFFECTED, CREATED,...

This is a proposition which requires further studies and perhaps implementation in an actual analysis of a natural language.

5.2.2 Priorities of relations

To construct a D-tree from a linear sequence, the simple adjacency principle is accepted to be the basic assumption of construction. However, strength of bond between adjacent lexemes are not equal, leading to the setting up of the priorities of relations. In this study, the three priorities, bottom-up priority, immediacy priority, and probability priority, are obtained from the study of relations of lexemes in the manually constructed D-trees. The generalization of the priorities for use in any sentence depends on the number of sample D-trees used as the basic to obtain priorities. To get a comprehensive set of priorities, all varieties of D-trees must be studied. However, this is a near impossibility because there exists no checklist of all possible D-trees. Other techniques are required to obtain these priorities. For example, the BPT can be obtained by considering the interaction of different relations between lexemes. If X is the head of Y and Y is the head of Z, these two relations can affect each other, and we can draw a conclusion that the relation X->Y has higher priority than Y->Z.

More of these techniques to determine priorities of relations are needed to facilitate the designing of parsing grammar. Until more

of these techniques are discovered, the study of D-tree data is still the best available way to obtain priorities of relations. This certainly requires further research. It would be even better if a software program can be written to extract these priorities according to given definitions.

5.2.3 Further development of CUPARSE

CUPARSE is a production system. The output is produced according to the set of rules specified. However, CUPARSE is not a pure production system in the sense that the knowledge and the process are clearly separated. The process in CUPARSE still relies on links and link orders. To write the analysis rules for CUPARSE, linguists have to know how to formulate algorithms, because linking rules is similar to writing a computer program. Therefore, this becomes a difficult task for most linguists without training in computer programming. CUPARSE needs further software development to enable linguists to add, delete and alter rules without having to acquire computer programming skills. This means that CUPARSE must be equipped with software to manage rule links and link orders. This additional software will make CUPARSE a friendlier tool for linguists.

However, CUPARSE is designed primarily as a tool for teaching dependency grammar. The results of analysis in every module can be displayed as graphic representations on the screen. Therefore, CUPARSE can additionally be used for teaching linguists algorithm writing as well.