Chapter I

Introduction

1.1 Rationale

There is no denying that the computer has come to be an important instrument in our life these days. Many tasks can be achieved more easily because the computer has reduced the amount of human energy and time required. Despite its efficiency, the capability of the computer is still limited. One of the limitations is in the man-machine interface. The two parties do not speak the same language. By necessity, man has to learn computer languages, such as Pascal, Basic, C, Assembly, etc., if he wishes to instruct or communicate with the computer. Therefore, it has been an aspiration of all human users to be able to use human languages in man-machine communication. What this means is that the computer must be equipped with the facilities to understand and to respond to messages in human language. The work done towards this goal is known as natural language processing and it is a very important development of the computer. Research towards this goal has been done in areas such as machine translation, a question-answering system, speech recognition, and speech synthesis. However, most research projects are on other languages, such as English, Japanese, and French. Not much has been worked on for the Thai language. To equip the computer with the facilities to understand Thai, the first step is to devise a system which can yield a representation of an understanding or an interpretation of the Thai language. This thesis is an attempt to formulate a grammar to be used in a prototype parser of Thai, which can be developed and improved and later incorporated in man-machine interface systems.

1.2 Objectives

1. To analyze syntactic and semantic structures of Thai sentences using dependency grammar.

2. To formulate linguistic rules for the parsing of a corpus of fifty Thai sentences.

3. To exhibit dependency analysis in a parsing system.

1.3 Scope

1. A corpus of fifty Thai sentences consisting of simple and complex sentences (see Appendix A).

2. The parser takes as input a sequential string of words with artificial boundaries and outputs a dependency tree and a conceptual network.

3. Concepts used in a conceptual network are restricted to word-sense meaning rather than abstract conceptual primitives.

1.4 Review of Literature

The literature to be reviewed in this study can be divided into four areas: parsing, semantic network, dependency grammar, and case relation.

1.4.1 Parsing

The work done in the area of natural language processing can be roughly divided into two independent though related types of processes: sentence analysis and sentence generation. The latter, which involves the production of sentences or texts from meaning representations, is not within the scope of this study so it will not be included in the discussion. The former process involves the

analysis of the input sentences or texts to achieve a certain level of understanding, which can serve as input for certain computer applications, such as machine translation, a question-answering system, etc., or can be outputted in terms of abstract representation. This analysis process is usually referred to as parsing.

Wilks and Jones gave the following definition for parsing.

Parsing is formally a computational process, and hopefully an actual working program on some computers, that takes sentences in a natural way (but preferably texts) and converts them by rules to some representational structure useful for further processing as might be required, for example, for translation or question-answering. (Jones 1988: 12)

A parser is, therefore, a computer program used for parsing. Many parsers have been developed using different techniques and following different theoretical frameworks. In addition, parsers can differ in the type of input they can work on, the processes they can perform and the type of output they can produce. It is possible to talk about parsers along these lines.

1. Computational process and knowledge

A parsing system usually comprises two main components: a computational process and knowledge. These two components can be either amalgamated or separated. Some parsers, such as Wood's ATN parser (1970), do not separate the knowledge used from the use of the knowledge. In other parsers, such as Colmerauer's Q-System (1970) used in the machine translation system TAUM-METEO, the knowledge and the process are clearly separate.

2. Deterministic and non-deterministic parsing

At a certain point during the parsing process more than one alternatives can be taken so a parser can be of a deterministic and a non-deterministic type. A deterministic parser, such as Marcus's PARSIFAL (1980), will select only one path discarding other less

desirable or plausible possibilities. On the other hand, a non-deterministic parser, such as Wood's ATN parser, can backtrack if the choice leads to a dead end or it will try to follow all possibilities simultaneously such as in Colmerauer's Q-System.

3. Syntactic and semantic parsers

On the basis of output, there are two types of parser: syntactic parser and semantic parser. A syntactic parser, such as Marcus's PARSIFAL, does not attempt to give the meaning of a sentence because it is intended to analyze a sentence only syntactically. A semantic parser, on the other hand, is designed to output the meaning of a sentence. This latter type can be extremely semantic-based, i.e. it does not process the syntax of a sentence. Some examples of a semantic parser are Cater's parser (1982), Riesbeck's Conceptual Dependency Analyzer (1975), and Wilks's Preference Semantics (1975). Other semantic parsers, such as Wood's ATN parser and Winogard's SHRDLU (1972), perform both syntactic and semantic parsing.

4. Linguistic basis

Knowledge for parsers need not be linguistically based. Many early parsers, such as Riesbeck's Conceptual Dependency Analyzer (1975) and Wilks's Preference Semantics (1975), do not exploit any linguistic information. However, other parsers are linguistically based, such as Phillips and Thompson's GPSG parser (1987) and Starosta and Nomura's Lexicase Parsing (1986).

CUPARSE is a prototype parser designed to implement the dependency analysis of Thai sentences in this study. It is linguistically based and non-deterministic in nature. The knowledge base consists of rules and a dictionary, and it is kept separately from the process. CUPARSE is capable of producing multiple outputs at both the syntactic and semantic analysis levels.

1.4.2 Semantic network

Output of parsers can differ. Some parsers yield syntactic representations, such as Marcus's PARSIFAL. Others are more ambitious and can yield meaning representations, such as Winogard's SHRDLU. CUPARSE is designed to give both syntactic and semantic representations of a sentence. Therefore, knowledge about meaning is important in this study.

Meaning has been studied by two groups of scholars: those interested in logical semantics and those interested in cognitive semantics. The first group consists of philosophers while the second group consists of theoretical linguists, cognitive psychologists and artificial intelligence researchers. The first group of scholars study meaning from the logical point of view, that is, seeking the truth conditional value of a sentence. The current study of meaning in this approach is influenced by the work of Frege (1970). There are many theoretical approaches which provide frameworks for studying the logical aspect of meaning, such as propositional logic, predicate logic, and modal logic. Montague Grammar (Dowty 1979) is the most outstanding framework in this approach.

The second group of scholars think that meaning is totally unrelated to truth and reference. This group of researchers study meaning from the psychological point of view. They are interested in what goes on in the human mind. They aim at explaining how humans understand language. There are several theories that coincide with this approach, such as the decompositional theory (Nida 1975), the meaning postulates theory (Carnap 1956), the semantic network theory (Quillian 1968), script (Schank 1977) and frame approach (Minsky 1975). In this study, the semantic network approach is adopted to represent the meaning of a sentence.

Semantic network was proposed by Quillian (1968) to represent the meaning of both words and sentences and is claimed to be powerful

enough to represent any sort of idea. Semantic networks are mathematical and computational structures composed of a set of nodes connected by directed arcs. Both nodes and arcs of the network can be labeled. Nodes represent concepts whereas arcs represent relations between concepts. Simmons (1973: 77) defined the network in terms of a context-free grammar as follows:

(1) *Network -> Node**

Node -> Atom + Relationset,Terminal Constant

Atom -> Ci,Li (a number prefixed with L or C)

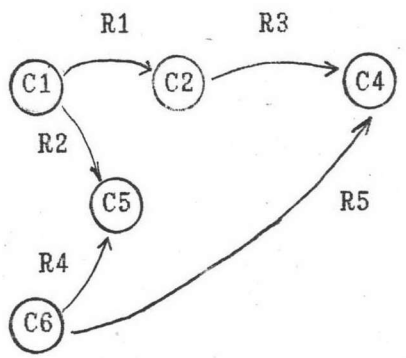
Relationset -> Relation + Node

Relation -> member of a list of semantic relations

Terminal Constant -> character string

The symbol "*" signifies one or more occurrences of the marked element; the symbol "," represents "or"; and the symbol "+" represents "and". A semantic network representation according to the definition given in the context free grammar here can be graphically represented as follows:

(2)



A semantic network can also be represented as a logical form such as R(X,Y) in which R is the relation of head X and depender Y. The same network can then be expressed in logical forms as follows:

(3) R1(C1,C2), R2(C1,C5), R3(C2,C4), R4(C6,C5), R5(C6,C4)

A semantic network is used by linguists to represent language meaning and by AI workers to represent knowledge. The meaning of the word "woman", for example, can be represented as the semantic network in (4).

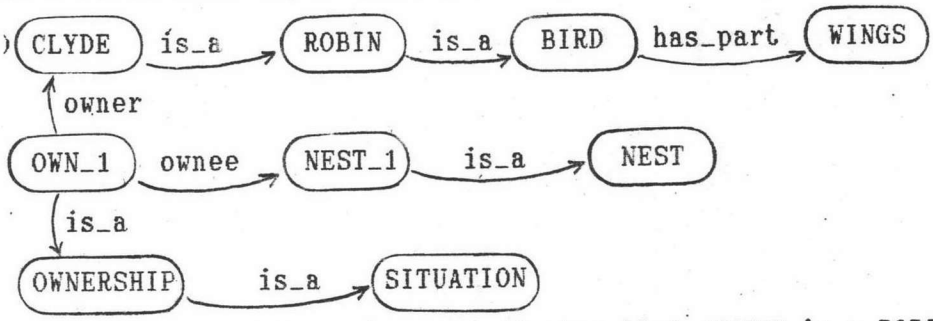
(4) is_a(WOMAN,HUMAN),
gender(WOMAN,FEMALE),
age(WOMAN,ADULT).

A semantic network can also represent the meaning of a sentence. The sentence "John broke a window with a hammer." can be represented in a semantic network as shown in (5).

(5) agent(BREAK,JOHN),
object(BREAK,WINDOW),
instrument(BREAK,HAMMER).

In addition to representing linguistic meaning, a semantic network is also used in Artificial Intelligence as a knowledge representation device. The following network represents the knowledge about a robin name Clyde (Barr and Feigenbaum 1981: 183).

(6)



This semantic network indicates that CLYDE is a ROBIN, ROBIN is a BIRD, and BIRD has WINGS as parts of it. OWN_1 is an OWNERSHIP and OWN_1 has CLYDE as an owner, NEST_1 as an ownee and NEST_1 is a NEST. In this network, NEST is a general concept of nest while NEST_1 is an instance of nest that CLYDE owns. From this knowledge representation, a question likes "What does CLYDE own?" can be answered easily. Since we can infer from this representation that CLYDE is an owner and NEST_1 is an ownee of OWN_1, and NEST_1 is an instance of NEST, we can conclude that "CLYDE owns a nest." In answering the question "Does CLYDE have wings?", we can see from the network that CLYDE does not have any direct relationship with WINGS. However, since the network indicates that CLYDE is a ROBIN and ROBIN is a BIRD, the property which belongs to BIRDS also belongs to CLYDE;

therefore the answer to the question above is yes.

It is obvious that a semantic network is merely a means of drawing relations between nodes. However, the assignment of content to each node depends on users or designers of the system. The previous examples illustrate the use of nodes to represent word concept, primitive concepts, as well as instances of concepts such as NEST_1, which is an instance of the concept NEST.

In this present study, semantic network is used to represent the meaning of a sentence. Each node is labeled with a concept, either a primitive concept or a word concept. The use of primitive concepts has an advantage, since a single semantic network with nodes labeled with primitive concepts can represent the shared conceptual meaning of the sentences the surface forms of which are lexically different. Such sentences are called semantic paraphrases. The pair of sentences in (7) are examples of semantic paraphrases. By the use of primitive concepts, the meaning of these semantic paraphrases can be represented as shown in (8).

(7) a. "John bought the boat from Mary".

b. "Mary sold the boat to John".

(8) and(TRANSFER,TRANSFER),
source(TRANSFER,JOHN), goal(TRANSFER,MARY), theme(TRANSFER,MONEY),
source(TRANSFER,MARY), goal(TRANSFER,JOHN), theme(TRANSFER,BOAT).

The semantic network above not only indicates the relationships between "sell" and "buy", it also expresses additional information which is not directly conveyed by either of the two sentences. This is the information about the money involved in this transaction.

However, there is no principled way in postulating the primitive concepts. The postulation of primitive concepts is usually determined by the framework or the nature of each researcher's work itself.

Primitive concepts can be evasive and are difficult to validate. Therefore, many prefer to use word concepts. These are concepts which correspond quite closely to words. Semantic relations in the second approach are, therefore, relations between words or word concepts.

Sentences expressing the same event with the same content words but different syntactic structures are represented by the same semantic network. Such sentences are called syntactic paraphrases. Consider the example (9). The three sentences are represented by the same semantic network as shown in (10).

(9) John broke the window with a hammer.
The window was broken by John with a hammer.
The window was broken with a hammer by John.

(10) agent(BREAK,JOHN),
object(BREAK,WINDOW),
instrument(BREAK,HAMMER).

In this present study the word-concept approach is adopted; concepts are supposed to correspond to surface word forms. This is because the objective of this study is to formulate a prototype grammar which can be used in a parser to analyze the semantic structure of a sentence. Since a semantic network is composed of concepts and relations between concepts, the term, "conceptual network" will be used to refer to the meaning representation outputted by CUPARSE.

1.4.3 Dependency grammar

A parser requires knowledge in theoretical linguistics as well as artificial intelligence. Earlier linguists used computer programs, similar to parsers, to check validity of their theories or generalizations. AI researchers used to look at languages for clues to understand human cognition, which they aimed at simulating.

Gradually, the two groups began to work together. Linguists now are writing formal grammars which can be accommodated into computer programs. As a consequence, new development in linguistic theories emerge, such as Kaplan and Bresnan's Lexical-Functional Grammar (1982), Gazdar's Generalized Phrase Structure Grammar (1985), and Hudson's Word Grammar (1984).

There are two main approaches in the theory of grammar: constituency approach and dependency approach. A constituency grammar describes language in terms of components or constituents of a construction at different levels. The most popular grammars in this approach are Structural, Phrase-structure, and Transformational grammars. A dependency grammar, on the other hand, describes language in terms of relations between head and depender in a construction.

Dependency theory was first formalized by Tesniere (1959 cited by Hudson, 1984: 76). It has always been used as a formal means for representing syntactic structures of sentences, especially in Europe and in Classical and Slavic study circles. However, it has been overshadowed by the popularity of constituency grammars, which have been applied extensively and in most cases, in the description of the English language, especially for the description of syntax during the 1950-1970's.

Many current grammars have been written within the dependency approach. Examples are Word Grammar (Hudson 1984), and Dependency Syntax (Melčuk 1988). There are more similarities than differences among these grammars. All agree that a proposition, or a sentence, can be represented formally as a network consisting of a number of nodes which are linked by dependency relations. It is true that no two dependency grammarians agree on the number of dependency relations required in the description of human language. Neither do they agree on the extent of the abstractness in the representation of meaning in language. However, the major difference lies in whether or

not a syntactic depender can be linked to multiple heads. Hudson (1984) allows this but Melčuk (1988) does not. In this study, Melčuk's Dependency Syntax has been adopted.

Melčuk represents syntactic structure of a sentence as a dependency tree (D-tree). He contrasts it with a phrase structure tree (PS-tree) in Transformational grammar and points out five major respects in which D-tree is different from PS-tree.

1. A PS-tree shows the structure of an expression in terms of the grouping of items into an expression of a higher order. It concentrates on constituency. It shows which items go together with other items. A D-tree, on the other hand, shows the structure of an expression in terms of hierarchy links between items. It concentrates on relations. It shows which items are related to which other items and in what way.

2. In a PS-tree, the syntactic class membership (i.e., categorization) of an item is specified as an integral part of the syntactic representation. This class membership also applies at immediate levels of a PS-tree , resulting in categories such as NP, VP. In a D-tree, on the other hand, class membership is not specified.

3. In a PS-tree, there are terminal and also non-terminal nodes but most nodes are non-terminal. A D-tree, on the contrary, contains terminal nodes only.

4. In a PS-tree, nodes must be ordered linearly. In a D-tree, on the other hand, linear ordering is of no importance.

5. A PS-tree does not specify the type of syntactic link existing between two items. A D-tree, on the other hand, puts particular emphasis on specifying in detail the type of any syntactic relation obtained between two related items.

In addition to Melčuk's idea, it is evident that a constituency grammar is usually used to describe only syntax, but a dependency grammar can describe both syntax and semantics in the same

manner as a D-tree and a conceptual network respectively. It is, therefore, convenient to use dependency grammar as an approach to analyze a sentence syntactically as well as semantically. Dependency grammar has been proved to be suitable for the analysis of many languages. This study attempts to show that this approach is also appropriate for the analysis of the Thai language.

1.4.4 Case relation

According to the traditional Case Grammar (Fillmore 1968), case is defined as a relation between verb and noun in the deep or abstract structure and it is claimed to be a universal meaning primitive, representing one of a set of judgements human beings are capable of making about the events that are going on around them. The notion has been widely accepted, but a weakness in the Case Grammar framework is that there has never been an agreed set of case relations even in one language. For Thai, Lekawatana (1970) postulated nine case relations whereas Sankaworn (1983) postulated sixteen case relations.

A new approach to case relation in Thai is proposed (Thepkanjana et. al. 1989). Instead of having direct mapping from surface form to case relation, two types of case relation, a syntactic case and a conceptual case, are proposed. Syntactic case relations integrate the notion of grammatical relations and some semantic elements. Manifestation of syntactic cases is language-specific in that they are based on surface syntactic characteristics of different languages. The postulation of syntactic case relations is based on the principle called "isomorphism", which states that there is one-to-one correspondence between form and meaning. The conceptual case, on the other hand, are pragmatically based and further removed from the surface linguistic forms. Syntactic case relations map, in a many-to-many fashion, onto conceptual case

relations. For example, the relator, "กับ", indicates that the arguments which are syntactically placed after it represent additional or background entities which are presented in a certain state of affairs; therefore, the same syntactic case will be the interpretation of the relator, "กับ", in all sentences. However, at the conceptual level, "กับ" is interpreted as different conceptual case relations, such as partner, location, and means in sentences "เขา พูด กับ พ่อ", "เขา นั่ง กับ พื้น" and "เขา ปั้น มา กับ มือ" respectively.

In this study, syntactic case relations are defined as the dependency relations at the syntactic level. Manifestation of syntactic cases is language specific. Conceptual case relations are defined as the dependency relations at the conceptual level. They are claimed to be language independent.