Tạp chí Tin học và Điều khiển học, T.21, S.2 (2005), 162–174

FEATURE-BASED GRAMMAR IN ADAPTATION TO VIETNAMESE NATURAL LANGUAGE PROCESSING

TRAN NGOC TUAN, PHAN THI TUOI

Ho Chi Minh City University of Technology

Abstract. This paper presents in brief about grammar augmented with feature system, unification-based grammar and unification parsing algorithm, applying to Vietnamese natural language processing. Vietnamese language has many syntactic differences from English language, that cause many difficulties in applying conventional methods to Vietnamese language processing. While English language processing takes full advantage of lexical morphology, Vietnamese processing is not able to. We propose here a semantic approach in creating feature system for Vietnamese lexicon, and the unification parsing for Vietnamese noun phrase which heads by a noun of type. The demonstration program is written in Java which uses library packages provided by SourceForge for education purposes.

Tóm tắt. Bài báo trình bày tóm tắt lý thuyết về văn phạm gia tố có hệ thống nét, văn phạm hợp nhất và giải thuật phân tích trên văn phạm hợp nhất, áp dụng trong xử lý ngôn ngữ tiếng Việt.

Những khác biệt về cú pháp giữa tiếng Việt và tiếng Anh làm cho khả năng vận dụng các phương pháp kinh điển vào xử lý ngôn ngữ tiếng Việt bị nhiều hạn chế. Chẳng hạn hình vị từ vựng trong tiếng Anh rất phong phú và cung cấp nhiều thông tin quyết định cho quá trình phân tích cú pháp, nhưng không áp dụng được cho tiếng Việt.

Chúng tôi đưa ra một tiếp cận xây dựng hệ thống nét ngữ nghĩa cho danh từ tiếng Việt, xây dựng văn phạm hợp nhất cho cụm danh từ tiếng Việt, giải thuật phân tích hợp nhất cho cụm danh từ tiếng Việt, đối với danh từ có từ chỉ loại đi kèm. Chương trình thử nghiệm được xây dựng bằng ngôn ngữ Java, trong đó sử dụng gói thư viện ngôn ngữ tự nhiên được cung cấp bởi SourceForge dành cho các mục tiêu học tập và nghiên cứu.

1. INTRODUCTION

Parsers belong to the most basic tools in natural language processing (NLP) and most NLP applications use some form of parser. In a machine translation system, a parser is used in the phases of source sentence analysis, and target sentence generation ([4, 6]). Parsers need grammatical description of the languages they analyse. Many grammar formalisms use feature structures to represent the syntactic properties of grammatical units, including Lexical Functional Grammar (LFG), Head-Driven Phrase Structure Grammar (HPSG), Definite Clause Grammar (DCG) [1]. They are commonly called feature-based augmented grammars.

Feature-based grammars along with traditional parsing algorithms have been taken advantage in many types of NLP applications in English language [1]. Researching on the application of feature-based grammar to Vietnamese NLP, we recognized the limitation of Vietnamese grammar: the independence between syntactic rules and lexical-morpheme in Vietnamese pre-

vents us to take advantages of feature-based grammar in Vietnamese syntactic parsing. In this paper, we present an adaptation of feature-based grammar in which we propose the semantic feature system for Vietnamese nouns and application of the unification parsing algorithm which applies to noun phrase analysis in Vietnamese language. This phrasal parsing approach can be extended into any combination parsing, which plays very important role in sentence parsing, the heart of any NLP application.

A demonstration program written in Java, in which free NLP packages provided by SourceForge [7] were used, also indicates the practical aspect of the semantic feature system. The result would provide advantages in Vietnamese NLP and particularly in English-Vietnamese machine translation research.

2. FEATURE-BASED AUGMENTED GRAMMAR

2.1. Feature structure

A feature set F consists of relevant properties of grammatical units; a set of feature values V_F consists of possible values which are able to assign to a feature in F. Constituent (also called feature structure) is a mapping from F to V_F which represents the relationship between features and their values.

Given

$$F = \{\text{ROOT, CAT, NUMBER}\}$$

$$V_F = \{\text{ART, s, p, “a”, “fish”}\}$$

Constituent

$$\begin{aligned} &\text{ART1 :(CAT ART} \\ &\qquad \text{ROOT “}a\text{”} \\ &\qquad \text{NUMBER } s) \end{aligned} \tag{1}$$

says it is a constituent in the category ART that has as its root the word “a” and is singular. In short form:

$$\text{ART1: (ART ROOT a NUMBER s).}$$

Feature structure can be used to represent larger constituent, in which feature structures themselves can occur as values. Special features 1,2,3,... will stand for the first constituent, second constituent,..., as needed. With this, the representation of the NP constituent for the phrase *a fish* could be

$$\begin{aligned} &\text{NP1 :(NP NUMBER } s \\ &\qquad \text{1 (ART ROOT “}a\text{”} \\ &\qquad\quad \text{NUMBER } s) \\ &\qquad \text{2 (}N\text{ ROOT “}fish\text{”} \\ &\qquad\quad \text{NUMBER } s)) \end{aligned} \tag{2}$$

The rules in an augmented grammar are stated in terms of feature structures rather than simple categories. Variables are allowed so that a rule can apply to a wider range of situations. For example, a rule of noun phrase would be as follow:

$$\text{(NP NUMBER ?n)} \rightarrow \text{(ART NUMBER ?n) (N NUMBER ?n)}$$

2.2. Morphological Analysis and Lexicon

A lexicon must be defined prior to the grammar specification. Instead of including all words with their different grammatical forms, a lexicon can consists of constituents as entries. Finite state techniques will be used to produce relevant grammatical forms based on a constituent entry. Table 1 is a small lexicon which contains constituents of nouns, adjectives, verbs, articles, and prepositions that are typical in morphological analysis.

2.3. Feature-based Augmented Grammar

An augmented rule has the form: $A \rightarrow X_1 X_2 X_n$ where the LHS is the super-constituent, the RHS consists of sub-constituents. Each symbol is a constituent with the form: (Category {Feature Variable | Value}*) The constituent X_i whose feature values are identical to those of constituent A is call the head sub-constituent. Such value set is called head features. Table 2 is a simple augmented grammar for English language.

3. Unification Grammar

Feature structures can be generalized to the extent that they make the context-free grammar unnecessary. The entire grammar can be specified as a set of constraints between feature structures. Such systems are called unification grammars. The key concept of a unification grammar is the extension relation between two feature structures.

3.1. Extension

Feature structure $F1$ extends (or is more specific than) a feature structure $F2$ if every feature value in $F1$ is specified in $F2$. For example, the feature structure

$$\begin{array}{rl} & \text{(CAT } V \\ & \text{ROOT } cry\text{)} \\ \textit{extends} & \text{(CAT } V\text{)} \end{array} \tag{3}$$

On the other hand, neither of the feature structures:

$$\begin{array}{rl} & \text{(CAT } V \\ & \text{ROOT } cry\text{)} \\ \text{and} & \text{(CAT } V \\ & \text{VFORM } pres\text{) extend the other.} \end{array} \tag{4}$$

3.2. Unification

Two feature structures unify if there is a feature structure that is an extension of both. The most general unifier is the minimal feature structure that is an extension of both.

Two feature structures:

(CAT V and (CAT V
ROOT cry) VFORM pres)

have their most general unifier as:

(CAT V
ROOT cry
VFORM pres)

3.3. Unification Grammar

Rule form:

(S INV- VFORM ?v {pres past} AGR ?a) → (NP AGR ?a) (VP VFORM ?v {pres past} AGR ?a)

to be specified in unification grammar using a rule and a set of feature equations:

$$\begin{aligned} X0 \rightarrow X1X2 \quad & \text{CAT0} = S \\ & \text{CAT1} = \text{NP} \\ & \text{CAT2} = \text{VP} \\ & \text{AGR0=AGR1=AGR2} \\ & \text{VFORM0=VFORM2} \end{aligned} \tag{5}$$

In short form:

$$\begin{aligned} \text{S} \rightarrow \text{NP VP} \quad & \text{AGR=AGR1=AGR2} \\ & \text{VFORM=VFORM2} \end{aligned} \tag{6}$$

Table 3 is a unification grammar with the same specifications as the grammar in table 2.

4. UNIFICATION ALGORITHM

4.1. Feature Structure as DAG

A node: for constituent or value

An Arc: for feature

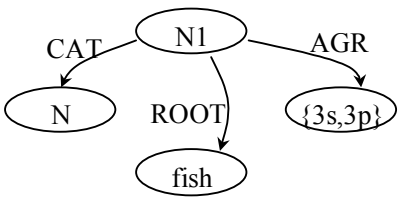
A source node: has no incoming edges. Feature structure DAGs have a unique source node, called the root node.

A sink node: has no outgoing edges. The sinks of feature structure DAGs are labeled with an atomic feature or set of features.

Constituent

N1: (CAT N
ROOT fish
AGR {3s 3p})

is represented as DAG:



With graph unification algorithm (tables and figures, Figure 3) in hand, the algorithm for constructing a new constituent using the graph unification equations can be described as follow.

4.2. Algorithm to Create New Constituent

Given a rule $X0 \rightarrow X1...Xn$ and a set of feature equations of form $Fi = V$, where $SC1, ..., SCn$ are the subconstituents corresponding to $X1, ..., Xn$. This algorithm builds a DAG that satisfies all the feature equations.

1. Create a node $CC0$ to be the root of new feature structure.
2. Make a copy of each DAG rooted SCi (call the new root of each CCi), add and arc labeled i from $CC0$ to each CCi.
3. For each feature equation $Fi = V$ (V is value), follow the F link from node CCi to node Ni and unify Ni with V.
4. For each feature equation of form $Fi = Gj$:

4a. If there is an F link from CCi, and a G link from CCj, then:

i. Follow the F link to node Ni and the G link to node Nj;

ii. Unify Ni and Nj, using graph unification algorithm, to create new node X;

iii. Change all arcs pointing to either Ni or Nj to point to X;

4b. If there is no F link from CCi, but there is a G link from CCj to Nj, create an F link from CCi to Nj;

4c. If there is no G link from CCj, but there is an F link from CCi to Ni, create a G link from CCj to Ni.

5. CHARACTERISTICS OF WORDS IN VIETNAMESE GRAMMAR

5.1. Characteristics of Vietnamese words

According to [2], Vietnamese words are not inflectional, compound nouns appear in a free-rule structure, and there is also homonymic phenomenon. In addition, important grammatical category such as person, gender, number, tense, and case, which are morphologic category in English, are syntactic and semantic categories in Vietnamese.

Consider English sentences:

I know her, and she knows me.

I've liked her for 3 years.

I liked her 3 years ago.

Corresponding sentences in Vietnamese:

Tôi biết cô ta, và cô ta biết tôi.

Tôi thích cô ta đã 3 năm.

Tôi đã thích cô ta 3 năm về trước.

In English, the lexical inflection: I-me, she-her, know-knows, like-liked makes the words implying syntactic functions, and the parser will take this advantage easily. The rules are

strict as me must plays objective function, while I is a subject. If the subject is she, then the verb must be in third person singular (knows).

In comparison with Vietnamese, the pronoun ti has no morphic change when being used in different functions: subjective or objective. The tense of the sentence is not defined by the verb morpheme (thích), which never inflects, but by adverbial words (đã, về trước) and their positions.

5.2. Vietnamese Word Categories

According to criteria using for categorization: lexical meaning, syntactic function, and possibilities of combination into phrase and sentence, ([2]) Vietnamese words are classified into two common groups, substantive and expletive. Substantive words are words with specific meanings. A Substantive word can be used as a grammatical component in a sentence, and it can be the central word of a phrase. Expletive words have no meaning, can not be used to create a sentence independently. They are used to link other words to create a phrase. Further on, substantive words are categorized into: nouns, verbs, adjectives, pronouns and numerals; expletive words includes: adjuncts, conjunction, particles, and interjection.

5.3. Discussion

Parsing Vietnamese sentence is a difficult task not only due to the word segmentation [5] - Vietnamese words are not explictly separated by blanks as in English language, the previous section indicated that the parsing process will need additional semantic and syntactic information. For this reason, it would be difficult for Vietnamese NLP if we process in two separated phases, syntactic analysis and semantic analysis as in Indo-Euro NLP (e.g. English NLP).

To overcome the poverty of lexical and syntactic features in Vietnamese words, we propose a sematic approach for Vietnamese word feature structure. The feature set will not only consist of syntactic properties but semantic properties as well. In parsing, the identification of thematic roles not only is fundamental to semantic interpretations but also can reduce syntactic branches and ambiguities. According to [3], five important parameters which help to determine the themantic roles of a constituent are:

a. Syntactic categories and semantic features of constituents.

b. Case frames and case restrictions of verbs.

c. Syntactic configurations and word order.

d. Inflection, including prefixes and suffixes.

e. Real world knowledge.

In other words, the relation between words is represented through syntactic and semantic constraints and word order. In the next section, we propose a feature structure system and unification grammar which is reasonable for effective parsing of noun phrases in Vietnamese language.

6. UNIFICATION GRAMMAR FOR VIETNEMESE NOUN PHRASE PARSING

This section presents our approach in adapting unification grammar to Vietnamese NLP. Based upon the fact that parsing rely at the heart of any NLP application, and in turn parsing depends itself on the constrain among constituents which is called here the feature system. By default, English has its own feature system derived from lexical morpheme (gender, number, case) and grammatical rules (tense, mood). Our adaptation is to build a structure system based on semantic features and to apply the effective tools-unification grammar, unification algorithm-for parsing noun phrase, which is an important phase of sentence parsing.

6.1. Feature Structure of Vietnamese Nouns

In Vietnamese language, nouns are classified into sub-categories [6]: proper noun, synthetic noun, type, unit, material, creature, thing, abstract noun. Most of noun phrases are combined from two nouns, and the combination must comply with certain rules. These combination rules are semantic specific, and we attempt to represent by feature structure.

For example, the nouns for type con, cái, chiếc, hòn, bức, cuốn, quả ... can combine with nouns for creature (gà, mèo) or nouns for thing (bàn, bi, vách, sách) to form noun phrases, but not always be meaningful. Legal combinations could be: cái bàn, hòn bi, con gà, bức vách, cuốn sách; on the other hand, hòn bàn, con chiếu, cái gà, cuốn vách, bức sách are not legal. The constituent for nouns includes necessary semantic features in order to prevent illegal combinations, as proposed in the following.

- Attribute: LEX, CAT, NATURE, SHAPE, SIZE.

- Value: *nk* (noun for type), *nt* (noun for thing), *na* (noun for animal), <lexical value>, round, thin, small, big,...

- Feature: LEX "bàn", CAT *nt*, SHAPE round, SIZE big,...

- Constituents:

NK1	(CAT *nk*	NT1	(CAT nk
	LEX quả		LEX bóng
	SHAPE round		SHAPE round
	SIZE big		SIZE big
	NATURE thing)		NATURE thing)

Table 4 is a small lexicon of Vietnamese nouns.

6.2. Unification Grammar

Table 5 is the unification grammar proposed for combination rules of Vietnamese compound nouns:

1. NP → NK NT	2. NP → NK NA
CAT0 = *nt*	CAT0 = *na*
CAT1 = *nk*	CAT1 = *nk*
CAT2 = *nt*	CAT2 = *na*
SHAPE0 = SHAPE1 = SHAPE2	SHAPE0 = SHAPE1 = SHAPE2
SIZE0 = SIZE1 = SIZE2	SIZE0 = SIZE1 = SIZE2
NATURE0 =NATURE1 =NATURE2	NATURE0 =NATURE1 =NATURE2

Table 5. Unification Grammar

Using feature-based lexicon, unification grammar, and unification algorithm, the compound nouns are created: quả bóng, hòn bi, cuốn sách; while preventing the creation of combination hòn bóng, quả sách, cuốn bi,..., as in the following illustration. Given constituents NK1 "quả" and NT1 "bóng":

NK1	(CAT nk	NT1	(CAT nt
	LEX quả		LEX bóng
	SHAPE round		SHAPE round
	SIZE big		SIZE big
	NATURE {thing, plants})		NATURE thing)

Their DAG representations are showed in Figure 1.

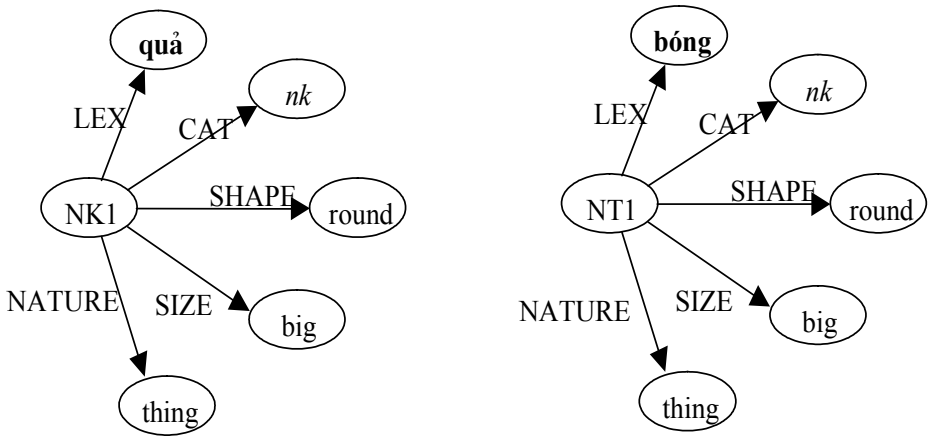


Figure 1. DAG representation of constituents NK1 "quả" and NT1 "bóng"

Apply the Algorithm to Create New Constituent (section 4.2) using unification grammar described in table 5, the new constituent of compound noun "quả bóng" with its DAG representation given in Figure 2. Practical results are showed in Figure 4.

7. CONCLUSION

Sentence parsing is the most important phase of NLP in general and machine translation in particular. In this paper, we present standard methods for sentence parsing applied to English language including feature structure, feature-based augmented grammar, unifica-

tion grammar and unification algogithm. Unfortunately, differences between Vietnamese and English languages prevent a smooth application of mentioned methods to Vietnamese NLP. Based on Vietnamese characteristics, a semantic approach is proposed so that we can adapt these effective methods to Vietnamese NLP, particularly to noun phrase parsing. The results can be extended to apply to parsing for other types of combination, based on syntactic and semantic combination rules. The results are also helpful in Vietnamese NLP systems and machine translation system which related to Vietnamese language.

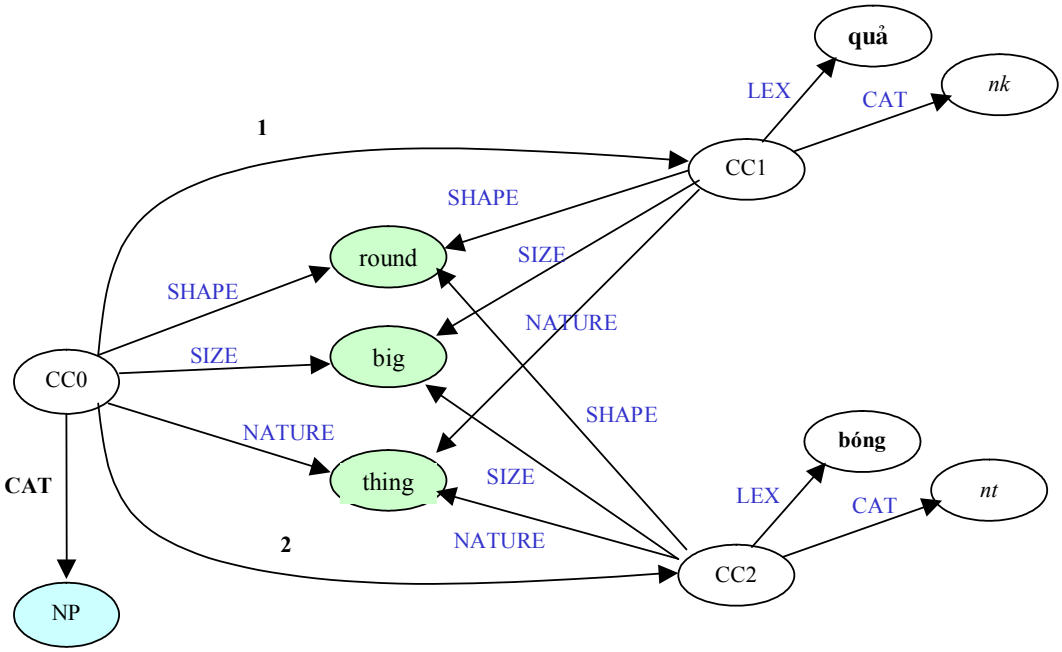


Figure 2. DAG representation of constituent "quả bóng" is the unify from DAG NK1 and DAG NT1 based on unification grammar given in Table 5.

In theoretical model, the unification grammar is able to apply to Vietnamese language parsing, but application would be much dependent on the availability of lexicon and grammar rules. In practice, feature-based lexicons have been built manually for English, French and other European languages by NLP research groups [8]. For Vietnamese language, to take full advantages of unification grammar, it would take time and close cooperation of multiple disciplines for building such type of resources.

8. TABLES AND FIGURES

Figure 3. Graph Unification Algorithm

Input: Two DAGs rooted at Ni and Nj

Output: Unified DAG

Method:

1. If $Ni = Nj$ then return Ni and succeed.
2. If both Ni and Nj are sink nodes, then if their labels have a non-null intersection, return a new node with the intersection as its label. Otherwise, the DAGs do not unify.
3. If Ni and Nj are not sinks, then create a new node N. For each arc labeled F leaving Ni to NFi:

3a. If there is an arc labeled F from Nj to NFj, then recursively unify NFi and NFj. Build an arc labeled F from N to the result of recursive call.

3b. If there is no arc labeled F from Nj, build an arc labeled F from N to NFi.

3c. For each arc labeled F from Nj to NFj where there is no F arc leaving Ni, create a new arc labeled F from N to NFj.

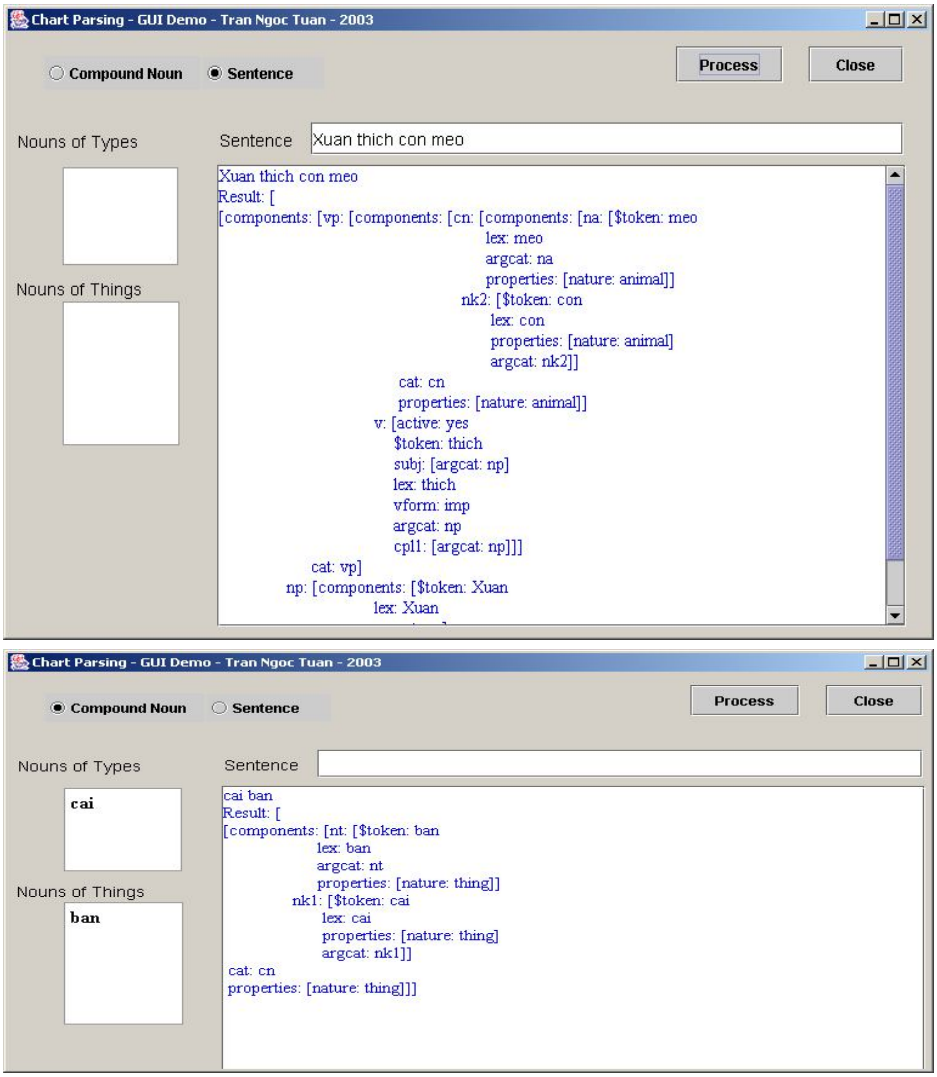


Figure 4. Practical results: parse trees of a sentence with compound noun, and a separate compound noun

Table 1. Lexicon of English Language

a:	(**CAT** ART **ROOT** A1 **AGR** 3s)
be:	(**CAT** V **ROOT** BE1 **VFORM** base **IRREG-PRES** + **IRREG-PAST** + **SUBCAT** {_adjp _np})
cry:	(**CAT** V **ROOT** CRY1 **VFORM** base **SUBCAT** _none)
dog:	(**CAT** N **ROOT** DOG1 **AGR** 3s)
fish:	(**CAT** N **ROOT** FISH1 **AGR** {3s 3p} **IRREG-PL** +)
happy:	(**CAT** ADJ **SUBCAT** _vp:inf)
he:	(**CAT** PRO **ROOT** HE1 **AGR** 3s)
is:	(**CAT** V **ROOT** BE1 **VFORM** pres **SUBCAT** {_adjp _np} **AGR** 3s)
Jack:	(**CAT** NAME **AGR** 3s)
man:	(**CAT** N1 **ROOT** MAN1 **AGR** 3s)
men:	(**CAT** N1 **ROOT** MAN1 **AGR** 3p)
saw:	(**CAT** N **ROOT** SAW1 **AGR** 3s)
saw:	(**CAT** V **ROOT** SAW2 **VFORM** base **SUBCAT** _np)
saw:	(**CAT** V **ROOT** SEE1 **VFORM** past **SUBCAT** _np)
see:	(**CAT** V **ROOT** SEE1 **VFORM** base **SUBCAT** _np **IRREG-PAST** + **EN-PASTPRT** +)
seed:	(**CAT** N **ROOT** SEED1 **AGR** 3s)
the:	(**CAT** ART **ROOT** THE1 **AGR** {3s 3p})
to:	(**CAT** TO)
want:	(**CAT** V **ROOT** WANT1 **VFORM** base **SUBCAT** {_np_vp:inf _np_vp:inf})
was:	(**CAT** V **ROOT** BE1 **VFORM** past **AGR** {1s 3s} **SUBCAT** {_adjp_np})
were:	(**CAT** V **ROOT** BE **VFORM** past **AGR** {2s 1p 2p 3p} **SUBCAT** {_adjp_np})

Table 2. Augmented Grammar

1. (S INV - VFORM ?v {pres past} AGR ?a) →
   (NP AGR ?a) (VP VFORM ?v {pres past} AGR ?a)
2. (NP AGR ?a) → (ART AGR ?a) (N AGR ?a)
3. (NP AGR ?a) → (PRO AGR ?a)
4. (VP AGR ?a VFORM ?v)→ (V SUBCAT _none AGR ?a VFORM ?v)

5. (VP AGR ?a VFORM ?v) → (V SUBCAT _np AGR ?a VFORM ?v) NP
6. (VP AGR ?a VFORM ?v) →
   (V SUBCAT _vp:inf AGR ?a VFORM ?v) (VP VFORM inf)
7. (VP AGR ?a VFORM ?v) →
   (V SUBCAT _np_vp:inf AGR ?a VFORM ?v) NP (VP VFORM inf)
8. (VP AGR ?a VFORM ?v) →
   (V SUBCAT _adjp AGR ?a VFORM ?v) ADJP
9. (VP SUBCAT inf AGR ?a VFORM inf) →
   (TO AGR ?a VFORM inf) (VP VFORM base)
10. ADJP → ADJP
11. ADJP → ADJP (SUBCAT _inf) (VFORM inf)

Table 3. Unification Grammar

1'.	S → NP VP	AGR = AGR1 = AGR2, VFORM = VFORM2
2'.	NP → ART N	AGR = AGR1 = AGR2
8'.	VP → V ADJP	SUBCAT1 = _adjp, VFORM = VFORM1, AGR = AGR1
10'.	ADJP → ADJ	

Table 4. Lexicon of Vietnamese nouns

NK1	(**CAT** nk **LEX** "quả" **SHAPE** round **SIZE** big **NATURE** {thing, plants})	NK3	(**CAT** nk **LEX** "cuốn" **SHAPE** square **SIZE** bé **NATURE** thing)
NT1	(**CAT** nt **LEX** "bóng" **SHAPE** round **SIZE** big **NATURE** thing)	NT3	(**CAT** nt **LEX** "sách" **SHAPE** square **SIZE** small **NATURE** thing)
NK2	(**CAT** nk **LEX** "viên" **SHAPE** round **SIZE** small **NATURE** thing)	NK4	(**CAT** nk **LEX** "con" **SHAPE** any **SIZE** small NATURE animal)
NT2	(**CAT** nt **LEX** "bi" **SHAPE** round **SIZE** small **NATURE** thing)	NA1	(**CAT** na **LEX** "mèo" **SHAPE** any **SIZE** small **NATURE** animal)

REFERENCES

[1] James Allen, *Natural Language Understanding*, Benjamin/Cummings Publishing Company, 1995 (83–118).

[2] PTS. Đỗ thị Kim Liên, *Ngữ pháp Tiếng Việt*, Education Publisher, second edition, 2002 (17–19, 44–47).

[3] Chen, K. J., C. R. Huang and L. P. Chang, The Identification of Thematic Roles in Parsing Mandarin Chinese, *Proceedings of ROCLING II* (Taipei, Taiwan) (1989).

[4] Phan Thị Tươi, Nguyễn Chí Hiếu, Phân tích cú pháp và dịch máy, *Journal of Science and Technology* **5** (3&4) (2002).

[5] Trần Ngọc Tuấn, Vietnamese Word Segmentation using Corpus and Statistical Models, *Proceedings of School on Scientific Computing and Applications, HCMUT*, March 2002, Ho Chi Minh City, VietNam (2002) 135–140.

[6] Helmut Schmid, Parsing and Disambiguation with Feature-Based Grammar, *Proceedings of AIMS 2000* (Arbeitspapiere des Instituts fr Maschinelle Sprachverarbeitung) Stuttgart University, Germany, 2000.

[7] SourceForge. net, 2003, *nlpFarm*, nlplib-0.2.1.

[8] www.ldc.upenn.edu, Linguistic Data Consortium-University of Pennsylvania.

Nhận bài ngày 18 - 5 - 2004

Nhận lại sau sửa ngày 11 - 8 -2005